HAL open science

Co-spécification système exécutable basée sur des modèles - Application à la conduite interactive d'un procédé industriel critique

Fabien Bouffaron

▶ To cite this version:

Fabien Bouffaron. Co-spécification système exécutable basée sur des modèles - Application à la conduite interactive d'un procédé industriel critique. Sciences de l'ingénieur [physics]. Université de Lorraine, 2016. Français. NNT : 2016LORR0001 . tel-01754485v2

HAL Id: tel-01754485

https://theses.hal.science/tel-01754485v2

Submitted on 27 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

École Doctorale IAEM
Lorraine

Co-spécification système exécutable basée sur des modèles

Application à la conduite interactive d'un procédé industriel critique

THESE

présentée et soutenue publiquement le 08/01/2016
pour l'obtention du grade de

Docteur de l'Université de Lorraine

(mention Automatique, Traitement du Signal et des Images, Génie Informatique)

par

Fabien BOUFFARON

Composition du Jury

Président : *Dominique Mery, Professeur, LORIA, Université de Lorraine*

Rapporteurs : *Ali Charara, Professeur, Heudiasyc, Université Technologie de Compiègne*
Daniel Krob, Professeur, LIX, Ecole Polytechnique

Examinateurs : *Eric Bonjour, Professeur, ERPI, Université de Lorraine*
Patrick Millot, Professeur, LAMIH, Université de Valenciennes et du Hainaut-Cambrésis
Gérard Morel, Professeur, CRAN, Université de Lorraine, (Directeur de thèse)

Centre de Recherche en Automatique de Nancy, UMR7039 Université de Lorraine CNRS

« Se réunir est un début, rester ensemble est un progrès, travailler ensemble est la réussite.»

Henry Ford, industriel et fondateur de Ford.

Remerciements :

Nous y voilà !!!

Les trois dernières pages de ce mémoire, se situant paradoxalement au début[1].

« *Nous avons revisité le processus itératif, récursif et concourant de co-spécification-système basée sur des modèles exécutables selon le paradigme holonique* », il m'aura fallu quatre années pour finalement comprendre cette phrase, mais pas que … =)

Thèse dans le domaine de l'Ingénierie des Systèmes, je me dois de commencer par des remerciements « systémiques ». En ce sens, MERCI A TOUS !!! En effet, « *la somme des parties n'est pas égale au tout* » et ce mémoire est le résultat de nombreuses rencontres et échanges avec un ensemble de personnes dont j'espère ne pas en oublier.

Ce travail de thèse a été réalisé dans le département ISET (*Ingénierie des Systèmes Eco-Techniques*) du CRAN (*Centre de Recherche en Automatique de Nancy*) de l'Université de Lorraine. Je remercie, les professeurs Alain RICHARD, puis Didier WOLF de m'avoir accueilli au sein du laboratoire, et offert un cadre propice à l'aboutissement de ces travaux.

Je remercie les rapporteurs de ce mémoire de thèse, le professeur Ali CHARARA du laboratoire HEUDIASYC (*HEUristique et DIAgnostic des SYstèmes Complexes*), ainsi que le professeur Daniel KROB du LIX (*Laboratoire d'Informatique de l'Ecole Polytechnique*) d'avoir consacré du temps pour l'examen de mon mémoire. Vos retours m'ont permis de prendre du recul sur mon travail.

Je remercie le professeur Dominique MERY du LORIA (*Laboratoire Lorrain de Recherche en Informatique et ses Applications*) qui a accepté de présider mon jury de thèse. Mais également, pour nos réunions de travail autour de la formalisation du processus de co-spécification système, durant lesquelles, je dois l'admettre, le niveau d'abstraction m'a parfois projeté dans la 4ème dimension.

Je remercie aussi les professeurs Patrick MILLOT du LAMIH (*Laboratoire d'Automatique de Mécanique et d'Informatique industrielles et Humaines*) et Éric BONJOUR professeur au laboratoire ERPI (*Équipe de Recherche sur les Processus Innovatifs*) d'avoir accepté d'être examinateurs de mon jury de thèse.

Je remercie vivement mon directeur de thèse, le professeur Gérard MOREL, pour ses conseils, pour le temps qu'il a su m'accorder au quotidien, les réponses à mes nombreuses questions, le partage de son expérience, mais également pour son perfectionnisme à toute épreuve qui m'aura valu plus de quarante versions de mon introduction de thèse !!! De cette expérience est née une réelle complicité scientifique et humaine, au cours de laquelle nos nombreux échanges diurnes mais

[1] Il est d'usage de rédiger les remerciements, une fois le mémoire et la soutenance soutenue

Remerciements

surtout nocturnes nous aurons repoussés aussi bien l'un que l'autre dans nos retranchements. Merci de m'avoir fait l'honneur d'être le dernier thésard qui clôt trente années d'une longue lignée.

Je tiens également à remercier mon équipe de spécialistes, qui ont partagé avec passion et patience leurs connaissances, et qui m'ont permis d'aboutir aujourd'hui à ce résultat : Jean-Marc DUPONT (*Tu auras réussi à m'initier et à m'intéresser à l'étude des humains à travers tes nombreuses anecdotes et expériences de la vie de tous les jours*), Cédric JOIN (*J'aurais aimé connaitre la CSM durant mes années de licence quand on me demandait d'identifier un système pour son contrôle*), Rémi PANNEQUIN (*le Géo Trouvetout » de la plateforme CISPI, jusqu'à la dernière minute tu m'auras assisté pour la faire fonctionner*), Jean-Pierre HENNEBICQ et Dominique GALARA (*Le partage de vos années d'expériences sur le terrain et en ingénierie, valent bien plus que tout ce que j'ai pu lire par ailleurs*), Frédérique MAYER (*La maitre Capello de la « Situation-Système » et de la « Penser-Système*»), Pascale MARANGE (*Ma grande sœur du laboratoire, pour notre collaboration animée durant la rédaction d'un article en commun, ….. et le gâteau aux 3 chocolats*) et David GOUYON (*Tu m'auras accompagné et guidé dans mon évolution d'étudiant de master à celui de doctorant*).

Au-delà de ces collaborations scientifiques, je tiens à remercier l'ensemble du personnel du CRAN notamment les secrétaires (Anne PIANT, Maryse FERRY, Sabine HURAUX et Christelle KONDRATOW) qui ont toujours facilité au mieux mon parcours au laboratoire, l'équipe informatique, mais aussi la bonne humeur quotidienne des femmes de ménage. Je remercie également l'ensemble de l'équipe de l'AIP Lorrain qui m'a vu évolué aussi bien en tant qu'étudiant que moniteur à l'Université de Lorraine. Je n'oublie pas la société KIASTEK pour m'avoir permis d'utiliser leur outil (Cosimate) et notamment Thierry ROUDIER dont le support technique a été précieux pour la mise en place de mon environnement de co-simulation système. Je remercie également le support de Philippe LEBLANC de la société IBM, qui se poursuit désormais au-delà du contexte de la thèse au travers de mes activités d'ingénieur système R&D chez Airbus Defence and Space à Toulouse. Cela m'amène naturellement à remercier mes collègues d'Airbus pour leurs soutiens et encouragements dans les dernières phases de rédaction du mémoire de thèse et lors de la soutenance.

Néanmoins, cette thèse va bien au-delà de l'aspect professionnel, la réussite de ce travail reviens à un ensemble de personnes m'ayant soutenus au cours de ces quatre dernières années. Je pense dans un premier temps à Alexandre (*Bichon, sans oublier ta Bichonne*) avec qui nous avons partagé le bureau 603B pour le meilleur et pour le pire =), mes compagnons de pauses (*elles étaient nombreuses mais nécessaires*) : Thomas C (*Le stagiaire*), Thomas R (*Ta braguette !!*), Jeremy (*et ses blagues légendaires*), Damien (*combo : Piscine + Kebab*), Florent (*Ma b…, Ma b…).* Les fidèles et moins fidèles de la BSR (*Bureau du Simple Repos : une institution au laboratoire où la convivialité, les plats exotiques, les fous rires, … étaient toujours au RDV*): Sylvain (*Et les anecdotes sur Saint-Quirin*), Gaby (*son smile et le chocolat*), Pierre (*le moniteur devenu collègue, puis amis*), William, Mario (*et ses litres d'huiles d'olives*), Benoit, Éric L (*ça a toujours été un plaisir d'échanger avec toi*) , Éric R, Francis, Thierry, David, Jean-Philippe (*Le vrai Vosgien*), Laetitia, Leïla, Yong Xin, Esma (*La voisine de bureau*), Hervé, Alexandre V, Jean-François, Patrick, Anthonio (*Tu tournes toujours en rond pour réfléchir ??*)…

Remerciements

J'ai également trouvé un équilibre en dehors du cadre du laboratoire, où parfois on me regardait avec de gros yeux lorsque je me lançais dans des explications (*souvent fumeuses*) de mon sujet de thèse, mais où j'ai toujours su y puiser de la motivation pour poursuivre mes travaux. Ainsi, je remercie: Elodie (*La vosgienne, et oui un jour les Vosges domineront le monde*), Isabelle (*Ma poulette, tu me suis depuis le lycée ! Beaucoup de « boulettes » et de chemin parcouru depuis le temps*), Thibaut, Julie et Dawa (*Pagaille, La Normande, Dawabibel … Mine de crayon l'investissement a porté ses fruits, vous avez pris soin de petit ourson devenu grand désormais, il aura fallu tout de même trois Rototom au compteur, Braaaa Braaaa !!!*), Julien (*Bachouss et les soirées du vendredi soir (muffins, taratata, ….), le meilleur remède pour déconnecter en fin de semaine)*, Jean-Baptiste et Ibrahim (*La team Prod-Show, va falloir revoir les flyers : DJ Doc aux platines ^^, que de bonnes soirées pleines de rebondissements*), Romain (*Enfin !!! Tu sais aussi bien que moi ce que veut dire une thèse avec Gérard, content que l'histoire se poursuive à Toulouse*), Lauriane (*Mon élève, l'entrainement continu !!! Merci d'avoir fait le déplacement pour la soutenance*) et Aurélie (*Coupineuuhh !! Finalement tu n'es plus si loin que cela, tu auras été le premier témoin de mon arrivée fracassante dans le sud …*).

Je remercie également grandement mes parents, qui dès nos premiers pas ont tout mis en œuvre pour en arriver là aujourd'hui, aussi bien pour mes frères, ma sœur et moi. Pourquoi faire simple quand on peut faire compliqué, cela n'a pas été de tout repos : quatre enfants à élever, mais le résultat est au RDV. C'est en partie grâce à vous si j'ai pu en arriver là, MERCI. Mélanie, Vincent et Thomas, vous m'avez donné l'envie d'aller jusqu'au bout.

La meilleure pour la fin, je tiens à remercier Roxane (*ma chérie*) qui m'a supporté durant ces quatre dernières années. Cela n'aura pas été facile tous les jours, j'étais souvent derrière mon écran d'ordinateur à ne pas pouvoir t'accorder tout le temps que j'aurais voulu. Cependant, tu as toujours été présente pour me soutenir, me faciliter mon travail, me distraire, me faire rire (*oui même tes blagues de littéraire*) !!! Cette thèse c'est aussi la tienne =) et pas seulement parce que tu as corrigé les fautes d'orthographes, de grammaires et de conjugaisons de ces 299 pages. Désormais, place à de nouvelles aventures mais cette fois-ci ensemble !!

PS : A toi lecteur qui aura été jusqu'au bout des remerciements, j'espère que tu as apprécié, car on passe aux choses sérieuses dans les prochaines pages ! Bonne lecture…

Table des matières

Introduction générale 1

Partie 1 : Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI. 6

Introduction : Interaction collaborative entre domaines d'exploitation et d'ingénierie constitutive de la situation-système d'intérêt ciblée 6

1. Chapitre 1 : Contexte opérationnel de conduite du système à faire CISPI 8

1.1. Introduction : Caractérisation de la situation-système de conduite CISPI 8

1.2. Caractérisation de la situation-système de conduite du procédé industriel critique d'intérêt 8

1.2.1. Caractérisation de la situation dynamique de conduite 8

1.2.2. Caractérisation de la situation cognitive de conduite 10

1.3. Caractérisation de l'architecture-système de conduite du procédé industriel critique d'intérêt 17

1.3.1. Caractérisation de l'architecture existante 17

1.3.2. Caractérisation de l'architecture visée 21

1.4. Conclusion : Plate-forme CISPI constitutive de la situation-système de conduite ciblée 24

2. Chapitre 2 : Contexte d'ingénierie interdisciplinaire du système pour faire CISPI 27

2.1. Introduction : Caractérisation de la situation-système d'ingénierie CISPI 27

2.2. Caractérisation d'une ingénierie de conduite centrée technique 28

2.2.1. Ingénierie système guidée par des processus 28

2.2.2. Ingénierie système basée sur des modèles 32

2.2.2.1. Ingénierie système basée sur les documents versus Ingénierie système basée sur les modèles 32

2.2.2.2. Modélisation d'architecture système 33

2.2.2.3. Virtualisation d'un modèle d'architecture système 36

2.3. Caractérisation d'une ingénierie de conduite centrée humain 39

2.3.1. Intégration Machine-Homme 39

2.3.2. Virtualisation d'un modèle d'intégration Machine-Homme 40

2.4. Caractérisation d'une ingénierie de conduite centrée système 41

2.5. Conclusion : Environnement de modélisation centré sur le système CISPI constitutif de la situation-système d'ingénierie ciblée 42

Conclusion : Environnement collaboratif d'interactions entre domaines d'exploitation et d'ingénierie constitutive de la situation-système d'intérêt ciblée 44

Table des matières

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI 45

Introduction : Interaction collaborative entre domaines d'ingénierie interdisciplinaire constitutive du système d'intérêt ciblé 45

3. Chapitre 3 : Heuristique de spécification-système pour faire CISPI 48

3.1. Introduction : Caractérisation du processus de spécification d'un modèle total de CISPI 48

3.2. Paradigmes de spécification système 48

3.2.1. Paradigme de spécification orientée automatisation 49

3.2.2. Paradigme de spécification orientée problème 50

3.2.2.1 Systèmes techniques 50

3.2.2.1.1 Relation de description-prescription entre espaces problème-solution 51

3.2.2.1.2 Récursivité entre domaines orientés problème-solution 52

3.2.2.1.3 Itérations entre espaces problème-solution 53

3.2.2.2 Systèmes sociotechniques 54

3.3. Paradigmes de modélisation système 56

3.3.1. Fondements holoniques 56

3.3.2. Fondements systémiques 59

3.4. Processus de modélisation holonique 60

3.5. Conclusion : Heuristique holonique constitutive du processus de spécification-système ciblé 65

4. Chapitre 4 : Environnement de spécification-système exécutable pour faire CISPI 66

4.1. Introduction : Caractérisation de l'environnement de spécification-système exécutable CISPI 66

4.2. Cadre de modélisation architecturale 66

4.2.1. Cadre de modélisation architecturale CMMS-IAMS 67

4.2.2. Cadre de modélisation architecturale SIAC 69

4.2.3. Cadre de modélisation architecturale Système 70

4.3. Environnement de co-modélisation et co-simulation exécutable 73

4.3.1. Langage d'interopération système 73

4.3.2. Bus de co-spécification système 75

4.3.3. Bus de co-simulation système 77

4.4. Scénario d'une co-spécification exécutable d'un modèle total du système de conduite CISPI 80

4.4.1. Co-spécification exécutable d'un modèle partiel d'un constituant physique 82

4.4.2. Co-spécification exécutable d'un modèle partiel d'un constituant logique 85

4.4.3. Co-spécification exécutable d'un modèle partiel d'un constituant technique d'actionnement 86

4.4.4. Co-spécification exécutable d'un modèle partiel d'un constituant physiologique de mesurage 88

4.5. Conclusion : Environnement collaboratif constitutif du processus de spécification-système exécutable ciblé 89

Conclusion : Co-spécification d'un modèle total constitutif du système d'intérêt ciblé 90

Table des matières

Partie 3 : Co-spécification exécutable de constituants de conduite du procédé du système CISPI 92

Introduction : Interaction collaborative entre domaines d'ingénierie interdisciplinaire centrée physique et centrée conduite 92

5. Chapitre 5 : Co-spécification exécutable d'un constituant physique du système de conduite CISPI 94

5.1. Introduction : Interaction physique ciblée du système CISPI 94

5.2. Co-spécification système d'un modèle d'interaction physique de modulation de débit d'eau 94

5.2.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction physique de modulation de débit d'eau 94

5.2.2. Spécification prescriptive d'un modèle comportemental constitutif d'une interaction physique de modulation de débit d'eau 96

5.2.2.1. Eléments de modélisation d'une interaction physique 96

5.2.2.2. Définition d'un modèle comportemental de l'interaction physique ciblée 97

5.2.3. Co-intégration système d'un modèle comportemental constitutif d'une interaction physique sûre de modulation de débit d'eau 100

5.3. Co-spécification système d'un modèle exécutable d'interaction multi-physique de modulation de débit d'eau.....103

5.3.1. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction multi-physique de modulation de débit d'eau 104

5.3.1.1. Eléments de connaissances en modélisation processive d'une interaction multi-physiques.....104

5.3.1.2. Construction d'un modèle multi-physique exécutable de l'interaction ciblée 112

5.3.1.3. Traduction d'un modèle exécutable constitutif de l'interaction multi-physiques ciblée 115

5.3.1.3.1. Modèle structurel multi-physiques 116

5.3.1.3.2. Modèle comportemental multi-physiques.....119

5.3.2. Co-intégration système d'un constituant d'interaction multi-physique de modulation de débit d'eau 121

5.4. Conclusion : Co-spécification exécutable d'un modèle partiel constitutif de la partie physique ciblée.....125

6. Chapitre 6 : Co-spécification exécutable d'un constituant logique du système de conduite CISPI 126

6.1. Introduction : Interaction logique ciblée du système CISPI 126

6.2. Co-Spécification système d'un modèle exécutable constitutif d'une interaction de contrôle de modulation de débit d'eau.....128

6.2.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction de contrôle de modulation de débit d'eau 128

6.2.2. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction de contrôle de modulation de débit d'eau 130

6.2.2.1. Eléments de connaissances d'une interaction de contrôle 130

6.2.2.2. Construction d'un modèle exécutable de contrôle de l'interaction ciblée.....132

6.2.2.3. Traduction d'un modèle exécutable de contrôle de l'interaction ciblée.....135

Table des matières

6.2.3. Co-intégration d'un modèle exécutable constitutif d'une interaction de contrôle de modulation de débit d'eau 137

6.3. Co-spécification exécutable d'un modèle constitutif d'une interaction de commande de modulation de débit d'eau 140

6.3.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction de commande de modulation de débit d'eau 140

6.3.2. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction de commande de modulation de débit d'eau 142

6.3.2.1. Eléments de connaissances d'une interaction de commande 142

6.3.2.2. Définition d'un modèle exécutable de commande de l'interaction ciblée 143

6.3.3. Co-Intégration système d'un modèle constitutif d'une interaction de commande de modulation de débit d'eau 151

6.4. Conclusion : Co-spécification exécutable d'un modèle partiel constitutif de la partie logique ciblée 153

Conclusion : Co-spécification d'un modèle total constitutif de l'interaction physico-logique ciblée 154

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage du système de conduite CISPI 155

Introduction : Interaction collaborative entre domaines d'ingénierie interdisciplinaire centrée technique et centrée physiologique 155

7.Chapitre 7 : Co-spécification exécutable d'un constituant technique d'actionnement du système de conduite CISPI 158

7.1. Introduction : Partie « actionnement intelligent » du système de conduite CISPI 158

7.2. Co-spécification système d'un modèle d'interaction logico-physique de modulation de débit 159

7.2.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction physique de modulation de débit 160

7.2.1.1. Éléments de connaissances d'une interaction physique 160

7.2.1.2. Définition des exigences de l'interaction physique ciblée 161

7.2.2. Spécification prescriptive d'un modèle d'exigences constitutif d'une interaction logique de modulation de débit 163

7.2.2.1. Eléments de connaissances d'une interaction logique 163

7.2.2.2. Définition des exigences de l'interaction logique ciblée 163

7.2.3. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction logico-physique de modulation de débit 165

7.2.3.1. Eléments de modélisation exécutable d'une interaction logico-physique 165

7.2.3.2. Définition d'un modèle exécutable de l'interaction logico-physique ciblée 170

7.3. Co-intégration système d'un constituant d'interaction logico-physique de modulation de débit 172

7.3.1. Intégration d'un modèle constitutif de l'interaction ciblée 172

7.3.2. Intégration d'un composant constitutif de l'interaction ciblée 177

7.4. Conclusion : Co-spécification exécutable d'un modèle partiel constitutif de la partie « actionnement intelligent » ciblée 180

Table des matières

8. Chapitre 8 : Co-spécification exécutable d'un constituant physiologique de mesurage du système de conduite CISPI *181*

8.1. Introduction : Partie « mesurage intelligent » du système de conduite CISPI *181*

8.2. Co-spécification système d'un modèle d'interaction physico-physiologique de détection sonore *183*

8.2.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction sensorielle de détection sonore *184*

8.2.1.1. Eléments de connaissances d'une interaction sensorielle *184*

8.2.1.2. Définition des exigences de l'interaction sensorielle ciblée *185*

8.2.2. Spécification prescriptive d'un modèle d'exigences constitutif d'une interaction physiologique de détection sonore *186*

8.2.2.1. Eléments de connaissances d'une interaction physiologique *186*

8.2.2.2. Définition des exigences de l'interaction physiologique ciblée *189*

8.2.3. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction physico-physiologique de détection sonore *193*

8.2.3.1. Eléments de modélisation exécutable d'une interaction physico-physiologique *193*

8.2.3.2. Définition d'un modèle exécutable de l'interaction physico-physiologique ciblée *198*

8.3. Co-intégration système d'un constituant d'interaction physico-physiologique de détection sonore *199*

8.3.1. Intégration d'un modèle constitutif de l'interaction physico-physiologique ciblée *200*

8.3.2. Intégration d'un composant constitutif de l'interaction physico-physiologique ciblée *203*

8.4. Conclusion : Co-spécification exécutable d'un modèle partiel constitutif de la partie « mesurage intelligent » ciblée *207*

<u>Conclusion</u> : Co-spécification d'un modèle total constitutif de l'interaction logico-physico-physiologique ciblée ***208***

<u>Conclusion et perspectives générales</u> ***209***

Bibliographie ***213***

Annexes ***220***

Table des figures

Figure 1 : Interprétation du diagramme de couplage système (HW Lawson, 2010) 2
Figure 2 : Systémigramme explicatif des relations TOUT et PARTIE de notre heuristique de co-spécification exécutable basée sur des modèles d'un système d'intérêt 3
Figure 3 : Proposition d'un environnement de co-spécification système exécutable basée sur des modèles.....4
Figure 4 : Domaine d'ingénierie et d'exploitation d'un système de conduite (Devic & Morilhat, 2013).....6
Figure 5 : Cadre d'analyse d'une situation-système de conduite 8
Figure 6: Principaux circuits de refroidissement d'un REP (adaptée de (Duco, 2004)).....10
Figure 7 : Système de conduite d'un procédé en exploitation 11
Figure 8 : Interaction physico-technico physiologiques de conduite.....12
Figure 9 : Principe de l'Approche par Etat (Appell & Chambon, 1998).....13
Figure 10 : Conduite du procédé nucléaire vue comme une évolution entre états stables à travers le temps (Galara & Hennebicq, 1999).....15
Figure 11 : Représentation de la situation d'états du système ASG à différents niveaux d'abstractions (Viel, Cheriaux, & Galara, 2012) 15
Figure 12 : Modèle de la conduite par un opérateur d'une situation dynamique (procédé nucléaire) (Vicente, Mumaw, & Roth, 2004).....16
Figure 13 : Schéma d'exécution du mode opératoire de « mise en brassage de la Bâche ASG » (Viel, Cheriaux, & Galara, 2012) 17
Figure 14 : Organisation du système de conduite.....18
Figure 15 : Distribution des fonctions et des opérateurs de conduite autour d'un bus d'objet métier 21
Figure 16 : Concept architectural d'un système de conduite 22
Figure 17 : Afin de pouvoir augmenter les capacités des vannes manuelle (OMT) un certain nombre de fonctions et de traitements (OLT) sont nécessaires, allant de la connexion physique des signaux de la vanne jusqu'à la communication (sans fil) des informations associées (Dobre, 2010) 23
Figure 18 : Architecture multi-agents du SIAC (vert = agent disponible, bleu = agent en exécution, rouge = agent en exclusion) (Dobre, 2010) 24
Figure 19 : Processus physique de la plate-forme CISPI à conduire (hydraulique) – Emulation du procédé ASG (Alimentation Secours des Générateurs de Vapeur) 25
Figure 20 : Salle de commande de la plate-forme CISPI du CRAN 25
Figure 21 : Système de conduite d'un procédé en exploitation, adaptée de (Galara & Hennebicq, 1999).....27
Figure 22 : Domaines couverts par les trois normes générales de l'ingénierie système (Fiorèse & Meinadier, 2012) 29
Figure 23 : Cartographie des processus du cycle de vie du système (ISO 15288) (Fiorèse & Meinadier, 2012).29
Figure 24 : Relations entre l'ingénierie du système et l'ingénierie des technologies (Faisandier, 2014) 31
Figure 25 : Itération et récursivité des processus de définition (Faisandier, 2014) 31
Figure 26 : Vision de l'INCOSE pour 2020 – Transition de l'ingénierie système basée sur les documents vers une ingénierie basée sur les modèles (INCOSE, 2007).....32
Figure 27 : Grille schématique de la représentation des systèmes (approche traditionnelle) (Fiorèse & Meinadier, 2012).....35
Figure 28 : Organisation générique d'un modèle « total » du système (Krob, 2012) 35
Figure 29 : Vision classique d'un système formel (Krob, 2012) 36
Figure 30 : Modèle d'intention (en vert) constitutif d'un modèle architectural multi-physique exécutable (Retho, 2015) 37
Figure 31 : Evolution interdisciplinaire de l'Ingénierie Système Basée sur des Modèles (Blochwitz, et al., 2012) 38

Table des figures

Figure 32 : Répartition des tâches de conduite entre des agents techniques ou humains (Millot, Debernard, & Vanderhaegen, 2011) 40
Figure 33 : Modèle d'un processus d'évaluation d'une situation (Endsley, 1995) 42
Figure 34 : Exemple d'une DSA (Distributed Situation Awareness) (P Salmon, et al., 2008) 42
Figure 35 : Environnement collaboratif d'ingénierie centrée système basée sur des modèles exécutables 43
Figure 36 : Environnement collaboratif de co-intégration système 44
Figure 37 : Environnement de co-spécification exécutable basée sur des modèles 47
Figure 38 : Processus de spécification concourante d'un SYSTEME automatisé 50
Figure 39 : Modèle de référence pour les exigences et les spécifications adaptée de (Gunter, et al., 2000) 50
Figure 40 : Partition des domaines autour du domaine du SYSTEME à faire (Dobre, 2010) 52
Figure 41 : Processus itératif, récursif et concourant de co-spécification en ingénierie de SYSTEME automatisé 53
Figure 42 : Twin Peaks model – itération entre les espaces problème et solution. Adaptée de (Hall, et al., 2002) 54
Figure 43 : Modèle de référence d'un SYSTEME sociotechnique. Adaptée de (Hall & Rapanotti, 2005) 54
Figure 44 : Proposition d'une organisation d'ingénierie d'un $\text{SYSTEME}_{\text{Homme-Machine}}$ au travers du processus de co-spécification SYSTEME. Adaptée de (Lieber, 2013) 55
Figure 45 : Vision moléculaire d'une construction SYSTEME (Mayer, 1995) 56
Figure 46 : Patron essentiel en formalisme FOSM (Fact Oriented System Modeling) (OMG) (G. Morel, et al., 2013) 56
Figure 47 : Systémigramme de définition multi-échelle d'un SYSTEME d'intérêt 58
Figure 48 : Le système vu comme un holon est composé d'un noyau interne et d'une couche externe en « communication » permanente (Boardman, Sauser, & Verma, 2009) 59
Figure 49 : Interprétation du « Conceptagon » (Boardman, Sauser, John, & Edson, 2009) selon le paradigme holonique 60
Figure 50 : Formalisation du processus de spécification 63
Figure 51 : Processus de co-spécification SYSTEME distribué autour d'un bus 64
Figure 52 : Méthodologie pour la définition du problème selon une vision systémique (Boardman & Sauser, 2008) 65
Figure 53 : Le Module Fonctionnel d'Automatisme dans le cadre CMMS-IAMS. (Pétin, Iung, & Morel, 1998) 68
Figure 54 : Exemple d'expression des exigences pour un agent de maintenance avec l'outil PRIAM 68
Figure 55 : Paradigme architectural de Système Interactif d'Aide à la Conduite (Dobre, 2010) 69
Figure 56 : Patron architectural logico-fonctionnel CMMS-IAMS proposé 71
Figure 57 : Types de Diagrammes SysML) (OMG, 2012a) 74
Figure 58 : Organisation de l'espace "Quickplace" par domaines d'ingénieries spécialistes 76
Figure 59 : Utilisation du bus de co-simulation tout au long du cycle de vie SYSTEME 78
Figure 60 : Environnement de co-modélisation de la plate-forme CISPI 78
Figure 61 : Couplage du modèle de l'environnement au bus de co-simulation 79
Figure 62 : Patron Architectural CMMS logico-fonctionnel visé 80
Figure 63 : Patron architectural CMMS-IAMS logico-fonctionnel visé 81
Figure 64 : Patron architectural CMMS-IAMS logique visé 82
Figure 65 : Formalisation du problème (P_S) sous la forme d'exigence (R_{S_SysML}) selon la connaissance (K_S) du domaine d'ingénierie opérationnelle 83
Figure 66 : Environnement de co-spécification du procédé de la plate-forme CISPI selon le patron architectural de la partie physique (Figure 64) 83
Figure 67 : Environnement de co-spécification du contrôle-commande de la plate-forme CISPI selon le patron architectural de la partie logique (Figure 64) 85

Table des figures

Figure 68 : Environnement de co-spécification de la plate-forme CISPI selon le patron architectural du système CMMS-IAMS visé (Figure 64) 86
Figure 69 : Environnement de co-spécification d'une fonction d'actionnement allouée à un agent technique selon le patron architectural d'un actionneur intelligent (Figure 64) 87
Figure 70 : Environnement de co-spécification d'une fonction de mesure allouée à un agent humain le patron architectural d'un mesureur intelligent (Figure 64) 88
Figure 71 : Impact du patron de modélisation systémique « Conceptagon » sur le patron architectural CMMS-IAMS visé 90
Figure 72 : Patron Architectural CMMS visé en partie 3 92
Figure 73 : Diagramme de séquence SysML descriptif du scénario de co-spécification système visé en partie 3 93
Figure 74 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence opérationnelle ($R_{IO\ 1.3}$) par l'espace-solution d'ingénierie système 95
Figure 75 : Modèle orienté flux du procédé CISPI (S_{IPro}) sous forme de schéma graphique standardisé à un niveau général 96
Figure 76 : Modèle de procédé orienté flux (S_{IPro}) sous forme de schéma graphique standardisé à un niveau de détail selon la norme (ISO 10628, 2014) 97
Figure 77 : Modèle orienté flux du procédé CISPI sous forme de diagramme d'activité (act) en langage SysML 98
Figure 78 : Traçabilité entre la *spécification* descriptive d'ingénierie système (Exigences - R_{IS_Pro}) et la *spécification* prescriptive d'ingénierie du procédé (Actions - S_{IPro_SysML}) 100
Figure 79 : Exigence $R_{IS1.5\text{-}0}$: Critère de Défaillance Unique 100
Figure 80 : Diagramme d'état (stm) SysML décrivant le « testCase » pour la validation des exigences ($R_{IS1.6}$ et $R_{IS1.7}$) 102
Figure 81 : Environnement de co-simulation pour la validation de la spécification du procédé (S_{IPro_SysML}) 102
Figure 82 : Exigence $R_{IS1.7\text{-}0.1}$: Débit d'alimentation en eau de secours 103
Figure 83 : Tétraèdre d'état de la théorie Bond-Graph (Paynter, 1961) 104
Figure 84 : Système de transformation induit par le tétraèdre d'état sur les parties de l'environnement constituées des flux d'énergies et des flux de puissances (Feliot, 1997) 107
Figure 85 : Le référentiel T, E, F (Temps, Espace, Forme) permettant de repérer la position des objets soumis à des processus (Décomposition triadique des processus) (Le Moigne, 1994). 107
Figure 86 : Exemple d'une séquence équivalente à un processus de type F. Adaptée de (Feliot, 1997) 109
Figure 87 : Exemple d'une séquence équivalente à un processus de type T. Adaptée de (Feliot, 1997) 110
Figure 88 : Exemple d'une séquence équivalente à un processus de type E. Adaptée de (Feliot, 1997) 110
Figure 89 : Modèle orienté flux énergétique du procédé CISPI sous forme de diagramme de bloc en langage Modelica 114
Figure 90 : Exigence $R_{IS1.7\text{-}0.1}$: Performance requise du procédé relative au débit d'alimentation 114
Figure 91 : Exigence prescrite par l'*espace-solution* ingénierie du procédé ($S_{IPro1.0}$) : Hauteur cuve stockage 115
Figure 92 : Diagramme de définition de bloc (bdd) SysML - Composition du processus CISPI 116
Figure 93 : FlowSpecification de type Puissance 117
Figure 94 : Diagramme interne de bloc SysML – Structure du processus CISPI 118
Figure 95 : Diagramme paramétrique SysML – Dynamique du processus T 120
Figure 96 : Interfaces du modèle Modelica du procédé connectées à des ports "Cosimate" (en orange) 122
Figure 97 : Première itération - Définition de l'architecture de co-simulation système (*Sous-système physique*) par l'espace-problème d'ingénierie système 122
Figure 98 : Interface de supervision du déroulement du "TestCase_1.6_1.7" définit dans un "Panel Diagram" Rhapsody 123

Table des figures

Figure 99 : Environnement de co-simulation système pour la validation de la spécification du processus de la plate-forme CISPI (S_{IPro_SysML}) 123
Figure 100 : Trace d'exécution du « *TestCase_1.6_1.7* » en co-simulation pour s'assurer de la satisfaction des exigences ($R_{IS1.6}$ et $R_{IS1.7}$) 124
Figure 101 : Diagramme d'exigences ($R_{IS_Aut\text{-}Cont}$) relatives à la fonction de contrôle de régulation en puissance d'alimentation en eau de secours 129
Figure 102 : Calcul de la commande en se fondant sur des outils d'estimation algébriques rapides (modèle ultra-locale) et non asymptotiques, sans démarche d'identification classique et décorrélée du procédé à contrôler. 131
Figure 103 : Mise en œuvre systémique de la CSM selon le paradigme architectural CMMS 132
Figure 104 : Boucle de rétroaction en puissance contrôlée par une fonction de conduite intelligente intégrant la CSM 133
Figure 105 : Diagramme fonctionnel hiérarchique Simulink ® d'une fonction de contrôle 134
Figure 106 : Description en vision boite blanche des blocs fonctionnels composant le modèle de contrôle de la partie logique de la plate-forme CISPI 134
Figure 107 : Diagramme de définition de blocs (bdd) - Architecture de la fonction de contrôle 136
Figure 108 : Diagramme interne de bloc (ibd) - Structure interne de la fonction de contrôle 136
Figure 109 : Environnement de co-simulation système pour la validation de la spécification ($S_{IAut\text{-}Cont}$) d'une fonction de contrôle du processus de la plate-forme CISPI 137
Figure 110 : Diagramme interne de blocs SysML (ibd) - Définition des interactions entre les modèles de contrôle et physique de la plate-forme CISPI pour la validation par co-exécution de modèles de la fonction de régulation en puissance à travers la voie 1 138
Figure 111 : Trace d'exécution modèle Simulink – Scénario test 1 (Dépassement Puissance Max) 139
Figure 112 : Traces d'exécution modèle Simulink – Scénario test 2 (Détection Défaut Global) 140
Figure 113 : Exigences ($R_{IS_Aut\text{-}Comm}$) relatives à la fonction de commande d'alimentation en eau de secours de la plate-forme CISPI 141
Figure 114 : Diagramme de séquence SysML – Scénario d'alimentation en eau de secours du circuit primaire par la plate-forme CISPI au travers de la voie 1 ou de la voie 2 142
Figure 115 : Diagramme de définition de blocs (bdd) - Architecture de la fonction de commande « *Uc_Alimenter_Eau_Secours* » 143
Figure 116 : Diagramme d'états SysML - Comportement du bloc fonctionnel : "*Acter*" 145
Figure 117 : Diagramme d'états SysML - Comportement du bloc fonctionnel : "*Commander*" 146
Figure 118 : Diagramme d'états SysML - Comportement du bloc fonctionnel : "*Rendre Compte*" 147
Figure 119 : Diagrammes d'états SysML - Modèles comportementaux des acteurs externes du sous-système de commande 148
Figure 120 : Diagramme interne de blocs SysML - Interactions entre le sous-système de commande et le système extérieur (*Circuit_Primaire + Sous_Systeme_Controle*) 149
Figure 121 : Environnement de co-simulation système pour la validation de la spécification ($S_{IAut\text{-}Comm}$) d'une fonction de commande de la plate-forme CISPI 151
Figure 122 : Seconde itération - Définition de l'architecture de co-simulation système (*Sous-système logique + Sous-système physique*) par l'espace-problème d'ingénierie système 151
Figure 123 : Trace d'exécution de la situation système de conduite d'alimentation en eau de secours Conclusion 152
Figure 124 : Patron architectural CMMS-IAMS logique visé 155
Figure 125 : Répartition de la fonction IAMS entre un agent technique d'actionnement intelligent et un agent humain de mesurage intelligent 156
Figure 126 : Diagramme de séquence (seq) SysML - Scénario de co-spécification d'un système d'actionnement et de mesurage intelligent 157

Table des figures

Figure 127 : Patron architectural d'Actionnement Intelligent par un constituant technique 159
Figure 128 : Principe d'un constituant technique d'actionnement intelligent (G. Morel, 1999) 160
Figure 129 : Processus C de modulation en puissance en sortie de la voie 1 d'alimentation en eau de secours 161
Figure 130 : Vanne de régulation et débitmètre de la voie 1 de la plate-forme CISPI.....161
Figure 131 : Modèle d'exigences système ($R_{IS_FAI_SysML}$) relatives au constituant technique de modulation de débit 162
Figure 132 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence système (R_{IS_FAI4}) par l'espace-solution fournisseur d'actionneur intelligent 164
Figure 133 : Interprétation du patron architectural d'Actionnement Intelligent pour la modulation en débit à travers la voie 1 de la plate-forme CISPI 166
Figure 134 : Modèle (Simulink) des blocs : « interaction Logico-Technologique », « Acter », « Percevoir », « Agir » et « Rendre-Compte » composant le constituant d'actionnement intelligent.....167
Figure 135 : Modèle logique (Modelica) du bloc : « interaction technico-physique » composant le constituant d'actionnement intelligent.....167
Figure 136 : Environnement de co-simulation pour la vérification de la spécification (S_{FAI}) du constituant d'actionnement intelligent de modulation de débit.....168
Figure 137 : Traces d'exécution de la co-simulation entre le modèle Simulink et Dymola du constituant d'actionnement intelligent.....169
Figure 138 : Raffinement de l'exigence $S_{IPro1.0}$ à partir de la connaisse K_{FAI4} de l'espace-solution fournisseur d'actionnement intelligent.....170
Figure 139 : Diagramme de définition de blocs (bdd) - Architecture externe du constituant d'actionnement intelligent de modulation de débit 171
Figure 140 : Diagramme interne de bloc (ibd) - Structure interne du constituant d'actionnement intelligent de modulation de débit 171
Figure 141 : Environnement de co-simulation système pour la validation de la spécification (S_{FAI}) du constituant d'actionnement intelligent de modulation de débit.....172
Figure 142 : Modèle logique du processus CISPI intégrant la partie physique du constituant d'actionnement intelligent 173
Figure 143 : Spécification (S_{IPhy_SysML}) des interfaces externes du processus CISPI à un niveau d'abstraction logique 173
Figure 144 : Définition de l'architecture de co-simulation système (*Bloc d'interaction de détection sonore + bloc logique + bloc physique*) par l'espace-problème d'ingénierie système.....174
Figure 145 : Trace d'exécution du scénario de test pour la validation du constituant d'actionnement intelligent de modulation de puissance 175
Figure 146 : Environnement de co-validation système pour la validation et l'évaluation du niveau de maturité de la spécification (S_{FAI}) du constituant d'actionnement intelligent de modulation de débit 176
Figure 147 : Trace d'exécution du constituant d'actionnement intelligent de modulation de puissance.....177
Figure 148 : Réalisation de la partie logique du constituant d'actionnement intelligent en tant qu'agent de la plate-forme JADE implémentant le SIAC.....178
Figure 149 : Intégration du composant d'actionnement intelligent avec le SIAC de la plate-forme CISPI 179
Figure 150 : Patron architectural de Mesurage Intelligent par un Agent Humain.....181
Figure 151 : Interaction logico-physiologique de détection sonore 183
Figure 152 : Modèle d'exigences système ($R_{IS_Phy_SysML}$) relatives au constituant sensoriel de mesurage intelligent 185
Figure 153 : Vue d'ensemble des processus physiologiques de détection sonore constitutifs d'un ensemble d'interactions fonctionnelles (Source/Puits) 186

Table des figures

Figure 154 : Interaction Fonctionnelle ψ entre la source u'(r'), unité de structure à la position r', et le puits u(r), unité de structure à la position r. Puits et source peuvent devenir tour à tour source ou puits (Chauvet, 1995). 188
Figure 155 : Modèle explicatif de l'interaction fonctionnelle : Analogie entre la représentation de Chauvet et un schéma électrique montrant l'importance de la stimulation d'un certain potentiel d'action pour ouvrir les « portes » de propagation du flux d'interaction physico-physiologique (Lieber, et al., 2013) 188
Figure 156 : Plages typiques du champ auditif humain 190
Figure 157 : Exigence physiologique ($R_{IPhy-1.1}$) : plage de perception sonore chez l'humain 190
Figure 158 : Exigence ($R_{IPhy-1.2}$) : Durée minimale alarme sonore 191
Figure 159 : Exigence physiologique ($R_{IPhy-1.3}$) : Alignement anthropométrique entre les constituants technique « source » et humain « puits » 191
Figure 160 : Exigence physiologique (R_{IPhy-2}) : Intensité signal sonore – bruits de fonds 192
Figure 161 : Exigence technique (R_{IPhy-2}) : Intensité signal sonore – bruits de fonds 193
Figure 162 : Bloc Simulink – constituant physiologique « puits » de détection sonore 194
Figure 163 : Bloc Simulink – Interaction fonctionnelle de propagation sonore 195
Figure 164 : Bloc Simulink – bloc technique « source » d'émission sonore 195
Figure 165 : Diagramme Simulink de l'interaction de détection sonore entre un constituant technique « source » et un constituant physiologique « puits ». 196
Figure 166 : Puissance sonore (W) en fonction de la distance « Source/Puits » 196
Figure 167 : Détection de l'alarme sonore en fonction de la fréquence du signal 197
Figure 168 : Détection de l'alarme en fonction du bruit de fond 197
Figure 169 : Diagramme de définition de blocs (bdd) - Architecture externe de l'interaction de détection sonore 198
Figure 170 : Diagramme interne de bloc (ibd) - Structure interne de l'interaction de détection sonore 199
Figure 171 : Environnement de co-simulation système pour la validation de la spécification (S_{IPhy}) d'une interaction de détection sonore 200
Figure 172 : Définition de l'architecture de co-simulation système (*Bloc d'interaction de détection sonore + bloc logique + bloc physique*) par l'espace-problème d'ingénierie système 201
Figure 173 : Trace d'exécution du scénario 1 de validation de l'interaction de détection sonore 202
Figure 174 : Trace d'exécution du scénario 2 de validation de l'interaction de détection sonore 203
Figure 175 : Interface de la maquette physiologique d'interaction Physico-Physiologique (configuration de l'interaction de détection sonore d'une alarme technique) 205
Figure 176 : Maquette physico-physiologique en interface avec la plate-forme CISPI 205
Figure 177 : Généralisation de l'heuristique de co-spécification système étudiée 210
Figure 178 : The System Thinker's Tree (Smith & Daniel-Allegro, 2015) 211
Figure 179 : Interprétation du paradigme systémique par (Galara, 1995) 221
Figure 180 : Spécification des interfaces d'une fonction AMI avec l'outil PRIAM 222
Figure 181 : Exemple de listes de requêtes et de comptes rendus (PRIAM) 223
Figure 182 : Spécification comportementale d'un AMI en langage FBD (Bloc Fonctionnel) avec l'outil SPEX 224
Figure 183 : Simulation du modèle comportementale réalisée sur l'outil SPEX 225
Figure 184 : Structuration de l'espace "Quickplace" de co-spécification de la plate-forme CISPI 226
Figure 185 : Structuration d'un domaine d'ingénierie en espaces problème et solution et connaissances 227
Figure 186 : Structuration de la salle "Co-Spécification" en dossier de spécification entre espaces problème et solution 228
Figure 187 : Diagramme d'exigence SysML - Exigences décrites par l'*espace problème* d'ingénierie opérationnelle (R_{IO_SysML}) 229
Figure 188 : Diagramme d'exigences SysML - Raffinement de l'exigence opérationnelle (*RIO 1.5*) par l'*espace-solution* d'ingénierie système 230

Table des figures

Figure 189 : Diagramme d'exigences SysML - Raffinement de l'exigence opérationnelle (*RIO 1.6*) par l'*espace-solution* d'ingénierie système 231
Figure 190 : Diagramme d'exigences SysML - Raffinement de l'exigence opérationnelle (*RIO 1.7*) par l'*espace-solution* d'ingénierie système 232
Figure 191 : Diagramme d'exigences SysML - Exigences et problèmes systèmes ($R_{IS_Pro_SysML}$) relatives au procédé 233
Figure 192 : Spécification des fonctions de transport par les voies V1 et V2 entre la plate-forme CISPI et son environnement 234
Figure 193 : Spécification de la dynamique des fonctions de transport par les voies V1 et V2 selon une fonction identité (*l'entrée est égale à la sortie*)....234
Figure 194 : Spécification du procédé CISPI à un niveau fonctionnel (Point de vue Structurel)....235
Figure 195 : Spécification de flux non atomique (Flow Specification) 243
Figure 196 : Diagramme de définition de blocs (bdd) SysML - "Constraint blocks " utilisés pour la définition du comportement des processus A,T,C 244
Figure 197 : Diagramme de définition de blocs (bdd) SysML - Composition des processus Accumulation, Transport, Conversion du processus CISPI 245
Figure 198 : Diagramme paramétrique (par) SysML – Dynamique processus A....246
Figure 199 : Diagramme paramétrique (par) SysML – Dynamique processus C....247
Figure 200 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle de la partie physique de la plate-forme CISPI 248
Figure 201 : Configuration de l'environnement de co-simulation pour la validation du modèle de la partie physique de la plate-forme CISPI : définition des interactions entre les modèles spécialistes au travers des ports de type "Cosimate" selon le diagramme interne de blocs (Figure 200). 249
Figure 202 : Modèle du processus de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle de contrôle de la partie-logique de la plate-forme CISPI 250
Figure 203 : Modèle de la partie contrôle de la plate-forme CISPI avec des ports de type "Cosimate"....250
Figure 204 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle de contrôle de la partie logique de la plate-forme CISPI 251
Figure 205 : Configuration de l'environnement de co-simulation pour la validation du modèle de contrôle de la partie logique de la plate-forme CISPI : définition des interactions entre les modèles spécialistes au travers des ports de type "Cosimate" selon le diagramme interne de blocs (Figure 204)....252
Figure 206 : Structure interne du modèle de commande....252
Figure 207 : Modèle de l'environnement de la plate-forme CISPI (circuit primaire) pour la validation du modèle de commande 253
Figure 208 : Modèle de la partie contrôle de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle de commande 253
Figure 209 : Modèle du processus de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle de commande 254
Figure 210 : Panel Diagram - Interface de supervision avec les modèles de commande et des acteurs....254
Figure 211 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle de commande de la partie logique de la plate-forme CISPI....255
Figure 212 : Configuration de l'environnement de co-simulation pour la validation du modèle de commande de la partie logique de la plate-forme CISPI : définition des interactions entre les modèles spécialistes au travers des ports de type "Cosimate" selon le diagramme interne de blocs (Figure 204) 256
Figure 213 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence système « R_{IS_FAI1}» par l'espace-solution fournisseur d'actionneur intelligent 257
Figure 214 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence système « R_{IS_FAI2}» par l'espace-solution fournisseur d'actionneur intelligent 258

Table des figures

Figure 215 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence système « R_{IS_FAI3}» par l'espace-solution fournisseur d'actionneur intelligent 258
Figure 216 : Modèle Modelica de la partie technique du constituant d'actionnement intelligent pour la vérification de la spécification (S_{FAI}) du constituant d'actionnement intelligent 260
Figure 217 : Modèle Simulink de la partie logique du constituant d'actionnement intelligent pour la vérification de la spécification (S_{FAI}) du constituant d'actionnement intelligent 260
Figure 218 : Configuration de l'environnement de co-simulation pour la vérification du modèle du constituant d'actionnement intelligent de modulation de débit 261
Figure 219 : Modèle de l'environnement de la plate-forme CISPI (circuit primaire) pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit 262
Figure 220 : Modèle du sous-système physique de la plate-forme CISPI (*incluant la partie technique du constituant d'actionnement intelligent*) avec des ports de type "Cosimate" pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit 262
Figure 221 : Modèle de la partie logique du constituant d'actionnement intelligent pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit 263
Figure 222 : Modèle de la partie commande du sous-système logique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit 263
Figure 223 : Modèle de la partie contrôle du sous-système logique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit 264
Figure 224 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit 265
Figure 225 : Modèle ORM de l'interaction de perception pour agir en contexte. L'objet interface affordant (F1 & F2) au centre de la relation de perception (F3 &F4) représente l'élément synergique entre objets hétérogènes techniques et humains (F5) qu'il faut arrivé à mesurer (Cette nécessité de mesure est représentée par l'affordance dans l'objet interface) 269
Figure 226 : Modèle ORM de l'interaction de détection sonore entre un objet technique et un opérateur humain, envisagée comme la mise en correspondance d'un percept, son de l'alarme, et d'une propriété cognitive ou connaissance de l'opérateur (Procédure incidentelle ou accidentelle) 270
Figure 227 : Modèle ORM de l'atome élémentaire d'interaction fonctionnelle d'un processus physiologique. L'interaction fonctionnelle émerge de la relation de stimulation entre deux unités structurales et est prescrite in fine par le produit émis par la source u1 ou reçu par le puis u2 qui traverse des discontinuités structurales. De telles propriétés des mécanismes physiologiques conduisent à une structuration de l'organisme vivant en niveaux d'organisation hiérarchique et de ses unités structurales (architecture organique), ainsi que de ces processus physiologiques (architecture fonctionnelle) 271
Figure 228 : Traçabilité entre exigences physiologiques (R_{IPhy}) et exigences systèmes (R_{IS}).....272
Figure 229 : Modèle de l'environnement de la plate-forme CISPI (circuit primaire) pour la validation du modèle d'interaction de détection sonore 274
Figure 230 : Modèle de la partie contrôle du sous-système logique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle d'interaction de détection sonore 274
Figure 231 : Modèle de la partie commande du sous-système logique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle d'interaction de détection sonore 275
Figure 232 : Modèle du sous-système physique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle d'interaction de détection sonore 275
Figure 233 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle d'interaction de détection sonore de la plate-forme CISPI.....276

Abréviations

AFIS *= Association Française d'Ingénierie Système*
AFNOR *= Association Française de NORmalisation*
AgT *= Agent de Terrain*
AIPL *= Atelier Inter établissement de Productique Lorrain*
AN/GV *= Arrêt Normal sur GV*
AN/RRA *= Arrêt Normal sur RRA*
ANSI *= American National Standards Institute*
APE *= Approche Par Etat*
AMI *= Actionnement et Mesure Intelligent*
API *= Arrêt Pour Intervention*
APR *= Arrêt Pour Rechargement*
ARE *= Système d'alimentation normale*
ASG *= Alimentation de Secours des Générateurs de Vapeur*
AQAL *= All Quadrant, All Level*
BKCASE *= Body of Knowledge and Curillicum to Advance Systems Engineering*
CE *= Chef d'Exploitation*
CED *= Chef d'Exploitation Délégué*
CESAME *= Centre d'Excellence Sur l'Architecture, le Management et l'Economie des Systèmes*
CISPI *= Conduite Interactive et Sûre de Procédés Industriels*
CMMS *= integrated Control Maintenance and technical Management System*
CSM *= Commande Sans Modèles*
CRAN *= Centre de Recherche en Automatique de Nancy*
DAST *= Direction de la recherche des Automatismes*
DER *= Direction des Etudes et Recherches*
DIAS *= Distributed Intelligent Actuators and Sensors*
DIN *= Direction Ingénierie Nucléaire*
DoDAF *= Departement of Defence Architecture Framework*
DSA = *Distributed Situation Awareness*
DSE *= Délégué sécurité en Exploitation*
EDF *= Electricité De France*
FAI *= Fournisseur d'Actionnement Intelligent*
FMI *= Functional Mockup Interface*
FIPA *= Foundation for Intelligent Physical Agents*
GV *= Générateur de Vapeur*
HMS *= Holonic Manufacturing System*
IA *= Actionnement Intelligent*
IAMS *= Intelligent Actuation and Measurement System*
IDM *= Ingénierie Dirigée par les Modèles*
IEEE *= Institute of Electrical and Electronics Engineers*
INCOSE *= INternation COuncil Of System Engineering*
IM *= Mesurage Intelligent*
ISBM *= Ingénierie Système Basée sur des Modèles*
ISET *= Ingénierie des Systèmes Eco-Techniques*
ITEA *= Information Technology for European Advancement*
MBSE *= Model Based Systems Engineering*
MODAF *= Ministry of Defence Architecture Framework*

Abréviations

JADE *= Java Agent DEvelopment framework*
MFA *= Module Fonctionnel d'Automatisme*
OLF *= Objet Logique de Flux*
OLI *= Objet Logique Interactif*
OLE *= Object Linking and Embedding*
OLT *= Objet Logique Technique*
OMF *= Objet Matériel de Flux*
OMG *= Object Management Group*
OMI *= Objet Matériel Interactif*
OMT *= Objet Matériel Technique*
OP *= Opérateur de Conduite*
OPC *= OLE for Process Control*
PLM *= Product Lifecycle Management*
PRIAM *= Prenormative Requirements for Intelligent Actuation and Measurement*
RCD *= Réacteur Complétement Déchargé*
REP *= Réacteur à Eau Pressurisée*
RFLP = *Requirement Functional Logical Physical*
RP *= Réacteur en Production*
RRA *= Refroidissement du Réacteur à l'Arrêt*
RTE *= Réseau de Transport d'Electricité / Rupture Tuyauterie d'Eau alimentaire*
RTV *= Rupture Tuyauterie Vapeur*
SA *= Situation Awareness*
SdI *= Système d'Intérêt*
SEBoK *= Systems Engineering Body of Knowledge*
SIAC *= Système Interactif d'Aide à la Conduite*
SRL *= System Readiness Level*
SysML *= System Modeling Language*
SYMPA *= Système de Production Ambiant*
TMI *= Three Mile Island*
TRL *= Technology Readiness Level*
UML *= Unified Modeling Language*

Introduction générale

Introduction générale :

Les travaux présentés dans ce mémoire de thèse ont été réalisés au sein du département ISET (Ingénierie des Systèmes Eco-Techniques) du Centre de Recherche en Automatique de Nancy (CRAN - CNRS UMR 7039). Les problématiques abordées dans ce mémoire trouvent leur source d'un partenariat académique entre l'équipe ISET et EDF au sein du cluster CONNEXION[2] (COntrôle Commande Nucléaire Numérique pour l'EXport et la rénovatION), et précédemment, d'un long partenariat au sein de programmes européens de R&D dans le domaine de l'énergie.

Par ailleurs, le constat **à postériori** de défaillances systémiques ayant conduit à des situations incidentelles ou accidentelles de systèmes industriels complexes soulève la question des relations de causalité du comportement d'ensemble des éléments constitutifs d'un « système-produit[3] » sur le comportement individuel de chacun de ces éléments et inversement.

> *The reality is that systems do not exist in isolation… Systems are also parts, and as such, they inhabit worlds of systems[4] …, which are largely unknown to us and we discover their existence all too tragically via systemic failure …. **Is it possible using systemic thinking as the natural antidote of systemic failure**? (Boardman & Sauser, 2013b).*

Ce constat montre que nous prenons conscience de façon courante d'un «TOUT» par ses effets que nous désignons par la matérialité fonctionnelle d'une « PARTIE » afin de le rendre tangible. « Ce qu'est un système ? **»** demeure ainsi un questionnement récurrent dans de nombreuses communautés scientifiques, y compris celle de l'« ***Ingénierie-Système*** » dans laquelle nous positionnons nos travaux. Cette communauté se réfère pour « Penser-Système » à un ensemble de travaux relatifs à une « **Théorie Générale des Systèmes** » (Von Bertalanffy, 1968) que (Pouvreau, 2013) a récemment actualisée dans un cadre de « **systémologie générale** ».

En réponse au questionnement conceptuel suscité par la relation récursive de couplage (Figure1) entre les domaines de la situation-système et de l'ingénierie-système, nous postulons dans nos travaux que la cause de certaines défaillances de ces systèmes critiques peut provenir de l'organisation de leur processus collaboratif d'ingénierie interdisciplinaire. Nous avons considéré une perspective holonique permettant de rendre compte de ces comportements perçus, requis, visés et produits « *partiellement comme une **totalité H** ou totalement comme une **partie h***» (Koestler, 1967) afin de prendre en compte la dimension subjective de la réalité d'une situation-système tant au niveau collectif qu'individuel.

---

[2] https://www.cluster-connexion.fr/

[3] Système Produit : A system context for which the system-of-interest (SoI) is a product (Pyster, et al., 2012). Il s'agit de l'artefact « ingéniéré » délivré par un système « ingéniérant » afin de produire un « système d'intérêt » requis par une situation-système.

[4] Avec de plus en plus la nécessité de prendre en compte les « systèmes-produits » par la nature, qui ne font pas l'objet de ce mémoire, bien que certains éléments constitutifs du « système à faire » soient « naturels »

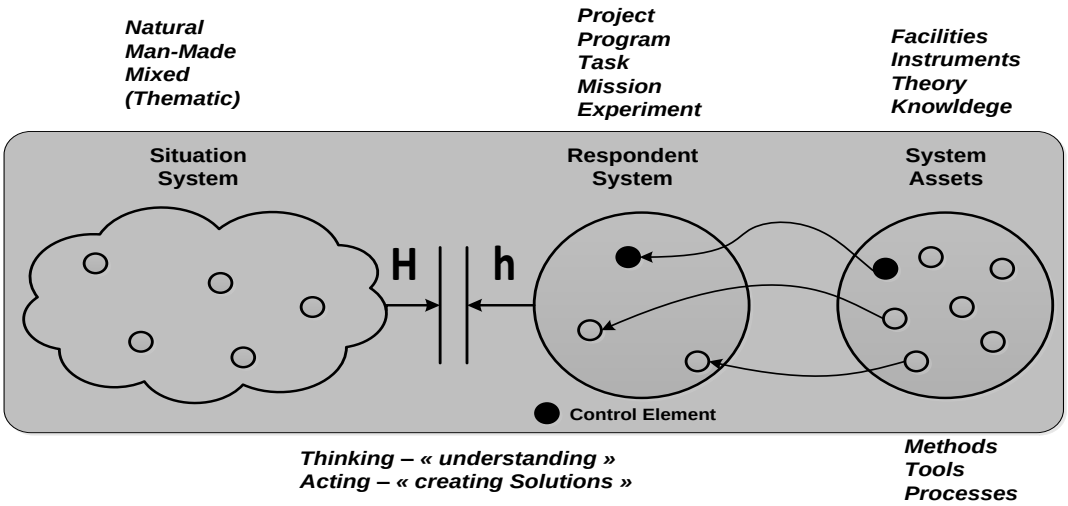


Figure 1 : Interprétation du diagramme de couplage système (HW Lawson, 2010)

En réponse au questionnement méthodologique sous-jacent au « système pour faire », nous avons interprété divers travaux de l'approche des « problem-frames » en explorant la nature **spécifiante**, **interdisciplinaire** et **itérative** de cette relation récursive de couplage (Figure 2) afin de prendre en compte la dimension objective de la réalité[5] d'une architecture-système tant au niveau collectif qu'individuel.

Cette heuristique de modélisation forme deux visions architecturales du « système à faire[6] » reflétant :

- Partiellement :
  - la connaissance K_S d'une totalité H d'une situation réelle afin de satisfaire la requête R_S d'un « système à faire » M_H selon : $K_S, M_H \vdash R_S$
  - les connaissances K_s de ses parties h dans la réalité de la situation afin de satisfaire la requête R_s d'éléments constitutifs M_h selon : $K_s, M_h \vdash R_s$,

- Totalement :
  - la connaissance systémique K_H d'un « système à faire » afin de satisfaire la requête R_H d'une faisabilité M_I selon : $K_H, M_I \vdash R_H$,
  - les connaissances interdisciplinaires K_I de « constituants à faire » M_h afin de satisfaire la requête R_I d'intégration d'un ensemble de faisabilités M_h selon : $K_I, M_h \vdash R_I$.

5 Requirements R are located in the reality and the entailment relationship K, S ⊢ R means that a proof can be given that an implementation of a specification S, introduced into a knowledge domain K, will guarantee satisfaction of the requirements R (interpretation recursive of (Gunter, Gunter, Jackson, & Zave, 2000) works).

6 Système à faire : système étudié par l'ingénierie système pour répondre au besoin (R). Le « système à faire » est initialement un concept, une abstraction, le résultat concret de sa réalisation est un « produit » (produit-système) ayant les caractéristiques d'un système et répondant à une définition qui en précise toutes les caractéristiques techniques, de production, d'exploitation de maintenance. (Fiorèse & Meinadier, 2012)

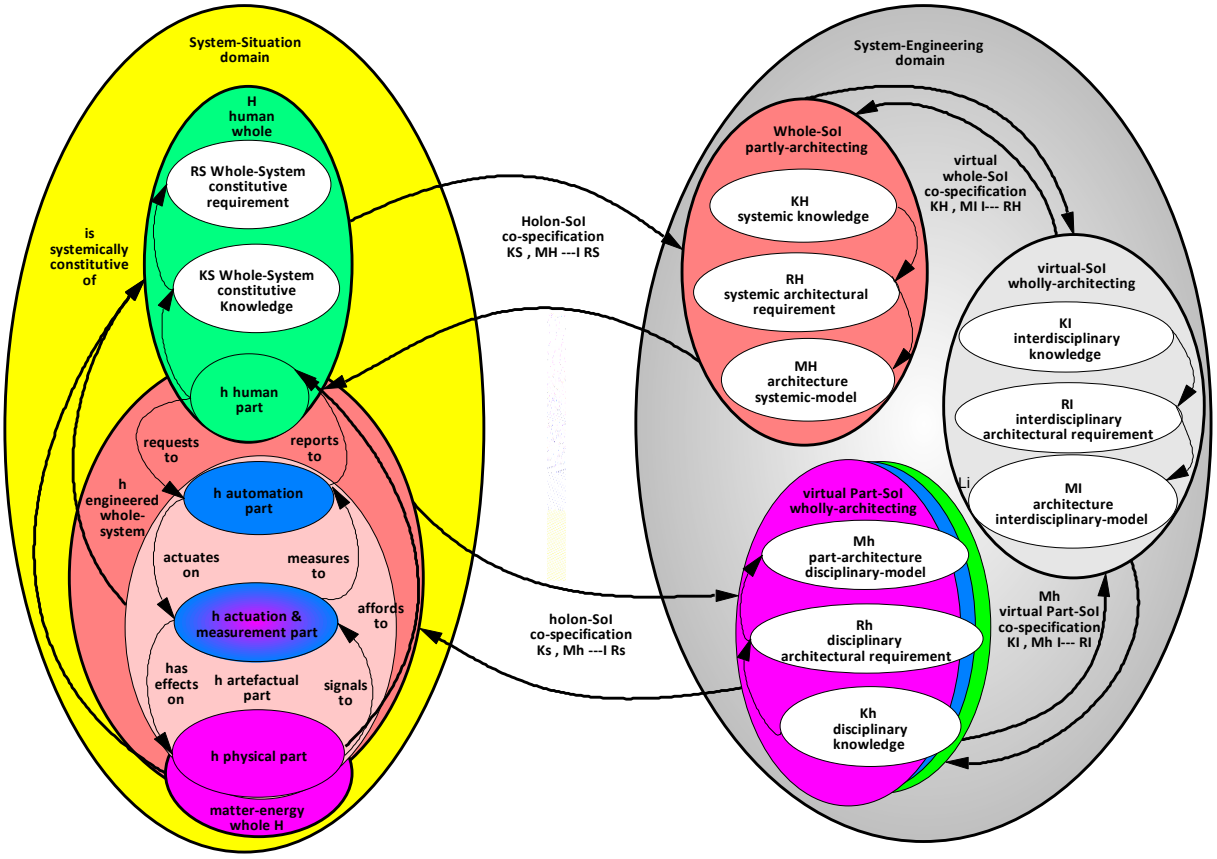


Figure 2 : Systémigramme explicatif des relations TOUT et PARTIE de notre heuristique de co-spécification exécutable basée sur des modèles d'un système d'intérêt

En ce sens, nous avons tiré parti de la maturité récente des technologies de modélisation et de simulation numériques facilitant le **couplage de modèles** entre l'ensemble des domaines d'ingénierie constitutifs d'un processus d'« **Ingénierie Système Basée sur des Modèles** ». La technologie mise en œuvre s'appuie sur le langage de modélisation système « **SysML** » pour le partage de connaissances ainsi que la vérification et la validation par co-exécution de modèles de la spécification virtualisée d'un système d'intérêt faisable (***M***$_I$) autour d'un bus de co-simulation, y compris in-situ avec notre plate-forme d'expérimentation.

> *A **paradigm shift** is occurring within Systems Engineering, where the level of complexity of system-projects can only accurately and precisely be modeled by a modeling construct that is **executable**. **Whole-system** design means optimizing not just parts but the entire system. Everything must be considered simultaneously and teased apart to reveal mutually helpful interactions (Garcia, 2009).*

Cependant, cette heuristique de modélisation sous-tend un processus cognitif de co-spécification interdisciplinaire entre l'ensemble des parties-prenantes afin de faciliter la construction « par détours » (Berthoz, 2014) d'une architecture-système réaliste.

> *Penser l'intégration d'un système complexe hétérogène, résultant de l'interaction de plusieurs systèmes homogènes, nécessite par conséquent de **raisonner transversalement en construisant des modèles d'un type nouveau** qui capturent l'émergence en intégrant de manière cohérente les parties de chaque modèle homogène constitutif d'un modèle systémique d'ensemble. Or force est de constater que rien ne nous prépare à ce nouveau paradigme qu'est le raisonnement systémique, qui oblige à mettre en relation des savoirs de natures extrêmement différentes et à transcender les silos traditionnels d'organisation de la connaissance (Krob, 2012).*

Notre proposition d'environnement collaboratif (Figure 3) a ainsi pour objectif de faciliter le travail interdisciplinaire en intégrant l'ensemble des environnements disciplinaires requis de co-modélisation M_h de parties-système, avec pour objectif assigné de préserver les outils, méthodes et processus de travail de chacune des parties prenantes.

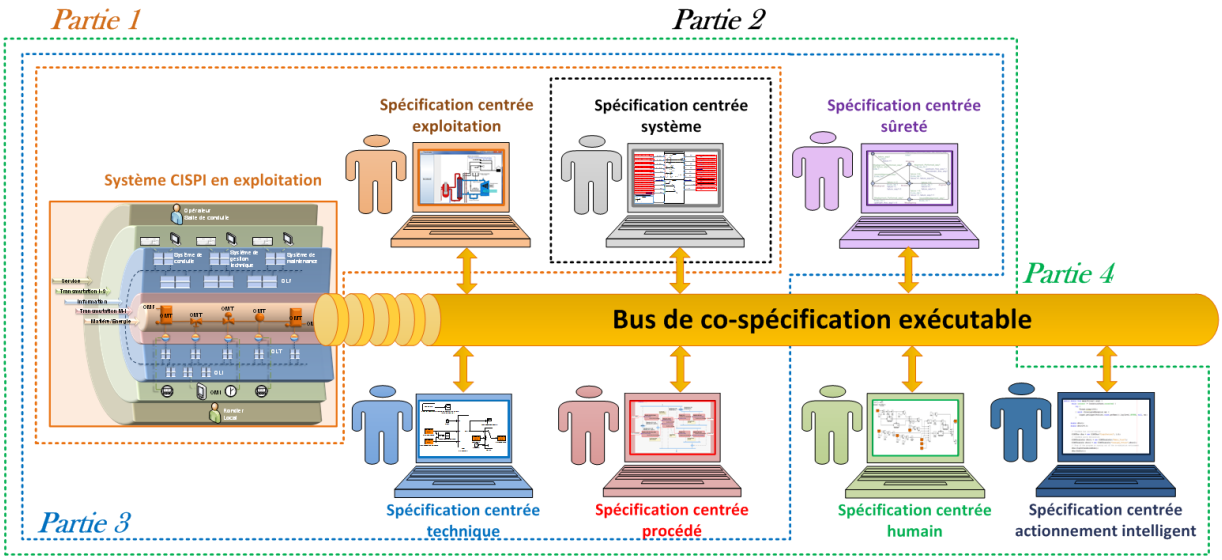


Figure 3 : Proposition d'un environnement de co-spécification système exécutable basée sur des modèles

Enfin, afin que nous puissions jouer le rôle d'architecte-système pour valider notre proposition, notre environnement cognitif de travail est lui-même constitué d'un ensemble d'experts[7] qui, à titres divers, nous ont permis de couvrir la dimension interdisciplinaire de modélisation et d'ingénierie système de notre étude de cas.

La structuration de notre mémoire reflète la perspective holonique TOUT et PARTIE visée que nous limitons dans cette introduction générale au couplage des quatre parties constitutives pour en expliquer la logique d'ensemble que complètent une conclusion générale, une bibliographie ainsi qu'une annexe détaillant les différents modèles. Chacune de ces parties cible une interaction constitutive de la dimension collaborative d'ingénierie d'un système d'intérêt, limité au facteur d'échelle de la situation-système de notre cas d'étude.

Les deux premières parties reflètent partiellement les collaborations interdisciplinaires à viser pour la construction d'un **modèle partiel M_H de notre situation-système d'intérêt**.

La **première partie**, intitulée « ***Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI*** *(Conduite Interactive et Sûre de Procédés Industriels)* », cible une interaction collaborative entre métiers d'exploitation et d'ingénierie constitutive de la situation du système à faire. Elle contextualise en deux chapitres notre situation-système de Recherche & Développement qui reflète la problématique industrielle visant une architecture de type PLM (Plant Life cycle Management) afin d'assurer un continuum d'informations entre actifs-techniques et actifs-humains en situation opérationnelle aussi bien que d'ingénierie d'un système de conduite d'un procédé industriel. Cette exigence infère un continuum collaboratif de modélisation (« Model-in-the-loop ») et de connaissances (« Knowledge-in-the loop ») entre les domaines courants

[7] Académiques : C. Join, R. Pannequin, F. Mayer, D. Mery, P. Marangé, D. Gouyon et industriels : J.M. Dupont, J.P. Hennebicq, D. Galara

d'ingénierie centrée technique ou centrée humain pour concourir véritablement à une ingénierie centrée système (« System-in-the-loop »).

La **seconde partie**, intitulée « ***Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI*** », cible une interaction collaborative entre domaines d'ingénierie interdisciplinaire constitutive du « système pour faire ». Elle approfondit les fondements scientifiques de notre heuristique de modélisation-système pour produire l'ensemble des modèles descriptifs de problèmes R_S, R_H, R_h et prescriptifs de solutions M_H, M_I, M_h dans une perspective holonique TOUT (H) et PARTIE (h) du système d'intérêt requis. L'architecture collaborative proposée vise à faciliter le couplage non linéaire de l'ensemble de connaissances à percevoir, à requérir et à intégrer K_S, K_H , K_I, K_s d'un point de vue systémique.

Les deux premières parties détaillent certaines collaborations interdisciplinaires ciblées pour la construction d'un **modèle total M_I de faisabilité virtuelle de notre système d'intérêt** « ***h engineered whole-system*** » dépassant une modélisation technique artefactuelle (« ***h artefactual part*** »).

La **troisième partie,** intitulée « ***Co-spécification exécutable de constituants de conduite du procédé du système CISPI*** », cible une interaction constitutive de parties physiques (« ***h physical part*** ») et automatisation (« ***h automation part*** ») du système ingénieré visé. Elle met en évidence dans un premier temps la difficulté de prendre en compte de façon plus large la dimension multi-physique de la matière-énergie *(« **H matter-energy whole** »*) à contrôler. Cette faisabilité physique préliminaire est contrôlée dans un deuxième temps par une logique de contrôle continu et par états ne nécessitant pas d'identification à priori d'une structure existante afin de faciliter des choix d'alternatives de conduite, par exemple en terme de sûreté de fonctionnement ou de maintenance prédictive.

La **quatrième partie**, intitulée « **Co-spécification exécutable de constituants d'actionnement et de mesurage de conduite du système CISPI** », cible une interaction constitutive de parties d'actionnement et de mesurage intelligents (« ***h actuation and measurement part*** ») raffinant le modèle exécutable en cours de spécification. Si, dans un premier temps, un constituant-système embarquant une forme d'intelligence technique est spécifié, une attention particulière est portée dans un deuxième temps sur la possibilité de co-spécifier partiellement un modèle exécutable d'une interaction technico-physiologique de conduite (« ***human-part*** » & « ***automation-part*** »).

En conclusion, nous présentons un bilan de ces travaux qui ont reçu, par certains aspects, un écho favorable tant d'un point de vue industriel (Bouffaron, 2014) que scientifique (Bouffaron, Dupont, Mayer, & Morel, 2014) et pédagogique (Gouyon, Bouffaron, & Morel, 2014), (G. Morel, et al., 2014). Ces travaux n'en restent pas moins exploratoires pour contribuer à pallier :

- l'approche guidée par des processus qui a longtemps prévalu en Ingénierie-Système mais qui montre ses limites en termes de partage objectif de connaissances,
- l'approche basée sur des modèles qui fait l'objet de nombreux travaux techniques mais qui nécessite de mieux prendre en compte la dimension subjective de la réalité pour traduire l'intention d'un architecte-système (Retho, 2015).

C'est à ce verrou, au facteur d'échelle de notre environnement d'expérimentation, que sont attachés nos travaux en faisant du **modèle exécutable** le pivot d'une collaboration interdisciplinaire de co-spécification d'un système d'intérêt sans chercher à appliquer un cadre reconnu mais plutôt en suscitant une nouvelle voie.

Partie 1 : Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI

Introduction : Interaction collaborative entre métiers d'exploitation et d'ingénierie constitutive de la situation-système d'intérêt ciblée

Le cluster CONNEXION a pour ambition de proposer et de valider une architecture innovante de plate-formes modulaires de contrôle-commande adaptée aux centrales nucléaires en France et à l'international. Cette architecture « *vise à orchestrer de façon dynamique autour d'un bus d'objet métier, des réseaux de services dans l'espace et dans le temps à partir de sources d'informations hétérogènes, centralisées ou réparties, de qualité et de sensibilité différentes pour offrir aux métiers en tout temps et en tout lieu de la centrale des services dont ils ont besoin* » (Devic & Morilhat, 2013). Elle contribue notamment à pallier la dichotomie courante entre les métiers d'exploitations et d'ingénierie. Une problématique émergente consiste alors à assurer un continuum depuis la conception, jusqu'à l'exploitation et la formation, afin de faciliter l'acquisition de données, les échanges et les analyses entre les différentes parties prenantes conceptrices et utilisatrices (Figure 4).

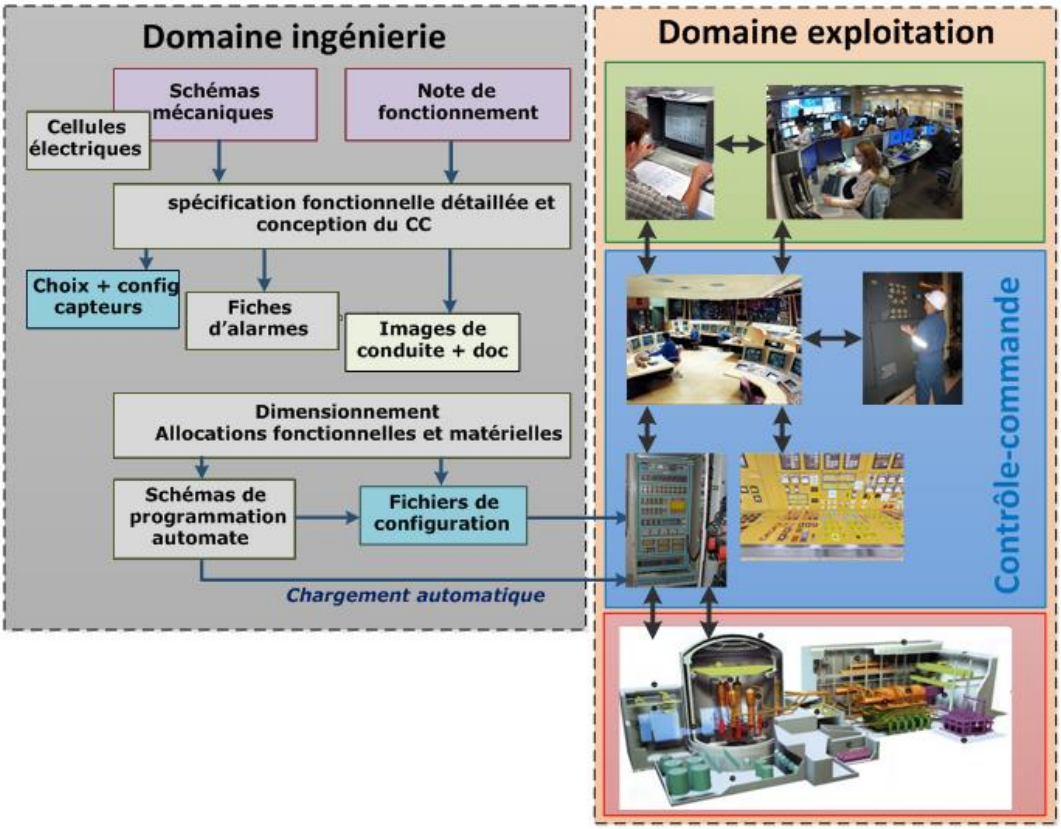


Figure 4 : Domaine d'ingénierie et d'exploitation d'un système de conduite (Devic & Morilhat, 2013)

En ce sens, l'objectif des deux chapitres de cette première partie est de contextualiser à la fois le domaine opérationnel de conduite du procédé industriel critique en exploitation, ainsi que le domaine d'ingénierie pour concevoir ce type de système.

Ainsi, le premier chapitre de notre mémoire précise notre domaine d'intérêt en détaillant la situation système de conduite d'un procédé industriel critique dans le domaine de l'énergie électrique. Une caractéristique essentielle de la conduite de ce type de procédé, siège d'émergences critiques, est qu'elle reste

majoritairement sous un contrôle humain (Devic & Morilhat, 2013), dont les métiers sont répartis en salle de commande, sur le terrain, dans les laboratoires…. Nous caractérisons ensuite les évolutions requises de l'architecture de conduite existante pour évoluer vers une architecture de conduite distribuée afin que les opérateurs puissent interopérer numériquement en tout lieu, à tout instant et pour toute action de conduite avec le procédé à contrôler

Le second chapitre, caractérise un ensemble de pratique en ingénierie de conduite centrée technique et centrée humain, pour défendre qu'une approche centrée système est requise pour spécifier ce type de système afin d'assurer la cohérence globale du continuum d'informations entre les agents techniques et humains. Nous statuons également sur l'utilisation de technologies de modélisations numériques, de plus en plus matures, pour la mise en œuvre de notre approche heuristique de co-spécification système basée sur des modèles exécutables.

Partie 1 : Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI

1. Chapitre 1 : Contexte opérationnel de conduite du système à faire CISPI

1.1. Introduction : Caractérisation de la situation-système de conduite CISPI

Dans un premier temps, nous caractérisons dans ce chapitre la situation système de conduite du procédé industriel critique faisant l'objet de nos travaux :

- d'un point de vue dynamique (*évolution du procédé au cours du temps*)
- d'un point de vue cognitif (*opérations de conduite pour maintenir le procédé dans un état donné*)

Dans un second temps, nous détaillons l'architecture de conduite existante par niveaux de hiérarchisation afin de faire émerger ses limites, notamment dans le partage d'informations entre les opérateurs de conduite répartis à la fois en salle de commande et sur le terrain. Nous présentons ensuite l'architecture de conduite visée, mise en œuvre dans notre cas d'étude (*plate-forme CISPI du CRAN*), permettant d'assurer un continuum d'information entre les agents humains, mais également les agents techniques de conduite

.

1.2. Caractérisation de la situation-système de conduite du procédé industriel critique d'intérêt

Nous basons notre étude sur un ensemble de travaux (G. Boy, 2011) et (Galara, 2011) statuant sur la notion de « systèmes en situation complexe » (Penalva, 1997). En ce sens, nous caractérisons dans un premier temps la situation dynamique à laquelle les opérateurs de conduite doivent s'adapter pour répondre, en exploitation, à un double objectif de production d'énergie (définis par le RTE (*Réseau de Transport d'Electricité*)) et de sûreté. Dans un second temps nous caractérisons la situation cognitive au cours de laquelle ces opérateurs se construisent une représentation de la situation dynamique, pour décider et agir sur cette dernière.

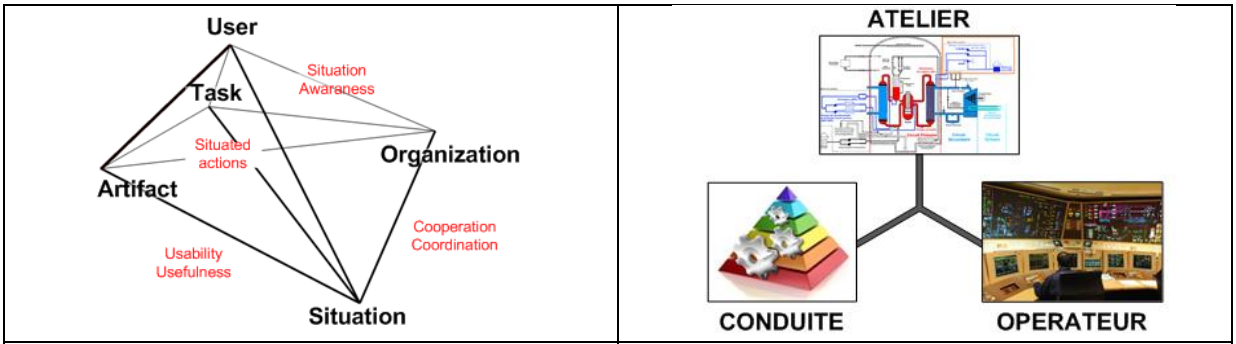


Figure 5a: Pyramide AUTOS (G. Boy, 2011)	Figure 5b : Situation de conduite (Penalva, 1997)
Figure 5 : Cadre d'analyse d'une situation-système de conduite	

1.2.1. Caractérisation de la situation dynamique de conduite

Notre situation de conduite du procédé industriel critique présente la caractéristique essentielle d'être une situation dynamique. En effet, le procédé nucléaire à conduire repose sur une réaction en chaîne[8], évoluant de

[8] Dans le domaine du nucléaire, une **réaction en chaîne** se produit lorsqu'un neutron cause la fission d'un atome fissile produisant un plus grand nombre de neutrons qui à leur tour causent d'autres fissions.

manière continue sans interruption et indépendamment des actions de l'homme et du système technique (Palaci, 2009), c'est-à-dire en fonction de la dynamique interne du procédé (en l'occurrence la réaction en chaîne). L'exploitation de ce procédé nécessite de modérer et de contrôler cette réaction afin d'éviter la fusion du cœur du réacteur, pouvant se produire en particulier lorsque ce dernier cesse d'être correctement refroidi. La fusion du cœur est considérée comme un accident nucléaire grave en raison de la probabilité de dispersion de substances radioactives et de contamination de l'homme et de l'environnement. En ce sens, le procédé nucléaire est un procédé critique, dont la sûreté repose sur le maintien de trois fonctions principales de sûreté :

Fonction 1	*Contrôler la réaction en chaine, pour garantir le non emballement ou toute excursion de puissance thermique d'amplitude et/ou de durée excessive*
Fonction 2	*Maintenir le refroidissement du combustible en toute circonstance, par évacuation de la puissance produite, y compris après l'arrêt du réacteur*
Fonction 3	*Confiner les produits radioactifs par le maintien des trois barrières étanches (la gaine des crayons combustibles, l'enveloppe du circuit primaire et l'enceinte de confinement)*

Tableau 1 : Principales fonctions de sûreté d'une centrale nucléaire

Dans ce mémoire nous nous focalisons sur le maintien de la fonction n°2 de sûreté concernant le refroidissement du combustible des réacteurs de technologie REP (Réacteur à Eau Pressurisée) de type 1300 MWe. Cette exigence doit être satisfaite en permanence, aussi bien pendant la marche du réacteur, qu'après son arrêt normal ou accidentel. Bien qu'il soit possible de stopper la fission nucléaire[9], le danger d'une fusion du combustible contenu dans le cœur du réacteur subsiste pendant des jours et même des semaines. En effet, l'activité des produits de fission continue de produire de la chaleur même après l'arrêt de la réaction. Ainsi, un des enjeux majeurs des REP concerne le refroidissement de la chaleur, afin d'éviter une montée en température excessive du réacteur pouvant entrainer la fusion du cœur.

En exploitation, l'évacuation de l'énergie d'un REP est assurée par trois circuits indépendants où circule de l'eau servant de fluide caloporteur chargée de transporter la chaleur du cœur entre ces différents circuits (Jacquemin, 2013) (Figure 6):

- Le circuit primaire permet de refroidir le cœur du réacteur par une circulation d'eau utilisée sous une pression suffisamment élevée pour rester liquide (155bars) à sa température de fonctionnement (300°C)
- Le circuit secondaire évacue l'énergie du circuit primaire par l'intermédiaire des Générateurs de Vapeur (GV).
- Le circuit tertiaire comporte un condenseur refroidissant l'eau du circuit secondaire, évacuant ainsi l'énergie restante vers une rivière, un fleuve, la mer ou dans l'atmosphère par des tours de réfrigérations

En cas d'indisponibilité du système d'alimentation normale (ARE) en situation incidentelle ou accidentelle de type (*Rupture Tuyauterie d'Eau alimentaire (RTE), Rupture Tuyauterie Vapeur (RTV), …*) l'alimentation en eau des GV bascule sur le système de sauvegarde d'Alimentation en eau de Secours des Générateurs de vapeur (ASG). La mission du système ASG est d'alimenter en eau la partie secondaire du GV afin de maintenir un

[9] La réaction nucléaire peut être stoppée en faisant chuter les grappes de barres de contrôle dans le cœur du réacteur et en injectant de l'acide borique dans l'eau où est immergé le combustible

niveau d'eau suffisant à l'intérieur pour continuer à refroidir le circuit primaire. Il extraie alors la puissance résiduelle du cœur jusqu'à ce que les conditions de mise en service du système de Refroidissement du Réacteur à l'Arrêt[10] (RRA) soient atteintes (Coppolani, 2004). Pour les réacteurs de type 1300 MWe, le système ASG est également utilisé en fonctionnement normal durant les phases d'arrêt et de démarrage du réacteur nucléaire pour le remplissage des GV. Le système ASG assure ainsi une des onze fonctions principales d'une centrale nucléaire, participant à la fonction de sûreté de refroidissement. Ce système critique, dont une défaillance peut entrainer une dégradation importante du combustible, constitue l'**A**rtefact[11] à conduire de notre situation d'intérêt.

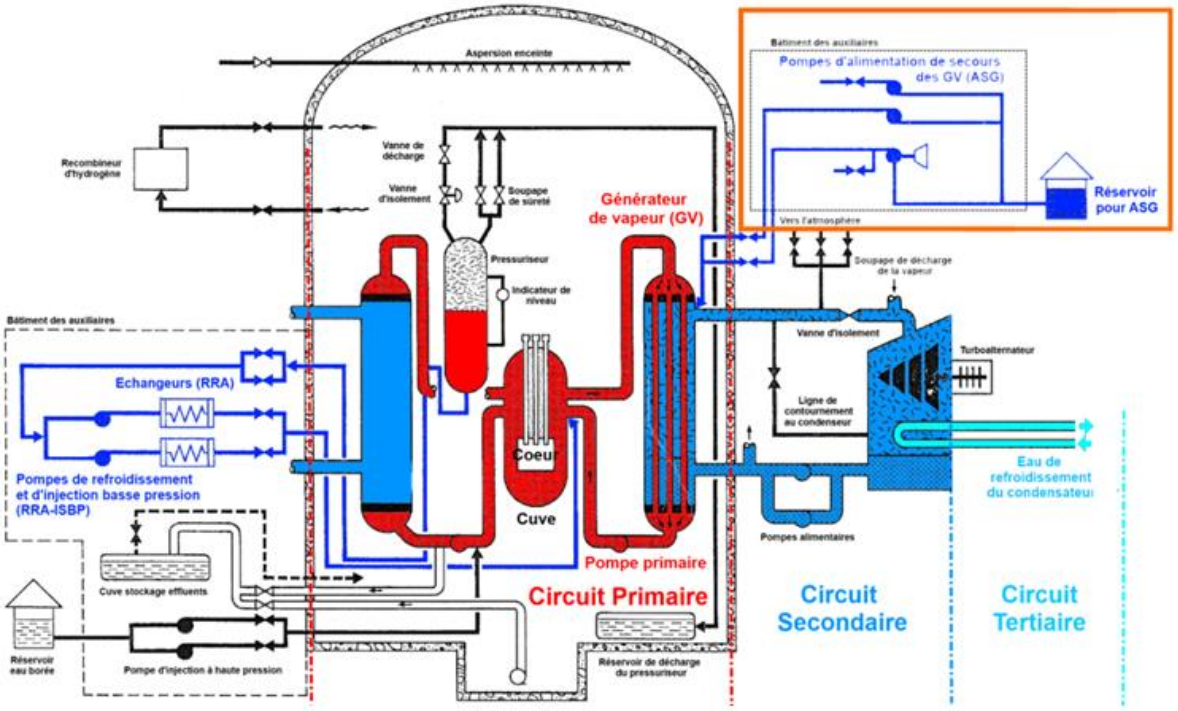


Figure 6: Principaux circuits de refroidissement d'un REP (adaptée de (Duco, 2004))

1.2.2. Caractérisation de la situation cognitive de conduite

Système socio-technique

La maitrise de cette situation dynamique, siège d'émergences critiques, reste majoritairement sous un contrôle humain (Devic & Morilhat, 2013), quel que soit le niveau d'automatisme.

---

[10] Le système RRA a pour fonction, lors d'une mise à l'arrêt normal du réacteur, d'évacuer la chaleur résiduelle produite par le combustible dans la cuve du réacteur et de maintenir l'eau primaire à une température modérée tant qu'il y a du combustible dans le cœur du réacteur. En effet, après l'arrêt de la réaction en chaîne, le cœur du réacteur continue de produire de la chaleur qu'il est nécessaire d'évacuer pour ne pas risquer d'endommager gravement le combustible (Medjaher, 2005)

[11] Un **A**rtefact est un objet matériel ou immatériel (conceptuel) conçu par l'homme, utile pour une classe d'**U**tilisateurs, afin d'effectuer tes **T**âches spécifiques (HB Lawson, Wade, & Hofkirchner, 2005-2014)

Partie 1 : Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI

> « *Bien que certains se font des illusions ou que d'autres continueront à s'en faire sur le mythe de l'entreprise complétement automatisée, l'homme est, et restera longtemps encore, seul capable d'animer des activités dans l'entreprise. Seule la nature du travail évoluera* ». (Després, 1991)

Cela s'explique par la nature résiliente de l'humain capable de s'adapter et de faire face à des situations inattendues, relatifs au caractère imprévisible et évolutif du procédé nucléaire à piloter. (Hollnagel, 2006) présente la résilience comme étant la caractéristique la plus importante des systèmes et organisations orientés sûreté. De plus, bien que ces métiers soient assistés à différents degrés par des technologies numériques, un pourcentage faible des informations de conduite est « automatisé ». En effet, pour des exigences techniques et de couts, il n'est pas possible d'automatiser l'ensemble des équipements d'une installation nucléaire. Ainsi, on estime à plus de 80% la part d'informations provenant d'un réseau de traitement via un opérateur ou le plus souvent de l'opérateur lui-même (en salle de commande, sur le terrain, laboratoire de chimie, maintenance…) (Figure 7). A titre d'illustration, plus de 91 % des vannes et robinets présents dans une tranche[12] de centrale nucléaire sont des équipements manuels (Tableau 2) que les agents de terrain manipulent durant les opérations de conduite.

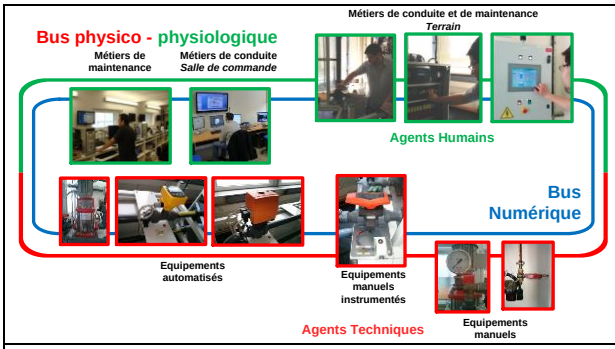


Figure 7 : Système de conduite d'un procédé en exploitation

Capteurs ou actionneurs	Nombres moyen par tranche
Vannes et robinets manuels	Entre 15000 et **20000**
Vannes motorisés	300
Vanne pneumatiques	950
Vannes réglantes	100
Capteur Logique	3000
Capteur Analogique	2500

Tableau 2 : Nombres de capteurs et d'actionneurs dans une tranche de centrale nucléaire (Devic & Morilhat, 2013)

Ces agents humains interagissent avec le procédé d'une part, via les informations et les moyens techniques disponibles en salle de commande, et d'autre part, en parcourant le site de production d'énergie électrique pour remonter un ensemble d'informations concernant l'exploitation de la centrale et non disponibles en salle de commande mais également pour agir sur des équipements non contrôlables à distance. Pour cela ils utilisent leur appareil musculo-squelettique pour actionner les équipements (*vannes, robinets, disjoncteurs,…*), ainsi que leur appareil sensoriel pour observer les phénomènes. Un agent de terrain, vu alors comme : « les yeux et les oreilles » des opérateurs en salle de commande utilise :

- le **toucher** pour percevoir les vibrations d'une pompe en fonctionnement, actionner les vannes manuelles (Figure 8b),…
- l'**ouïe** pour percevoir les « gargouillis » de l'eau dans une canalisation, le sifflement de l'air lors d'opérations d'éventages, une alarme en salle de commande ou sur le terrain,…
- la **vue** pour percevoir un niveau d'eau dans une cuve, un voyant (Figure 8a), un afficheur analogique/numérique (Figure 8c) …

[12] La tranche est l'unité de production qui réunit l'ensemble des installations permettant de produire l'électricité à partir du combustible primaire. Elle regroupe donc le bâtiment réacteur, le bâtiment combustible, la salle des machines et une ou plusieurs tours de refroidissement. (www.lenergieenquestions.fr/)

Partie 1 : Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI

Figure 8a) Panneau de conduite d'une centrale avec de multiples voyants, compteurs, cadrans…	**Figure 8b) Actionnement d'une vanne manuelle par un agent de terrain**	**Figure 8c) Lecture d'un afficheur numérique par un agent de maintenance**
Figure 8 : Interaction physico-technico physiologiques de conduite		

En exploitation, l'équipe de conduite a pour mission première de garantir un haut niveau de sûreté, en maintenant l'installation dans le domaine de fonctionnement défini par les Spécifications Technique d'Exploitation (STE). En parallèle, l'équipe doit également assurer l'adaptation du point de fonctionnement de l'installation aux besoins du réseau. En situation normale, l'activité des opérateurs se caractérise par deux grands types de **T**âches à effectuer : surveiller quotidiennement l'installation pour détecter et diagnostiquer les anomalies de fonctionnement et effectuer les manœuvres d'exploitation nécessaires à la conduite et aux opérations de maintenance. En situation dégradée, leur mission est de diagnostiquer les situations incidentelles ou accidentelles. Le cas échant, les équipes doivent replier l'installation dans l'état le plus sûr de fonctionnement, tout en minimisant les conséquences radiologiques et la dégradation du matériel à partir des procédures.

Approche Par Etats

« A l'origine les procédures de conduite étaient fondées sur une pure approche évènementielle » (Appell & Chambon, 1998). Cette approche prévoyait une identification préalable des évènements initiateurs pouvant conduire à un incident ou à un accident afin d'adopter la procédure de conduite la plus appropriée. Ces procédures sont élaborées à partir du déroulement prévisible de l'incident ou de l'accident et sont applicables si les évènements initiateurs, pouvant conduite à la situation dégradée, ont été correctement diagnostiqués. Le retour d'expérience de l'accident de « Three Miles Island-2[13]» a montré la nécessité d'une évolution de cette approche évènementielle. En effet, comme le souligne (Birraux, Bataille, & Sido, 2011), cette approche présente plusieurs inconvénients majeurs lié à la difficulté, voire à l'impossibilité, de prévoir tous les problèmes pouvant survenir sur une centrale, notamment dans le cas où plusieurs seraient concomitants et indépendants. De plus, le risque d'un mauvais diagnostic ou d'un diagnostic partiel des sources du problème est important, et ce, pour des raisons qui peuvent être d'origine humaine comme technologique (par exemple dans le cas de mauvaises informations fournies aux opérateurs : *Lors de l'accident de TMI-2, les voyants n'indiquaient que l'effectivité de l'envoi de l'ordre et non son exécution. Ainsi, la vanne de vidange du circuit primaire est apparue fermée aux opérateurs en salle de commande alors qu'elle était en fait ouverte. Cela a conduit à l'asséchement quasi complet de la cuve du réacteur, et une des causes principales de la fusion partielle du cœur*). Enfin cette approche rend très difficile l'évolution de la stratégie de pilotage dans le cas où les paramètres physiques de celle-ci n'évolueraient pas comme prévu. L'ensemble de ces problèmes et leur

[13] L'accident de Three Man Island (TMI) s'est produit le 28 mars 1979 : A la suite d'une chaîne d'évènements accidentels, le cœur du réacteur n°2 de la centrale de Three Mile Island (TMI-2) a en partie fondu, entrainant le relâchement dans l'environnement d'une faible quantité de radioactivité (Duco, 2004)

impact potentiellement très aggravant a conduit, dans le cadre des conditions post TMI-2, à l'abandon de l'approche évènementielle pour l'exploitation des centrales nucléaires françaises.

En ce sens, EDF et Areva (Framatome à l'époque) ont proposé une nouvelle approche pour pallier aux problèmes de l'approche évènementielle : l'Approche Par Etat (APE). L'APE, à partir de la connaissance de l'état physique du réacteur et de la disponibilité du matériel, permet de définir la stratégie de conduite à adopter, puis les actions à mener, sans avoir à identifier la séquence d'évènements antérieurs (initiateurs et aggravants) ayant conduit à cet état (Figure 9).

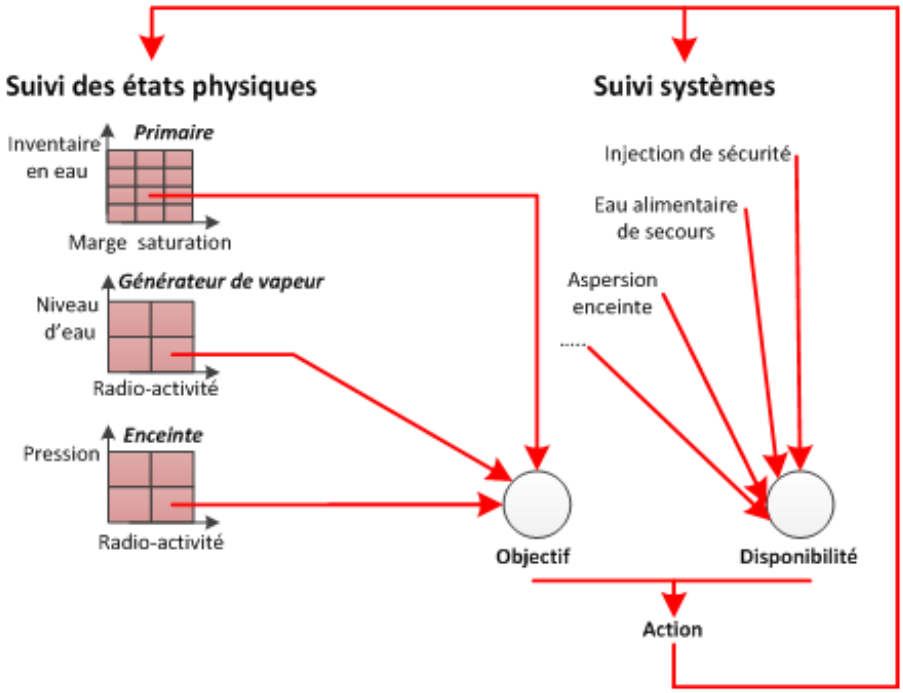


Figure 9 : Principe de l'Approche par Etat (Appell & Chambon, 1998)

Les conditions d'évènements de défaillances multiples peuvent être reproduis à l'infini. En revanche une analyse du comportement systémique du réacteur, engagée suite à l'accident de TMI-2, a permis de démontrer que les états physiques possibles du réacteur sont en nombre limité. Ils peuvent être identifiés à partir d'un nombre limité de données caractérisant l'état physique des principaux composants du réacteur (Tableau 3).

Conduite	Etats physiques des principaux composants	Fonction de sûreté associée	Grandeurs mesurées
Primaire	**Sous criticité et contrôle de la puissance nucléaire**	1	Wn = puissance neutronique
	Inventaire en eau du circuit primaire	2	Nc = niveau cuve
	Evacuation de la puissance résiduelle	2	*T*RIC = température sortie cœur
Secondaire	**Inventaire en eau secondaire**	2	NGV = niveaux GV
	Intégrité secondaire (GV)	3	Activité GVs
Enceinte	**Intégrité de l'enceinte de confinement**	3	Pe = pression enceinte DdDe = débit de dose enceinte

Tableau 3 : Etats physiques des principaux composants du réacteur et grandeurs physiques associées (Tarride, 2013)

Un état physique du réacteur est ainsi caractérisé par un regroupement des états des principaux composants relevant des mêmes priorités et donc des mêmes grands types d'action de conduite. Il y'a donc correspondance entre un état type et une stratégie de conduite adaptée. Par exemple, le fonctionnement normal des centrales de type 1300 MWe est décrit par 6 domaines d'exploitations (Figure 10) subdivisés en 14 états standards (Cf. Annexe 1-1) :

- Réacteur Complétement Déchargé (RCD) : (1) Tout combustible dans le bâtiment réacteur,
- Arrêt Pour Rechargement (APR) : (2) Arrêt à froid pour rechargement,
- Arrêt Pour Intervention (API) : (3) Arrêt à froid pour intervention primaire suffisamment ouvert, (4) Arrêt à froid pour intervention primaire entrouvert, (5) Arrêt à froid normal primaire fermé et dépressurisé,
- Arrêt Normal sur RRA (Refroidissement Réacteur à l'Arrêt) (AN/RRA) : (6) Arrêt à froid normal, (7) Arrêt intermédiaire monophasique, (8) Arrêt intermédiaire diphasique aux conditions RRA connecté
- Arrêt Normal sur GV (AN/GV) : (9) Arrêt intermédiaire diphasique aux conditions RRA isolé, (10) Arrêt intermédiaire diphasique sur GV, (11) Arrêt à chaud,
- Réacteur en Production (RP) : (12) Recherche de la criticité, (13) Attente à chaud, (14) Fonctionnement en puissance

L'APE se base sur l'étude de la séquence d'activités mentales permettant à l'humain de prendre une décision dans le contexte de la conduite à partir de son expérience, sa connaissance et de procédures (Rasmussen, 1986). D'un point de vue ergonomique, l'état physique du réacteur correspond à la situation dynamique que l'opérateur de conduite doit comprendre, afin de la maintenir dans des limites acceptables. Pour plus de précisions, nous qualifions cette situation de situation d'états de la centrale. Cette compréhension de la situation d'états garantit à l'opérateur un niveau de sécurité et de performance satisfaisants et repose sur un double axe, d'une part diachronique (compréhension de la situation d'états au cours du temps) et d'autre part, synchronique (prise d'information dans la situation d'états à un instant « t ») (Figure 10).

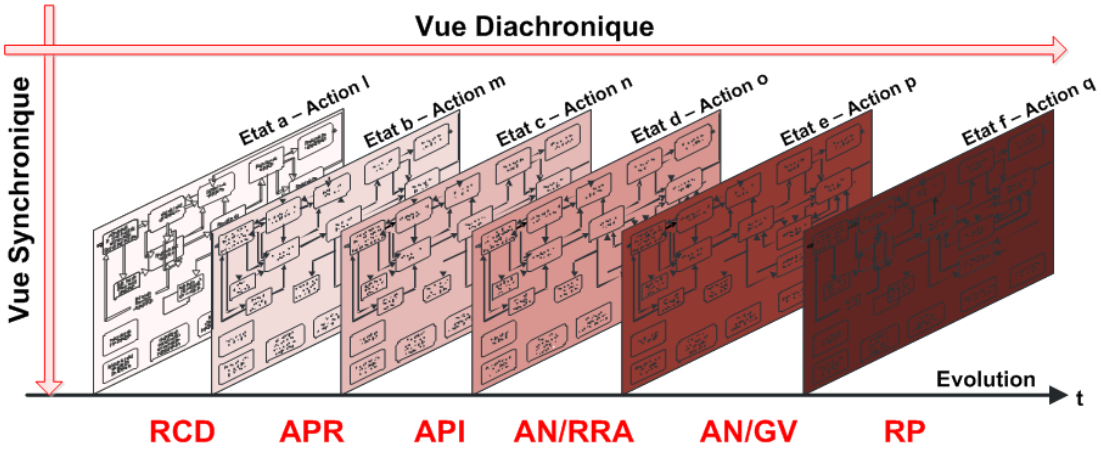


Figure 10 : Conduite du procédé nucléaire vue comme une évolution entre états stables à travers le temps (Galara & Hennebicq, 1999)

Etant donné que l'opérateur ne contrôle que partiellement la situation dynamique, celui-ci doit disposer à tout moment d'une représentation de la situation, représentation qui évoluera ainsi au fil du temps et qui sera à certains moments incomplète du fait des caractéristiques de l'environnement (imprévisibilité, contrôle partiel, multiples variables en interaction) (Neyns, 2011). La représentation de la situation, est une construction mentale élaborée dans un contexte particulier à des fins spécifiques pour faire face aux exigences de la tâche en cours (Richard, 1990). Ainsi cette représentation est finalisée par la tâche et la nature des décisions à prendre. La représentation de la situation d'états occupe une place centrale dans l'activité de conduite par et entre les opérateurs pour faciliter la compréhension individuelle et collective de la situation dynamique en exploitation. Pour maitriser la complexité de l'installation nucléaire, les opérateurs peuvent distinguer différents niveaux d'abstractions (Viel, Cheriaux, & Galara, 2012) :

- Le niveau « fonctionnement général » correspondant au niveau « Unité de production »
- Le niveau « système élémentaire » correspondant au niveau « fonctions principales d'un système »
- Le niveau « Groupe d'Equipements Fonctionnels (GEF) » correspondant à un regroupement d'équipements de base correspondant en général à un sous-ensemble d'un système élémentaire participant à une « sous-fonction d'un système »
- Le niveau « équipement de base » (vannes, pompes,…)

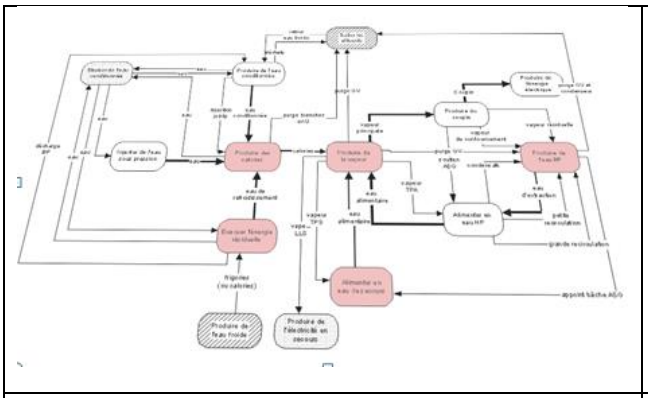


Figure 11a : En tant que réseau de systèmes élémentaires

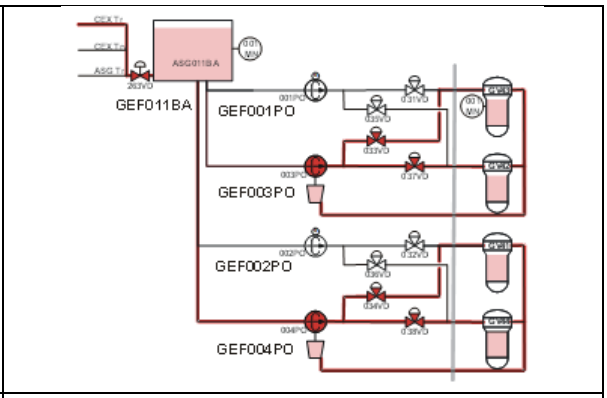


Figure 11b : En tant que réseau de GEF et équipements

Figure 11 : Représentation de la situation d'états du système ASG à différents niveaux d'abstractions (Viel, Cheriaux, & Galara, 2012)

A partir de la représentation de la situation d'états et des objectifs de production, les opérateurs de conduite définissent les objectifs de conduite visant à amener ou maintenir l'installation dans un état sûr de fonctionnement (Figure 12).

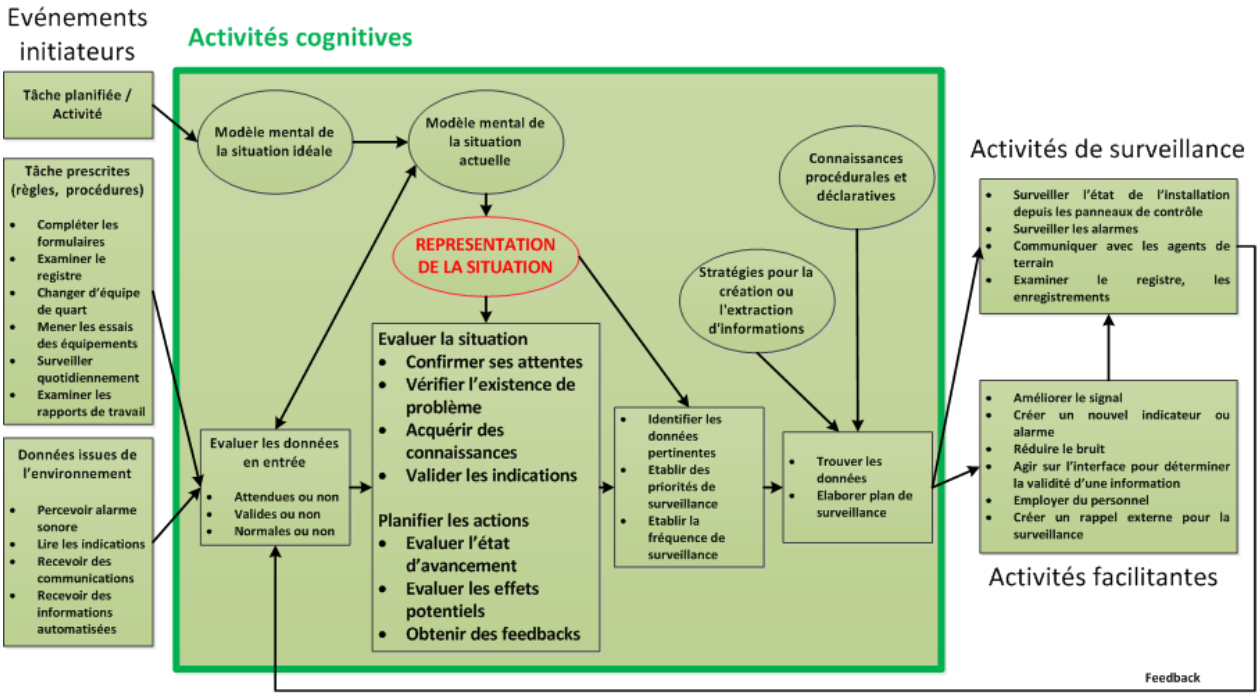


Figure 12 : Modèle de la conduite par un opérateur d'une situation dynamique (procédé nucléaire) (Vicente, Mumaw, & Roth, 2004)

Ainsi, la conduite du procédé nucléaire est perçue comme une évolution d'un état stable initial vers un état stable final, à travers le temps (Figure 10). Cette évolution est réalisée par des actions (**T**âches) exécutées séquentiellement par les opérateurs de conduite depuis la salle de commande, mais également en local par les agents de terrain. (Galara, 2011) distingue trois niveaux de conduite : deux niveaux pour la capacité d'agir (stratégie et tactique) et le niveau de l'action sur le système technique :

- Le niveau stratégique est global pour le système technique. Il correspond à un enchainement en série et/ou en parallèle des objectifs de conduite des fonctions principales, pour atteindre l'objectif de production du système technique, lors de la remise en exploitation et pour le maintenir en production.
- Le niveau tactique est spécifique à chacune des fonctions principales. Il correspond à un enchainement en série et/ou en parallèle des actions de conduite des fonctions principales, pour atteindre les objectifs des différentes fonctions principales, lors de la remise en exploitation et pour le maintenir en production.
- Le niveau de l'exécution est spécifique à chaque action de conduite. Il correspond à une suite de manœuvres et de contrôles à effectuer, appelé mode opératoire, pour atteindre l'objectif élémentaire de l'action. La formalisation d'un mode opératoire en schéma de manœuvre (Figure 13), consiste à représenter sur le schéma organique d'une fonction principale, les manœuvres à effectuer, ainsi que l'état du circuit concerné par le mode opératoire. Le schéma des manœuvres complète les fiches opérateur et de terrain sur lesquelles sont explicitées les suites de contrôles et de manœuvres effectués depuis la salle de commande et sur le terrain, par l'opérateur et l'agent de terrain, ainsi que

les messages de coordination entre l'opérateur et l'agent de terrain et les métiers d'exploitation (chimiste, automaticien, …) nécessaires à l'accomplissement d'un mode opératoire.

Il est ainsi possible de représenter les modes opératoires des différentes situations de conduite : En consignation (pose et dépose de régimes), en arrêt de production (lignage de circuits[14], essais périodiques), en production (mise en service de GEF ou de fonctions principales).

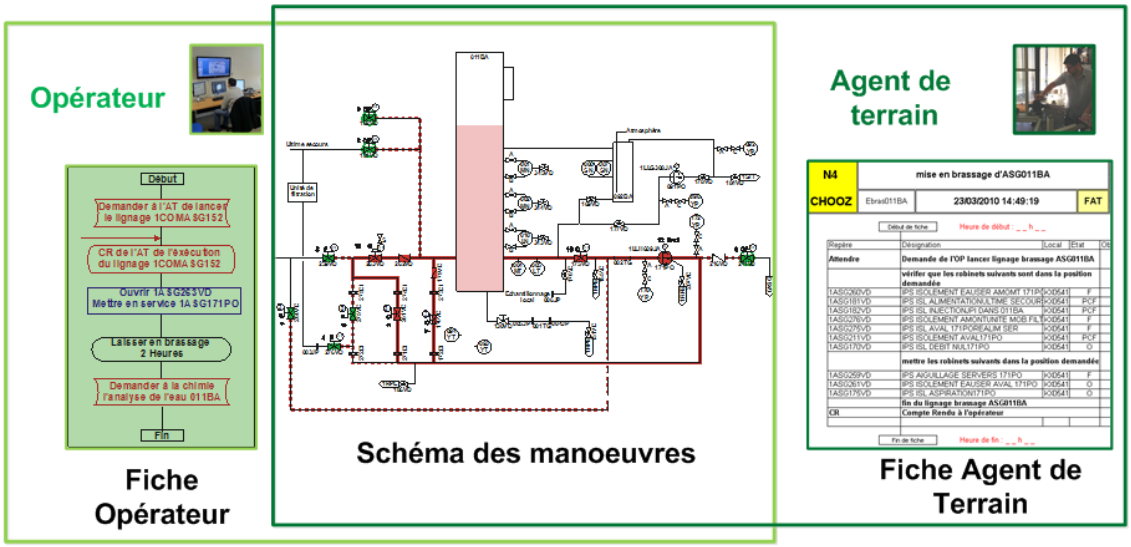


Figure 13 : Schéma d'exécution du mode opératoire de « mise en brassage de la Bâche ASG » (Viel, Cheriaux, & Galara, 2012)

1.3. Caractérisation de l'architecture-système de conduite du procédé industriel critique d'intérêt

Dans ce paragraphe, nous présentons dans un premier temps l'architecture de conduite traditionnellement utilisée selon une approche hiérarchique, pour démontrer les limites de cette organisation en exploitation. Dans un second temps, nous proposons une architecture distribuée pour la conduite, comme celle visée dans le cluster CONNEXION, afin de pallier aux problèmes rencontrés avec une architecture hiérarchique.

1.3.1. Caractérisation de l'architecture existante

La conduite d'un procédé industriel critique (**S**ituation) est répartie entre des agents humains (**U**ser) et technologiques (**A**rtefact) (Figure 7) organisés (**O**rganisation) pour exécuter des actions de conduite (**T**âche) sur le procédé afin de satisfaire la demande de production dans le strict respect des objectifs de sûreté, environnementale et de sécurité.

[14] Ligner un circuit signifie intervenir sur des ensembles de matériels (vannes, pompes, etc.) pour les configurer c'est à dire régler leurs positions (ouvertes, fermées, etc.) si celles-ci ne sont pas conformes à la position requise (Palaci, 2009)

Partie 1 : Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI

A l'heure actuelle, l'**O**rganisation mis en place dans les installations nucléaires repose sur une architecture par niveaux de hiérarchisation (Devic, 2014), aussi bien d'un point de vue technique (Figure 14a), que d'un point de vue humain (Figure 14b). Cette hiérarchisation repose sur le concept CIP (Computer Integrated Process) introduit par (Rao & Wang, 1993) et dédié aux processus de fabrication continu tels que les centrales nucléaires. Le concept CIP se présente comme un concept analogue au CIM (Computer Integrated Manufacturing) (Ranky, 1986) dédié au processus de fabrication discret. Le CIM repose sur un modèle pyramidale comportant une hiérarchie logique organisée en 4 niveaux correspondant à des niveaux de décision. Chaque niveau possède la capacité à formuler les objectifs locaux qui lui sont spécifiques, définir les actions à exécuter en se basant sur l'analyse de la situation courante. Néanmoins, chacun de ces niveaux est borné et contrôlé par les décisions du niveau supérieur en termes d'espace d'application de la décision, d'horizon temporel et d'effets de diffusion (Zarifian, 1990). Ainsi, plus on s'élève dans cette pyramide, plus le niveau de décision est important, plus la visibilité sur la situation dynamique est globale et plus les cycles opérationnels s'allongent.

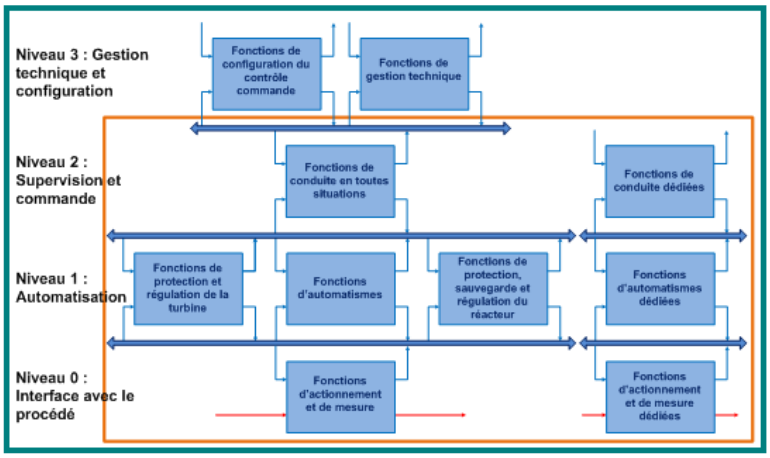


Figure 14a : Architecture hiérarchique du système de contrôle-commande (Appell & Chabert, 1998)

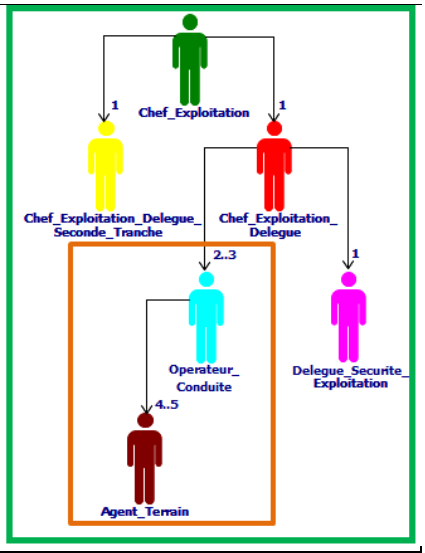


Figure 14b : Organisation hiérarchique équipe de conduite (Palaci, 2009)

Figure 14 : Organisation du système de conduite

Dans un premier temps nous détaillons la constitution de chacun de ces niveaux de hiérarchisation selon le point de vue technique correspondant au système de contrôle-commande (**A**rtefact) composé d'automates, de capteurs d'actionneurs, d'Interfaces Homme Machine (IHM),… (Figure 14a) :

<u>Niveau 0 :</u> Ce niveau assure l'interface entre le procédé physique à contrôler et le niveau des automatismes. Cette interface est constituée d'une part par l'instrumentation, et d'autre part par les actionneurs pouvant intégrer du traitement. Les informations issues des capteurs peuvent être logiques (fin de courses de vannes, présence de tension sur les tableaux, position ouverte ou fermée de cellule électrique, …) ou analogique (températures, débits, niveaux, pression, position des grappes de contrôle, intensité,…). Les ordres envoyés aux actionneurs peuvent aussi être analogiques (position d'une vanne réglante, hauteur dans une cuve, débit d'une pompe, …) ou logiques (demande d'enclenchement ou déclenchement d'une pompe, demande d'ouverture ou fermeture de vannes).

Niveau 1 : Ce niveau supporte les fonctions d'automatisation du procédé à savoir la mise en et hors service de l'installation, le maintien en régime et les variations de charge, la protection des équipements, de l'installation, du personnel et de l'environnement, la surveillance, les essais et la maintenance de l'installation. On distingue :

- Les fonctions de protection, de surveillance et de régulation liées directement à la sûreté (contrôle de la réactivité, du refroidissement et du confinement) et assurant la protection du réacteur.
- Les fonctions de protection, de surveillance et de régulation du groupe turboalternateur et du poste d'eau. Bien que ces fonctions ne soient pas classées sûreté, certaines d'entre-elles sont importantes pour la disponibilité de l'exploitation.
- Les fonctions d'automatismes des auxiliaires assurant l'automatisation, la surveillance du procédé, la protection des équipements et le maintien de la tranche dans un état stable

Niveau 2 : Ce niveau regroupe les fonctions de conduite, en salle de commande centralisée ou décentralisée permettant à l'opérateur de surveiller l'état et l'évolution du procédé, mais également d'agir sur celui-ci quelle que soit la situation de la tranche. Il comprend l'interface avec les opérateurs de conduite, ou Interface Homme-Machine qui, outre les moyens de commande et d'information (organes de commandes, indicateurs, enregistreurs, écran de visualisation) inclut les fonctions de conduite, de supervision ou d'aide à la conduite. En cas d'indisponibilité des systèmes de conduite principaux, les opérateurs disposent d'un panneau auxiliaire permettant de stabiliser l'installation et d'atteindre un état sûr de fonctionnement, avec des moyens conventionnels plus limités (verrines, indicateurs, relais de point de consigne et relais de commande à main, Tourner-Pousser-Lumineux (TPL), …). Dans le cas d'une indisponibilité complète de la salle de commande les opérateurs ont la possibilité d'utiliser un panneau de repli comprenant aussi des moyens de conduite limités.

Niveau 3 : Alors que les trois premiers niveaux se rapportent au contrôle-commande propre à une tranche d'une centrale nucléaire, ce niveau correspond aux fonctions communes à plusieurs tranches telles que la gestion technique, la gestion des opérations de maintenance, la surveillance météorologique du site et aux échanges avec des installations externes au site tels les dispatchings ou les centres nationaux de crise. Au-delà de ces fonctions centrales du contrôle-commande, il existe également des fonctions de contrôle-commande dites décentralisées qui concerne soit :

- Des fonctions autonomes sans liens avec les fonctions centralisées (la détection d'incendie, le contrôle des accès,…)
- Des fonctions pour lesquelles il n'a pas été jugé nécessaire de ramener leur supervision en salle de commande (production d'eau déminéralisée, traitements d'effluents,…)
- Des fonctions qui ne sont nécessaires qu'occasionnellement (auscultation de l'enceinte, …)

Dans un second temps, nous détaillons l'organisation hiérarchique de l'équipe de conduite structurée selon quatre niveaux fonctionnels et composée : d'un Chef d'Exploitation (CE), d'un Chef d'Exploitation Délégué (CED), de deux Opérateurs de conduite(OP), d'un Délégué Sécurité en Exploitation (DSE) et de quatre ou cinq Agents de Terrain (AgT) (Figure 14b). Notre étude se focalise plus précisément sur les opérateurs de conduite et les agents de terrains constituant les utilisateurs (**U**ser) de notre situation d'intérêt.

1 Chef d'Exploitation (CE) : supervise deux tranches, donc deux équipes de conduites au sein desquelles il assure un rôle managérial en tant que responsable hiérarchique. Il pilote les activités d'exploitation en situation normale, incidentelle ou accidentelle et en arrêt de tranche. Il contribue ainsi à la maitrise du niveau de sûreté, à la meilleure disponibilité des tranches, à la maitrise des couts d'exploitation et à la performance de

la conduite des installations. En parallèle, il occupe un rôle technique en supervisant les acteurs de la conduite, mais aussi un rôle de contrôle et surveillance de l'activité de conduite.

1 Chef d'Exploitation Délégué (CED) : seconde le Chef d'Exploitation de son équipe, contribuant à l'organisation du planning des activités du quart, à l'encadrement et au soutien technique de l'équipe de conduite. En ce sens, le CED assure la coordination des métiers de la maintenance et de la conduite. Il joue également un rôle dans la planification et le management des compétences des agents en élaborant les plans internes de formations des agents.

1 Délégué Sécurité en Exploitation (DSE) : est responsable de la sécurité des interventions sur le matériel. Il travaille dans le bureau des consignations, assurant la mise hors(ou en) service des équipements au préalable (ou suite) à la réalisation de travaux de maintenance de tranche en fonctionnement ou en arrêt. Il apporte une expertise « exploitation » aux différents métiers de la conduite.

2 ou 3 Opérateurs de conduite (OP) : surveillent et conduisent l'installation depuis la salle de commande. Chaque opérateur est responsable d'une partie de l'installation. Ainsi, l'opérateur réacteur (OPR) s'occupe du circuit primaire (du réacteur au générateur de vapeur) et l'opérateur eau-vapeur (OPEV) se charge du secondaire (du générateur de vapeur au groupe turbo-alternateur). L'activité des opérateurs de conduite se caractérise par plusieurs types de tâches (la conduite, la surveillance, l'encadrement des AgT, la coordination des différents intervenants et les essais périodiques) avec pour objectif de satisfaire au plus près les exigences de performance et de sûreté.

- Au cours de la conduite, les OP assurent l'adaptation du point de fonctionnement de l'installation aux besoins du réseau. Ils optimisent au maximum la production de leur tranche tout en restant dans le domaine de fonctionnement de l'installation.
- En parallèle, les OP effectuent la surveillance de l'installation, par le biais d'équipements et d'afficheurs de différentes natures (digitaux, papier, lumineux, informatique), en suivant l'évolution d'un certain nombre de paramètres. Ils doivent ainsi détecter, acquitter les alarmes et appliquer les mesures correctives adéquates si besoin.
- L'encadrement des agents de terrain : Certains paramètres ne pouvant être contrôlés depuis la salle de commande, les opérateurs font appel aux AgT afin d'intervenir directement sur un élément de l'installation. En ce sens, les OP coordonnent l'activité des AgT et contrôlent leurs comptes rendus.
- Les OP sont également en charge de la coordination des différents intervenants (internes ou extérieurs). En ce sens, ils contrôlent la compatibilité des besoins et des interventions de chacun avec les contraintes du planning du quart et anticipe tout phénomène induit sur le système.
- Finalement les OP exécutent quotidiennement des essais périodiques planifiés destinés à contrôler l'état de l'installation

4 ou 5 agents de terrain (AgT) : ils réalisent certaines manœuvres non réalisables depuis la salle de commande en intervenant physiquement sur les organes et équipements de l'installation. Leur travail se décompose en trois tâches principales : les rondes, les manœuvres d'exploitation et les essais périodiques :

- Les rondes sont de deux types : rondes d'observation et rondes de relevés. Au cours des rondes d'observations, l'AgT est affecté à une zone géographique de la tranche (salle des machines, bâtiment réacteur, etc.) dans laquelle il doit se déplacer et observer le matériel ainsi que la propreté de son environnement. Il fait alors appel à l'ensemble de ces sens (vue, ouïe, odorat, toucher) pour percevoir les écarts, les noter et les signaler aux OP ou CED en y associant une première analyse.

Durant les rondes de relevés, l'AgT doit collecter des informations (valeurs de paramètres, ou configuration) spécifiées qui ne sont pas relayés en salle de commande par les automatismes.

- Au cours des manœuvres d'exploitation les AgT manipulent le matériel de l'installation à la demande des OP. Il peut s'agir d'opérations de lignages, de modifications de configuration de circuits, de mise en ou hors services de matériels, … La plupart de ces manœuvres doivent être réalisées en suivant une procédure (gamme de lignage, fiche de manœuvre) indiquant la séquence des manipulations à effectuer.
- Les AgT effectuent également des essais périodiques au cours de leur ronde sur demande du CED ayant fourni au préalable la procédure correspondante.

Les retours d'expériences, ainsi que les tendances internationales montrent que ce type d'architecture hiérarchique a pour effet de filtrer les informations de conduite au niveau de la salle de commande, sans prendre en compte de manière intégrée les besoins des opérateurs en local (AgT) durant les interventions de surveillance, de conduite, de maintenance et d'entretiens d'arrêt de tranche (consignation, lignage, essais de remise en service). Chaque métier d'exploitation a une vision tronquée du procédé à conduire. Ainsi, pour accéder aux informations disponibles sur les interfaces du système de contrôle-commande, les autres métiers perturbent la sérénité de la salle de commande. L'accès à des informations complémentaires s'effectue par voie orale pouvant être sources d'ambiguïtés et entrainer le risque d'interprétations erronées.

1.3.2. Caractérisation de l'architecture visée

Il devient nécessaire d'aider d'avantage les équipes de conduite à actualiser facilement et rapidement leurs représentations de l'état de l'installation et de la conduite pour leur permettre d'anticiper les évolutions de paramètres et les actions à venir, et fournir ainsi une vision globale de l'installation en tout lieu et à tout instant. Cela nous amène à nous questionner sur une évolution des architectures hiérarchiques des systèmes de conduites des centrales nucléaires, vers une architecture distribuée (Figure 15) afin d'améliorer la disponibilité des informations (Neunreuther, 1998). Le terme distribué est utilisé pour désigner le fait que l'application ne soit pas centralisée. En ce sens, l'architecture résultant de la distribution des fonctions de conduite autour d'un bus d'exploitation, constitue une hétérarchie (Waldner & Duffin, 1992) dans laquelle la coopération de ces fonctions dépend de leur interopérabilité. Cette évolution n'est rendu envisageable qu'aujourd'hui de par l'évolution et la pénétration des systèmes numériques de contrôle commande dans le domaine du nucléaire.

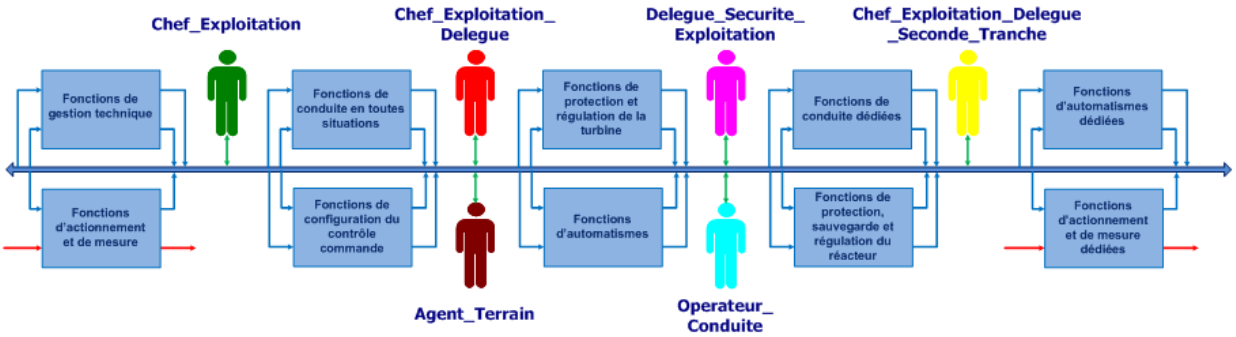


Figure 15 : Distribution des fonctions et des opérateurs de conduite autour d'un bus d'objet métier

Partie 1 : Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI

Divers travaux antérieurs du laboratoire, en collaboration avec la DER (Direction des Etudes et Recherches) d'EDF ont œuvré en ce sens dans le cadre d'importants programmes européens de R&D dans le domaine de l'énergie : ESPRIT **II-DIAS** (Distributed Intelligent Actuation and Sensors*),* **REMAFEX** (Remote Maintenance for Facility Exploitation) **ESPRIT III PRIAM** (Prenormative Requirements for Intelligent Actuation and Measurement), **ESPRIT III EIAMUG** (European Intelligent Actuation and Measurement User Group) et **ESPRIT IV IAM-PILOT** (Intelligent Actuation and Measurement Pilot). Ces derniers ont montré l'intérêt d'architectures techniques interopérables, distribuant au plus près du procédé une forme d' «intelligence technique» afin d'intégrer, en un système de conduite, de maintenance et de gestion technique (CMMS) [15], des systèmes d'actionnement et de mesure intelligents (IAMS) autour d'un bus de terrain (Figure 16a) (Iung, 1992), (Pétin, 1995), (Neunreuther, 1998), (Leger, 1999) et (Liu, 2002). Ces travaux ont récemment été revisités par (Dobre, 2010) pour proposer un Système Interactif d'Aide à la Conduite (SIAC) permettant aux agents de terrain d'interopérer en tout lieu, à tout instant et pour toute action de conduite avec le processus physique, via des objets interactifs « logiques » (*fournissant un ensemble de services aux métiers d'exploitation*) et « matériels » (*fournissant l'interface homme-machine*) (Figure 16b).

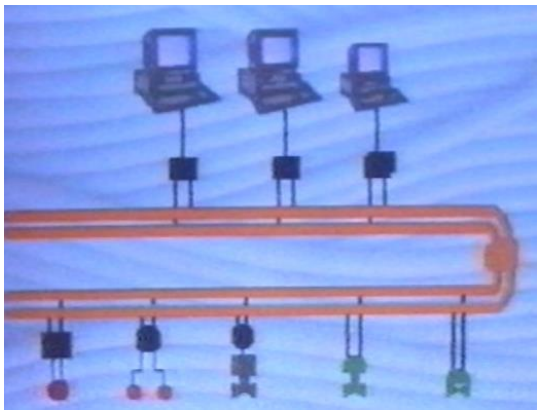
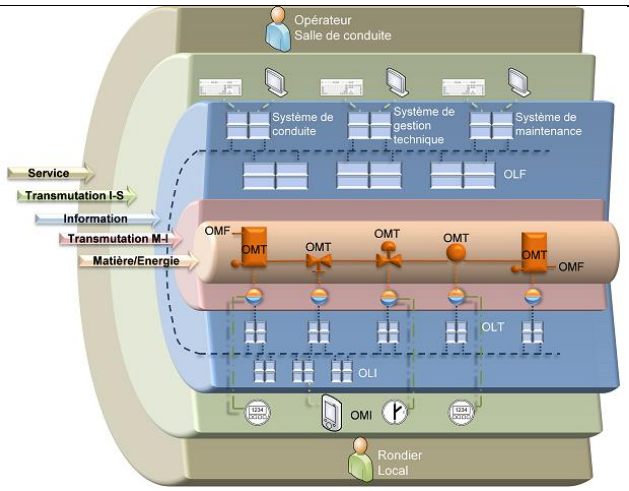


Figure 16 : Concept architectural DIAS (Source ENEL[16])



OMT : Objet Matériel Technique	OLF : Objet Logique de Flux
OLT : Objet Logique Technique	OMI : Objet Matériel Interactif
OMF : Objet Matériel de Flux	OLI : Objet Logique Interactif

Figure 16b : Concept architectural SIAC (Dobre, 2010)

Figure 16 : Concept architectural d'un système de conduite

Le concept architectural du SIAC encapsule le canal physique de flux de matière/énergie (*World of Physics*) par un bus d'informations numérique articulant un ensemble d'objets logiques (*World of Logic*) exécutant des traitements de données, afin de refléter dynamiquement et au mieux le comportement du procédé. Ainsi d'un point de vue global, l'objectif du SIAC est de développer l'interactivité numérique entre l'ensemble des constituants humains et techniques.

[15] Paradigme CMMS-IAMS (Integrated Control Maintenance and technical Management System – Intelligent Actuation and Measurement System) (AFIS - GT Ingénierie Système, 2005)
[16] https://www.enel.it/it-it

D'un point de vue plus détaillé, les objets logiques constituant le SIAC sont de deux types :

- **Techniques et physiques** : pour assurer la transmutation entre le canal physique et celui de l'information. Parmi ces objets, on trouve les Objets Logiques Techniques (OLT), qui reflètent le comportement des Objets Matériels Techniques (OMT) et assurent le filtrage des actions du rondier par rapport à la situation d'état du processus à consigner (Figure 17). C'est à ce niveau que l'on retrouve les constituants d'actionnement et de mesure pouvant éventuellement embarquer une forme d'intelligence technique. On trouve également, les Objets Logiques de Flux (OLF) représentant le comportement des Objets Matériels de Flux (OMF) et assurant le filtrage des actions du rondier par rapport à l'état du procédé, mais en se limitant cependant aux modes opératoires (MO). En ce sens, les OLF permettent la gestion des exclusions entre les MO, la transmission de la liste des actions au rondier, ainsi que l'affichage des actions à exécuter par l'opérateur.

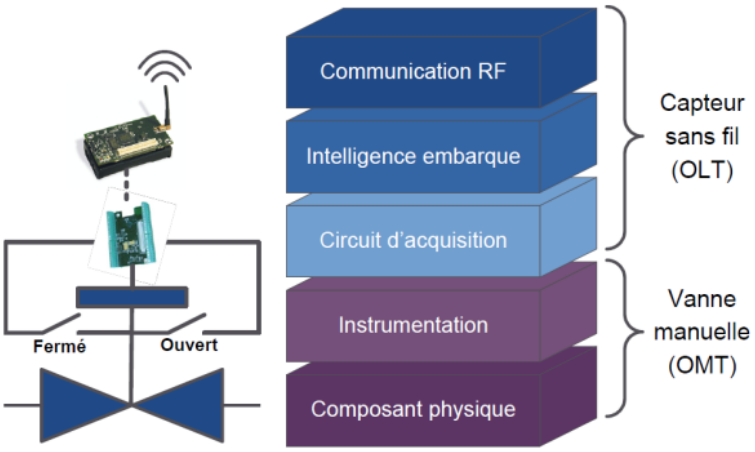


Figure 17 : Afin de pouvoir augmenter les capacités des vannes manuelle (OMT) un certain nombre de fonctions et de traitements (OLT) sont nécessaires, allant de la connexion physique des signaux de la vanne jusqu'à la communication (sans fil) des informations associées (Dobre, 2010)

- **Interactifs** : pour assurer la transmutation[17] entre le canal d'information et celui des services. Parmi ces objets, on trouve les Objets Logique Interactif (OLI) qui fournissent au rondier des services d'assistance au cours de ces activités, et des Objet Matériel Interactif (OMI) fournissant l'interface Homme-Système.

Les différents objets logiques composant le SIAC, avec leur comportement propre, interagissent entre eux dans un but commun à travers un système multi-agents implémenté à l'aide de la bibliothèque multi-agents JADE (Java Agent DEvelopment framework). Ainsi, l'architecture multi-agent du SIAC définis trois types d'agents (Figure 18) :

- Un agent guide d'exploitation (***GE***) permettant à l'opérateur (***Op***) de démarrer des modes opératoires (***MO***)
- Des agents ***MO*** reprenant chacun la connaissance relative à un mode opératoire et correspondant à un OLF
- Des agents ***assistant*** facilitant l'échange des messages entre les agents MO et le rondier

[17]La transmutation assure la transformation de nature

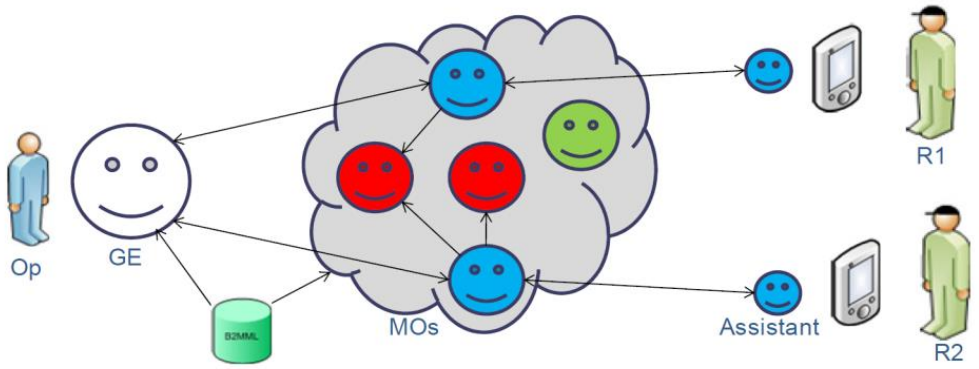


Figure 18 : Architecture multi-agents du SIAC (vert = agent disponible, bleu = agent en exécution, rouge = agent en exclusion) (Dobre, 2010)

1.4. Conclusion : Plate-forme CISPI constitutive de la situation système de conduite ciblée

L'objectif de ce chapitre était de présenter le contexte opérationnel d'un système de conduite interactive d'un procédé industriel critique. En ce sens, nous nous sommes intéressés à la situation système de conduite ainsi qu'à l'architecture distribuée visée pour ce type de système dont le but est d'assurer un continuum d'informations entre agents techniques et agents humains en situation opérationnelle de conduite aussi bien que d'ingénierie, dans une perspective globale de management du cycle de vie des données techniques de l'installation.

Afin de prendre en compte tous les phénomènes critiques émergents de ce type de situation, il nous a fallu retenir un cas d'étude qui reflétait au mieux cette réalité. En ce sens, notre choix s'est orienté vers la plate-forme CISPI (Conduite Interactive et Sûre de Procédé Industriel) du CRAN, simulant (*à une grandeur d'échelle moindre*) la fonction élémentaire d'alimentation de secours des générateurs de vapeur (ASG) et implémentant l'architecture visée à travers le déploiement d'un SIAC sur cette plate-forme. Cette plate-forme, issue du projet SAFETECH (Clanché, Gouyon, Dobre, Pétin, & Morel, 2010), se veut la plus proche de la réalité, en reconstituant des situations concrètes de travail en centrale, à travers des modes opératoires variées (*en production, en arrêt, en démarrage, etc.)* adaptés à la criticité des modes d'exploitation rencontrés (*conduite normale, conduite incidentelle et accidentelle*) et selon des constantes de temps différentes (*conduite en temps réelle, maintenance hors ligne*). Elle est constituée d'une partie processus physique (Figure 19) représentant le système en local, et d'une partie salle de commande (Figure 20) permettant d'interagir avec des équipements du procédé et d'en effectuer la supervision.

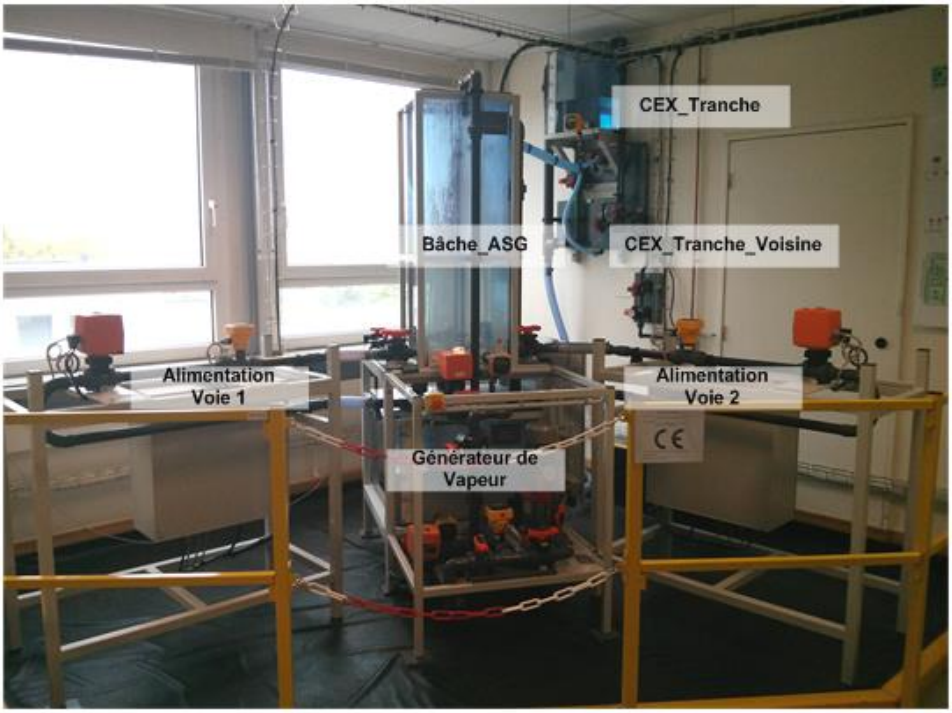


Figure 19 : Processus physique de la plate-forme CISPI à conduire (hydraulique) – Emulation du procédé ASG (Alimentation Secours des Générateurs de Vapeur)

Figure 20 : Salle de commande de la plate-forme CISPI du CRAN

Partie 1 : Contexte opérationnel et d'ingénierie interdisciplinaire du système de conduite CISPI

Une caractéristique essentielle de cette situation système de conduite concerne l'eau circulante (« flowing object ») donnant la vie au système à travers deux types d'interactions (Hall & Rapanotti, 2005):

- Causale : c'est-à-dire réagissant à des événements de façon prévisible et déterministe. Cela est notamment le cas avec les agents techniques de conduite
- Docile : c'est-à-dire pouvant réagir à des événements, mais de façon pas toujours prévisible ni déterministe. Cela est notamment le cas avec les agents humains de conduite

En ce sens, nous nous intéressons dans le second chapitre aux différents domaines d'ingénieries centrées technique et humain requis pour la conception de ce type de système « Machine-Homme ». Dans un second temps, nous montrons leurs limites dans la prise en compte global des phénomènes interdisciplinaires émergeants, ce qui nous amène à prôner une ingénierie centrée système.

2. Chapitre 2 : Contexte d'ingénierie interdisciplinaire du système pour faire CISPI

2.1. Introduction : Caractérisation de la situation-système d'ingénierie CISPI

La conduite d'un système critique reste majoritairement sous un contrôle humain (Devic & Morilhat, 2013). En effet, de par sa nature résiliente l'humain est capable de s'adapter à des situations inattendues, ce qui s'avère être primordial dans une organisation orientée sûreté (Hollnagel & Woods, 2005). Cependant l'étude de ces systèmes fait également apparaitre que l'humain ne peut les assumer toutes notamment celles répétitives, fastidieuses ou dans des environnements hostiles.

> *"Human reliability needs to be taken carefully from two points of view: (1) Human have limitations (2) Human are unique problem-solvers in unanticipated situations"* (G. Boy, 2011).

En ce sens, les systèmes de conduite d'un procédé industriel (***Process Part***) sont vus comme des systèmes socio-techniques où les opérations de conduite sont réalisées par l'interaction de deux constituants de natures différentes : humaine (***Human Part***) et technique (***Technical Part***) (Figure 21).

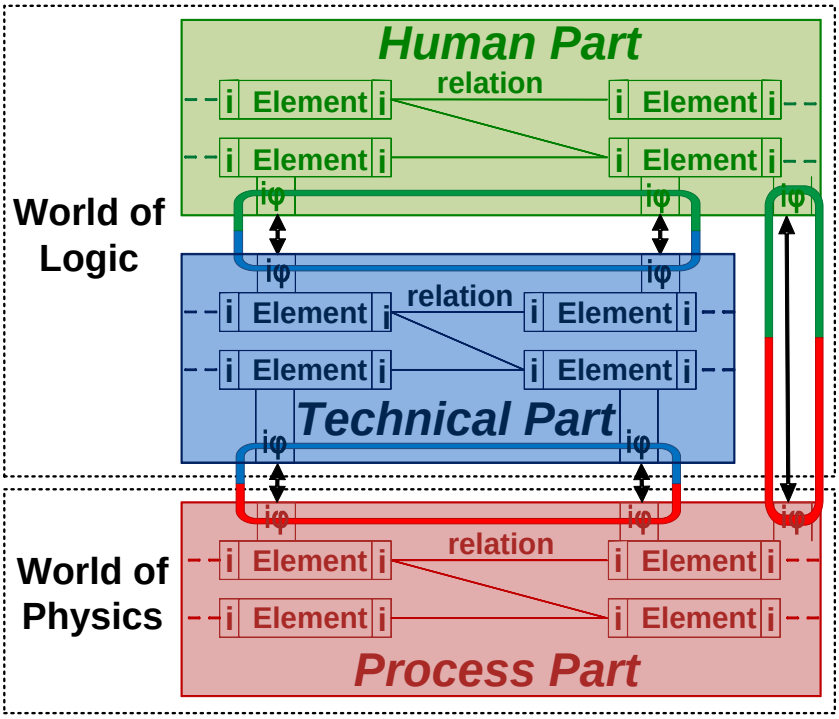


Figure 21 : Système de conduite d'un procédé en exploitation, adaptée de (Galara & Hennebicq, 1999)

Un constat récent a mis en évidence qu'il n'existe pas véritablement d'ingénierie centrée système à même de construire un modèle (M_H) partiel de la totalité d'une situation réelle, mais qu'il existe en revanche des ingénieries (*centrée technique ou centrée humain*) contribuant à la construction d'un ensemble de modèles constitutifs de parties (M_p) permettant la définition d'un modèle (M_h) total d'une faisabilité de la situation.

> *Despite the holistic approach of systems engineering (SE), systems still fail, and sometimes spectacularly…. Human-Centred Design (HCD) provides the creativity factor that SE lacks. It promotes*

> *modelling and simulation from the early stages of development and throughout the life cycle of a product. Unifying HCD and SE will shape appropriate Human System Integration (HSI) and produce successful systems (G. A. Boy & Narkevicius, 2014)*

Ainsi, nous explorons dans ce chapitre les apports potentiels des approches d'ingénieries centrées technique et humain, pour conclure sur les approches d'ingénieries centrées systèmes, les seules à même de pallier à la dichotomie courante entre ces ingénieries disciplinaires.

2.2. Caractérisation d'une ingénierie de conduite centrée technique

Comme le souligne (Devic & Morilhat, 2013), pour faire face à la complexité croissante des systèmes de conduite, il convient d'adopter une approche d'ingénierie système où l'on considère le système dans sa globalité selon différents points de vues interdisciplinaires.

2.2.1. Ingénierie système guidée par les processus

> *L'Ingénierie Système est une démarche méthodologique coopérative et interdisciplinaire qui englobe l'ensemble des activités adéquates pour concevoir, développer, faire évoluer et vérifier un ensemble de produits, processus et compétences humaines apportant une solution économique et performante aux besoins des parties prenantes et acceptable par tous. Cet ensemble est intégré en un système, dans un contexte de recherche d'équilibre et d'optimisation sur tout son cycle de vie (Fiorèse & Meinadier, 2012).*

La démarche d'ingénierie système s'appuie sur la définition et l'utilisation de processus formalisés dans différentes normes (IEEE 1220, 2005), (ANSI-EIA-632, 1999), (ISO/IEC 15288, 2008) par les associations d'ingénierie système INCOSE (International Council of Systems Engineering) et AFIS (Association Française d'Ingénierie Système) en collaboration avec de grands organismes de normalisation comme l'AFNOR, l'ANSI ou l'IEEE. Ces normes décrivent les activités à réaliser ainsi que les résultats à produire pour gérer un système[18] de son concept initial à sa mise à disposition en passant par son développement et sa réalisation. Ces trois normes sont complémentaires, en effet, elles recouvrent des champs différents avec des niveaux d'approfondissements différents.

La comparaison (Figure 22) proposée par l'AFIS nous permet d'évaluer :

- Le domaine de couvertures de ces normes (*axe horizontal des ellipses*) en termes de processus (techniques, management, contractuels, d'entreprises) mais également en terme de phase du cycle de vie de système
- Leurs niveaux de détail dans la description des activités du processus (*axe vertical des ellipses*)

[18] Un système est caractérisé par un ensemble de constituants (matériels technologiques, logiciels, opérateurs humains, matériaux, procédures, services) ; les constituants sont en forte interaction, et échangent des flux de matière, d'énergie et d'information dans un environnement ou contexte donné. Cet ensemble satisfait des besoins, des attentes ; il accomplit une mission assortie d'objectifs prescrits permettant de répondre à une finalité (Faisandier, 2014).

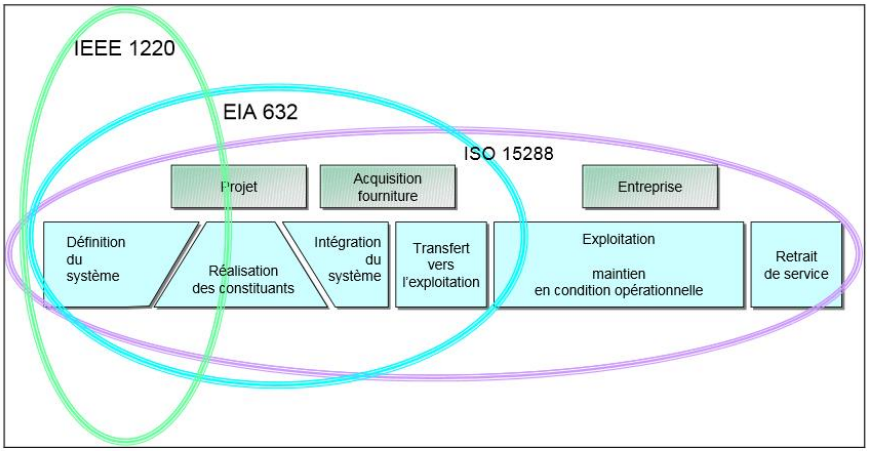


Figure 22 : Domaines couverts par les trois normes générales de l'ingénierie système (Fiorèse & Meinadier, 2012)

Le choix de guider l'ingénierie système à travers des processus s'explique par le fait que les activités à mener pour maîtriser la conception des systèmes restent invariantes quel que soit les projets et domaines d'application. Il est donc plus naturel de définir des processus plutôt que des cycles de vie[19] comme cela était fait précédemment : cycle en cascade (Royce, 1970), cycle en spirale (Boehm, 1988), cycle en V (Forsberg & Mooz, 1995). Ainsi, les processus constituent une vision opératoire de l'ingénierie système, complémentaire à la vision séquentielle des activités de management du projet quel que soit le cycle de vie retenu (AFIS - GT Ingénierie Système, 2005).

Ces normes classent les processus en quatre catégories (Figure 23) :

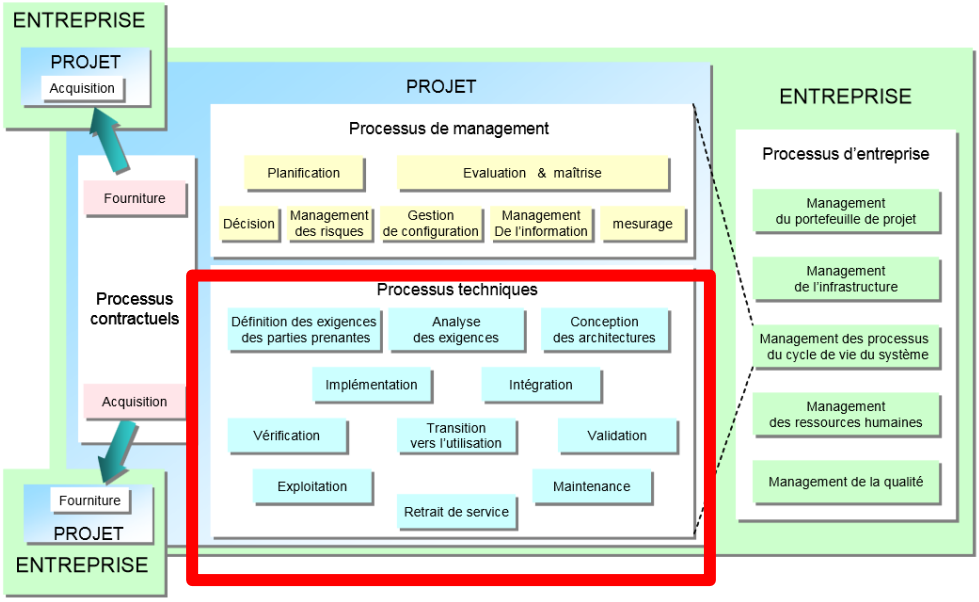


Figure 23 : Cartographie des processus du cycle de vie du système (ISO 15288) (Fiorèse & Meinadier, 2012)

[19] Le cycle de vie d'un système est l'ensemble organisé des activités afférentes à ce système tout au long de sa vie. La vie du système comporte différents états et les décisions de transition entre ces états. Un état (ou une situation de vie) du système est défini par une finalité, des activités, les résultats de l'exécution de ces activités (Faisandier, 2014).

- Les processus techniques qui participent à la transformation des besoins en solution et à la vie opérationnelle des produits correspondants
- Les processus de management (ou processus de projet) qui participent à la maîtrise des processus techniques dans le contexte du ou des projets (particulièrement de l'IS dans les projets)
- Les processus contractuels qui assurent les relations acquéreur-fournisseur (par exemple avec le client et les sous-traitants pour un projet dans une entreprise)
- Les processus d'entreprise qui ont pour rôle de développer le potentiel en IS de l'entreprise en manageant les domaines communs aux différents projets d'IS.

Trois ouvrages de référence (NASA, 2007), (INCOSE, 2010) et (Fiorèse & Meinadier, 2012) viennent compléter ces normes, constituant un référentiel d'ingénierie système reconnu par l'AFIS pour travailler aussi bien en approfondissement que pour servir d'aide à l'organisation de la connaissance (Le Put & Meinadier, 2009). Notons également que le corpus de connaissances en ingénierie système a récemment fait l'objet d'un consensus international dans le cadre du projet BKCASE (Body of Knowledge and Curriculum to Advance Systems Engineering) sous la forme d'un guide : SEBoK[20] (Systems Engineering Body of Knowledge) (Pyster, et al., 2012). La collection « Ingénierie et architecture des systèmes pluridisciplinaires » de Faisandier ainsi que les travaux du Centre d'Excellence sur l'Architecture, le Management et l'Economie des Systèmes (CESAMES[21]) confortent ce référentiel des bonnes pratiques en ingénierie système.

Dans la suite de ce mémoire, nos travaux se focalisent sur les processus techniques d'ingénierie système. Ces derniers s'exécutent de façon concourante par itérations successives et de façon récursive (Figure 25) à chaque niveau de décomposition du système étudié en sous-systèmes (Figure 24) à travers la notion de bloc-système[22] (« building block ») (ANSI-EIA-632, 1999). Quel que soit le niveau de cette décomposition arborescente, un ou plusieurs constituants terminaux implémentant une ou plusieurs technologies peuvent être définis. En ce sens, le travail de l'ingénieur système se termine par la définition (***Définition des Concepts***) et la description d'exigences pour chacun des constituants du système étudié. Chaque métier relatif à une technologie (*mécanique, électrique, hydraulique, logiciel, automatique,…*) collecte alors ces exigences et les consolide vis-à-vis de sa connaissance pour prescrire des exigences techniques (***Définition du système***) satisfaisant les exigences parties-prenantes du bloc système de niveau supérieur. Ces relations de description-prescription, ou en d'autres termes, ces « feedbacks » sont appliqués de manière récursive à chaque niveau de décomposition du système impliquant les mêmes processus génériques d'ingénierie système (*Définition des exigences parties prenantes, analyse des exigences, conception des architectures, vérification, validation, …*) et requérant des capacités d'interactions interdisciplinaires entre l'ingénieur système et l'ensemble des ingénieries métiers. (Dobre, 2010) définit cette relation de description-prescription comme un processus de spécification, se situant au centre du processus global d'ingénierie système et que nous étudions plus en détail dans le prochain chapitre de ce mémoire.

[20] http://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)
[21] http://www.cesames.net/cesames
[22] Bloc-système : Selon l' (ANSI-EIA-632, 1999) un système se présente sous la forme d'un arbre constitué d'un certain nombre de blocs systèmes. Cette structure de bloc décrit un système comme étant constitués de produits de deux sortes : les produits opérationnels (ou finals) qui réalisent les fonctions opérationnelles du système et les produits support assurant les fonctions de support du système (Micouin, 2011).

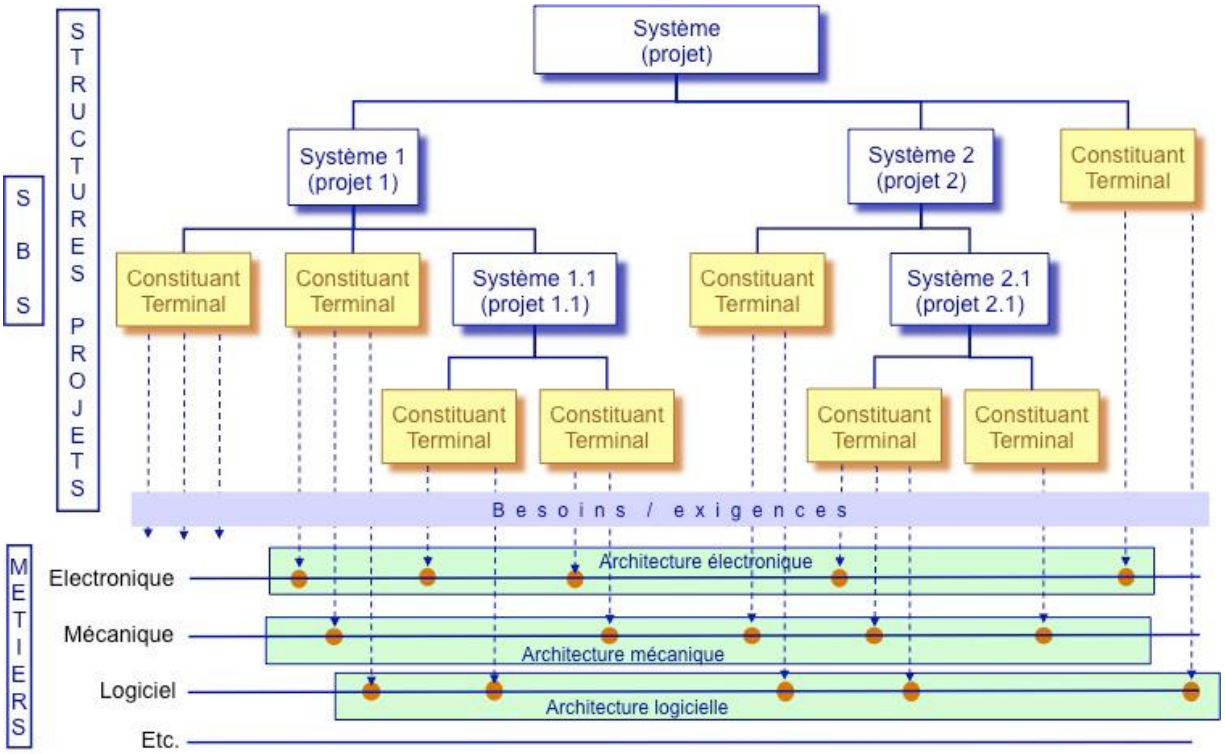


Figure 24 : Relations entre l'ingénierie du système et l'ingénierie des technologies (Faisandier, 2014)

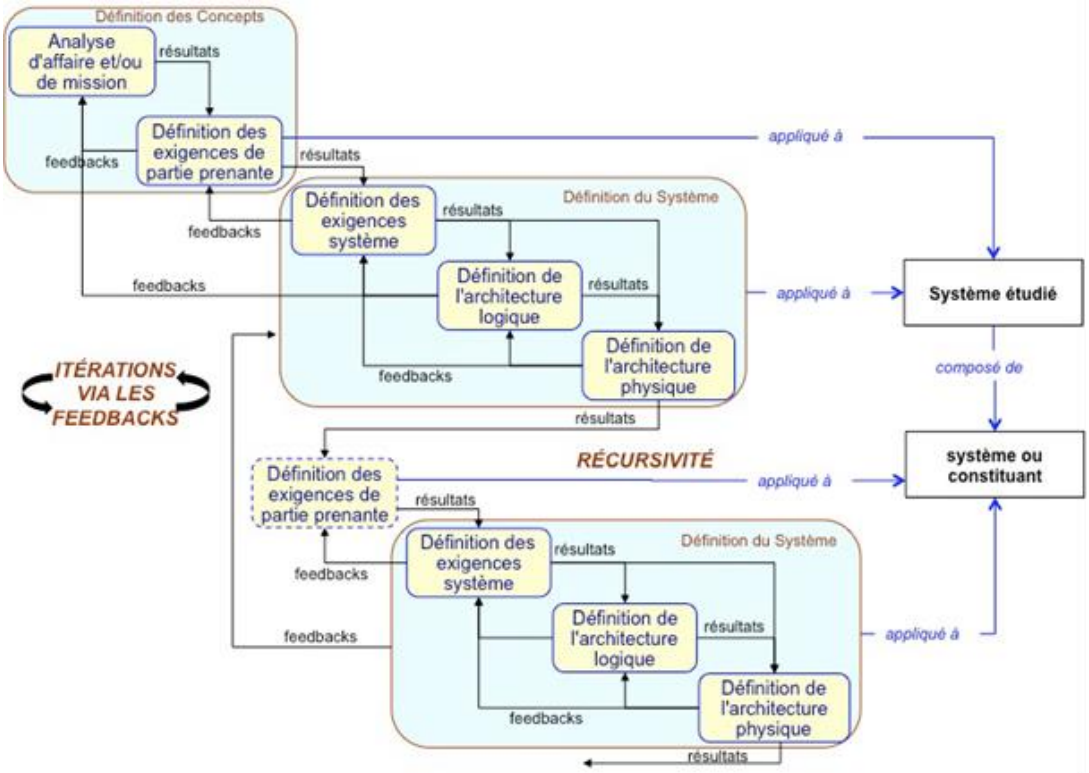


Figure 25 : Itération et récursivité des processus de définition (Faisandier, 2014)

2.2.2. Ingénierie système basée sur des modèles

2.2.2.1. Ingénierie système basée sur les documents versus Ingénierie système basée sur les modèles

Pour faire face à la complexité croissante des système, l'(INCOSE, 2007) attire l'attention sur la nécessaire évolution d'une ingénierie système traditionnelle basée sur les documents vers une ingénierie centrée sur les modèles (Figure 26) afin de prendre en compte la dimension collaborative et interdisciplinaire de l'ingénierie (Nugent & Collar, 2015). En effet, jusqu'alors les documents, en tant que résultats des processus d'ingénierie système, étaient le principal moyen de communication et de partage entre les différentes parties-prenantes d'un projet (Micouin, 2011).

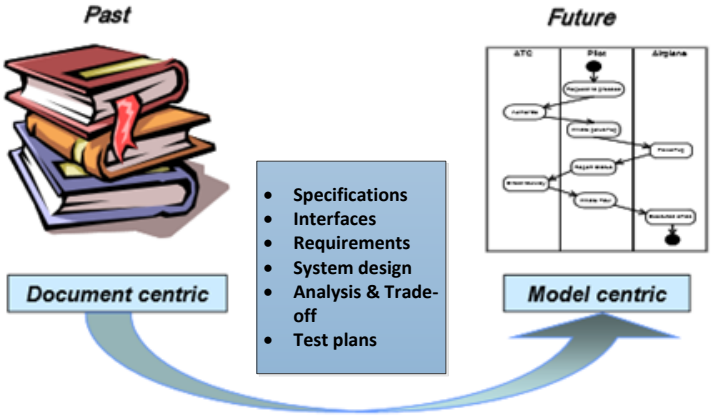


Figure 26 : Vision de l'INCOSE pour 2020 – Transition de l'ingénierie système basée sur les documents vers une ingénierie basée sur les modèles (INCOSE, 2007)

Bien que l'ingénierie système basée sur les documents ait montré son efficacité dans le passé, elle atteint ses limites dans la prise en compte des systèmes de plus en plus complexes, et pour lesquelles l'ingénierie basée sur les modèles apporte des réponses en simplifiant leur conception en automatisant un grand nombres d'activités, tout en améliorant la communication entre les différentes parties prenantes (Tableau 4).

		Conception système centrée modèles	Conception système centrée documents
Fonctions	Référentiel des informations	Modèles	Documents
	Revues	Par interrogation de modèles (automatisée)	Lecture et interprétation du texte puis comparaison
	Vérification	Implicite, incrémentale, automatisée, Co-simulation modèle local/ total	Processus d'audit humain
	Communication	Reproductible et consistante Un modèle de référence	Les réponses peuvent dépendre des perspectives des lecteurs
	Validation	Exécutée dans différents contextes Co-simulation modèle global	Révisions, revues de papiers
	Traçabilité	Intégrale	Au prix d'un effort intensif
	Réutilisation	Librairies, Patron de conception	Paragraphe standard uniquement

Tableau 4 : Comparaison des approches d'ingénierie système basées sur documents et basées sur les modèles – adapté de (Rochet, 2007)

L'Ingénierie Système Basée sur les Modèles (ISBM) « *Model Based Systems Engineering (MBSE)* » consiste en l'utilisation de langages de modélisation comme supports aux activités d'ingénierie système : conception, analyse, vérification et validation des systèmes, débutant de la phase de conception préliminaire et se poursuivant à travers le développement jusqu'aux dernières phases de vie du système (INCOSE, 2007). L'utilisation de modèles dans un processus d'ingénierie système présente de nombreux avantages (Friedenthal, Moore, & Steiner, 2008) :

- Capture de la connaissance spécialiste compréhensible et partagée avec l'ensemble des parties prenantes
- Amélioration de la communication entre les différents domaines d'ingénierie spécialiste impliqués dans le projet, les architectes systèmes et les clients
- Meilleure cohérence entre les différentes vues du système
- Capture des spécifications et des données de conception dans un format normalisé et neutre
- Exigences vérifiables (au plus tôt dans le processus d'ingénierie par co-exécution de modèles ou preuve formelle)
- Meilleure traçabilité entre les exigences, les données de conception, les analyses et tests
- Evaluation de l'impact des modifications facilitée
- Capitalisation de la connaissance, de patrons de conception, librairie de composants réutilisés dans plusieurs projets
- Réduction des erreurs (vérification syntaxique et sémantique des modèles) et tests d'intégrations rapides
- Génération automatique de documents

2.2.2.2. Modélisation d'architecture système

La modélisation apparait comme une technique au service de l'ingénieur système pour l'assister dans la définition de l'architecture d'un système. En ce sens, il s'agit pour l'ingénieur système de coupler l'ensemble des modèles d'ingénierie interdisciplinaires et constitutifs du système, à travers leurs multiples interactions. Cette modélisation systémique tient alors compte :

- Du point de vue structurel du système représentant les composants logiques ou physiques reliés par des liens d'interfaces entre eux et avec les éléments de l'environnement
- Du point de vue comportemental du système représentant ce que font ses éléments, l'enchainement de leurs interactions ainsi que les flux échangés.

Notons que cette modélisation du « système-à-faire » ($\boldsymbol{M_H}$) est classiquement effectuée de manière itérative où l'architecte-système modélise son système cible à l'aide d'un mécanisme de concrétisation progressive (Krob, 2009) consistant à modéliser le système selon trois niveaux d'abstraction (Figure 27) :

- Opérationnel (*centré sur l'extérieur du système*) : qui a pour but de définir le pourquoi du système, autrement dit le système est vu comme une boite noire rendant les services attendus à son environnement. D'un point de structurel, il s'agit de représenter la boite noire système dans son contexte, c'est-à-dire en relation avec les éléments de son environnement. D'un point de vue comportemental, il s'agit de représenter les scénarios de comportement du système vis-à-vis des éléments de son environnement.
- Fonctionnel (*centré sur l'intérieur du système*) : qui a pour objectif d'expliciter le fonctionnement du système indépendamment de la façon dont il le réalisera. D'un point de vue structurel, il s'agit de

représenter une vision statique décrivant l'agencement des fonctions ou blocs fonctionnels. D'un point de vue comportemental, il s'agit de représenter les enchainements d'activités de ces fonctions.

> *La simple connaissance des composants concrets d'un système et de leurs lois d'interaction n'est jamais suffisante pour décrire le comportement d'un système intégré résultant de leur interconnexion. Cela explique pourquoi il existe toujours des modèles purement fonctionnels d'un système dont le rôle est bien d'exprimer les fonctions émergentes que l'on ne pourra pas lire au niveau organique (Krob, 2012).*

- Organique (*centré sur l'intérieur du système*) : qui définit la façon dont le système est concrètement réalisé. D'un point de vue structurel, il s'agit de représenter l'organisation des composants matériels, logiciels et humains entre eux. D'un point de vue comportemental, il s'agit de représenter la dynamique de ces composants matériels.

  Notons que dans le paradigme RFLP (Requirement, Functional, Logical, Physical) (Kleiner & Kramer, 2013) que nous avons retenu pour ces travaux, le niveau organique se compose de deux niveaux d'abstractions :

  - Logique : qui définit comment le système est implémenté selon un agencement de constituants logiques représentent une abstraction des composants physiques
  - Physique : qui définit une définition virtuelle des composants physiques (du monde réel) au sein d'une maquette numérique physique du système

Nos travaux présentés dans la partie 3 de ce mémoire relatifs à la définition d'un modèle du « système-à-faire » ($\boldsymbol{M_H}$) de conduite de la plate-forme CISPI du CRAN, se situent au niveau d'abstraction fonctionnel. Ceux présentés en partie 4 se situent au niveau d'abstraction logique à travers la définition de modèles de «constituants à faire » ($\boldsymbol{M_h}$) d'actionnement et de mesure intelligents.

Bien que ces différents points de vue et niveaux d'abstractions contribuent à la modélisation du « système-à-faire » ($\boldsymbol{M_H}$), ils sont également source d'erreurs de cohérence et d'intégration entre modèles constitutifs de parties ($\boldsymbol{M_h}$) d'ingénierie spécialistes. Les cadres de modélisation de système peuvent apporter une solution à cette problématique pour se repérer face à la multitude de types de modèles. En ce sens, nous nous focalisons sur la synthèse récente des cadres de modélisation des architectures des systèmes d'informations et matériels (Zachman (Zachman, 1987), SAGACE (Penalva, 1997), MODAF (Ministry of Defence Architecture framework), DoDAF (Departement of Defence Architecture Framework),…) réalisée par (Krob, 2012) pour proposer un cadre de modélisation systémique (Figure 28). Ce cadre d'architecture système repose sur une définition d'un système selon un modèle mathématique (*Système formel*) qui nous est apparu en adéquation avec le système cible étudié (système de conduite de la plate-forme CISPI).

> *Un système formel S (*Figure 29*) est caractérisé d'une part par la donnée d'ensemble d'entrées X, de sortie Y et d'états internes Q et d'autre part les deux types de comportements suivants qui lient ces variables systémiques au cours du temps :*
>
> - *Un comportement fonctionnel qui produit une sortie y(t) Є Y à tout moment t du temps en fonction d'une entrée x(t) Є X et de l'état interne courant q(t) Є Q du système ;*
> - *Un comportement interne qui fait évoluer l'état interne q(t) Є Q du système au cours du temps t sous l'action d'une entrée x(t) Є X du système*

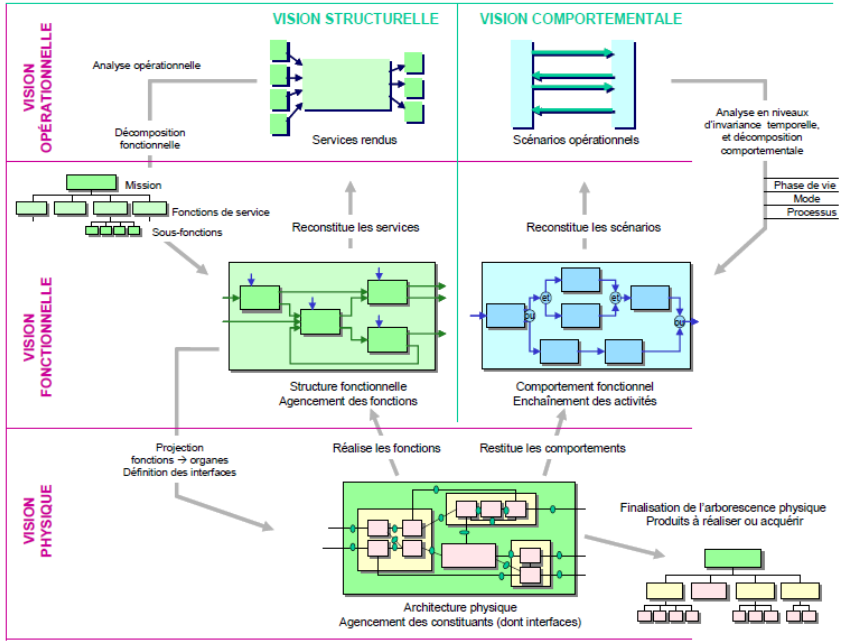


Figure 27 : Grille schématique de la représentation des systèmes (approche traditionnelle) (Fiorèse & Meinadier, 2012)

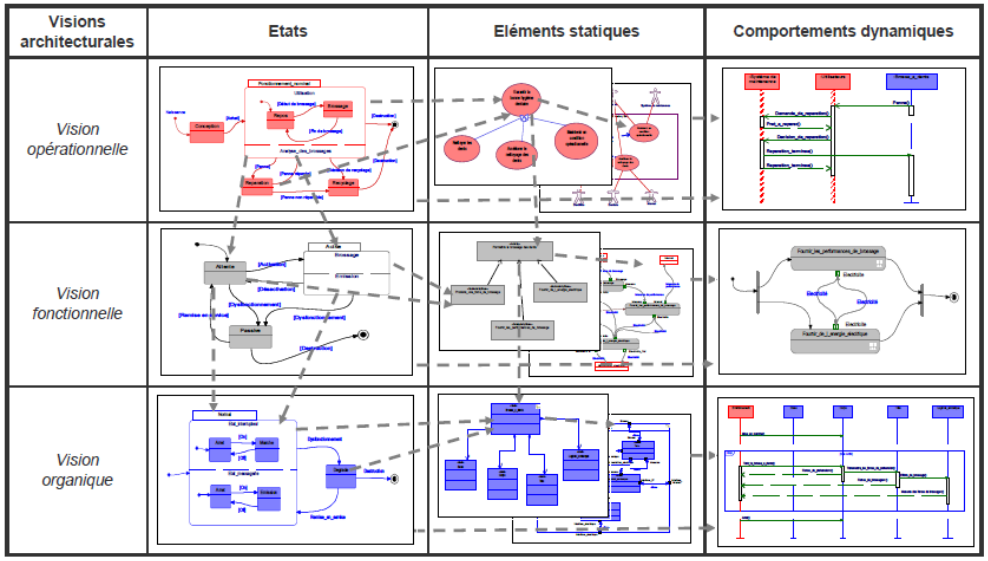


Figure 28 : Organisation générique d'un modèle « total » du système (Krob, 2012)

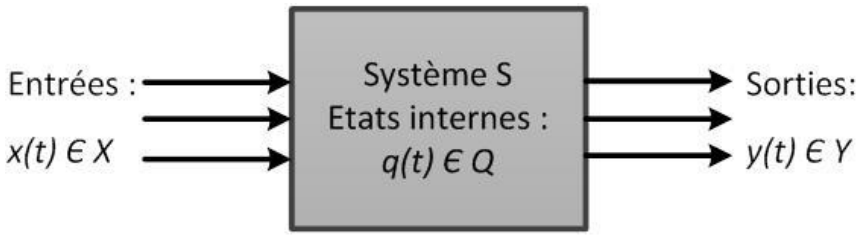


Figure 29 : Vision classique d'un système formel (Krob, 2012)

De cette définition formelle d'un système, (Krob, 2012) définit trois points de vues à considérer lors de la définition d'un système à chaque niveau d'abstraction d'architecture (opérationnelle, fonctionnelle et organique (vue dans ce cas à travers le niveau logique) :

- Etats du système : regroupant les représentations statiques (listant des états) et dynamique (évolution des états)
- Description statique des comportements du système (listant les composants ainsi que leurs entrées-sorties)
- Description dynamique des comportements du système (entrée/sortie)

A partir de ce cadre théorique (Figure 28), (Krob, 2012) identifie les artefacts de modélisation permettant de représenter les architectures d'un système industriel selon la quasi-totalité des langages de description architecturale. Parmi ces langages, il s'intéressent notamment au langage de facto de l'ingénierie système : SysML[23] (System Modeling Language) pour lequel il recommande l'utilisation des diagrammes suivants : Le diagramme de bloc (***systèmes et objets métiers***), la machine à états (***états***), le diagramme de bloc interne (***structure syntaxique d'un opérateur d'intégration***), le diagramme d'activité ou de séquence (***comportement dynamique induit par un opérateur d'intégration***) et le diagramme d'exigences (***conditions sur le comportement observable d'un système***). Selon nous, ces diagrammes permettent de représenter uniquement des systèmes discrets dont les comportements peuvent être représentés par un ensemble d'états. En ce sens, nous montrons dans le chapitre 5 de ce mémoire l'intérêt des diagrammes paramétriques SysML pour la modélisation des systèmes continus tels que le procédé physique ciblé de la plate-forme CISPI. .

2.2.2.3. Virtualisation d'un modèle d'architecture système

Un nouveau paradigme en ingénierie système introduit l'exécution du modèle ($\boldsymbol{M_h}$) total d'une faisabilité de la situation étudiée comme étant un vecteur d'intégration entre l'ensemble des modèles constitutifs de parties ($\boldsymbol{M_p}$).

> *A **paradigm shift** is occurring within Systems Engineering, where the level of complexity of system-projects can only accurately and precisely be modeled by a modeling construct that is executable Whole-system design means optimizing not just parts but the entire system. Everything must be considered simultaneously and teased apart to reveal mutually helpful interactions (Garcia, 2009) .*

[23] SysML (System Modeling Language) : est le langage de-facto de l'ingénierie système supportant les activités de spécification, d'analyse, de conception de vérification et de validation d'un large éventail de systèmes complexes (OMG, 2010). Nous détaillons dans le chapitre 4 le langage SysML.

Cela nécessite de la part de l'ingénieur système de « penser système (M_H) », en rupture avec l'analyse traditionnelle de « penser solution (M_I)», en se préoccupant de la façon dont les parties (M_h) interagissent pour reproduire ou transformer l'ensemble (HB Lawson, et al., 2005-2014) afin de satisfaire les exigences des parties prenantes. En ce sens, cette approche systémique sous-tend des capacités d'interactions interdisciplinaires pour architecturer l'ensemble des connaissances spécialistes encapsulées à l'intérieur des modèles (M_h). Ce travail est facilité aujourd'hui par des technologies numériques de modélisation et de simulation de plus en plus matures permettant la création d'un modèle virtuel de faisabilité (M_I) interdisciplinaire et exécutable du système pour en faciliter sa conception. Nous développons par la suite deux de ces techniques d'exécution de modèles :

Full-integration

La full-integration consiste à capturer l'ensemble des modèles spécialistes au sein d'un même outil, langage exécutant le comportement global du système. Cette technique présente les défauts de ses qualités : l'ensemble des modèles spécialistes sont facilement intégrables avec l'ensemble des autres modèles mais elle limite le choix des outils, langage et méthode d'ingénierie.

Parmi les travaux d'ISBM utilisant cette technique d'intégration système, nous nous sommes intéressés aux récents travaux de (Retho, 2015). Ce dernier montre l'intérêt d'un modèle d'intention exécutable en architecture-système comme pivot d'une collaboration multidisciplinaire agile (Figure 30).

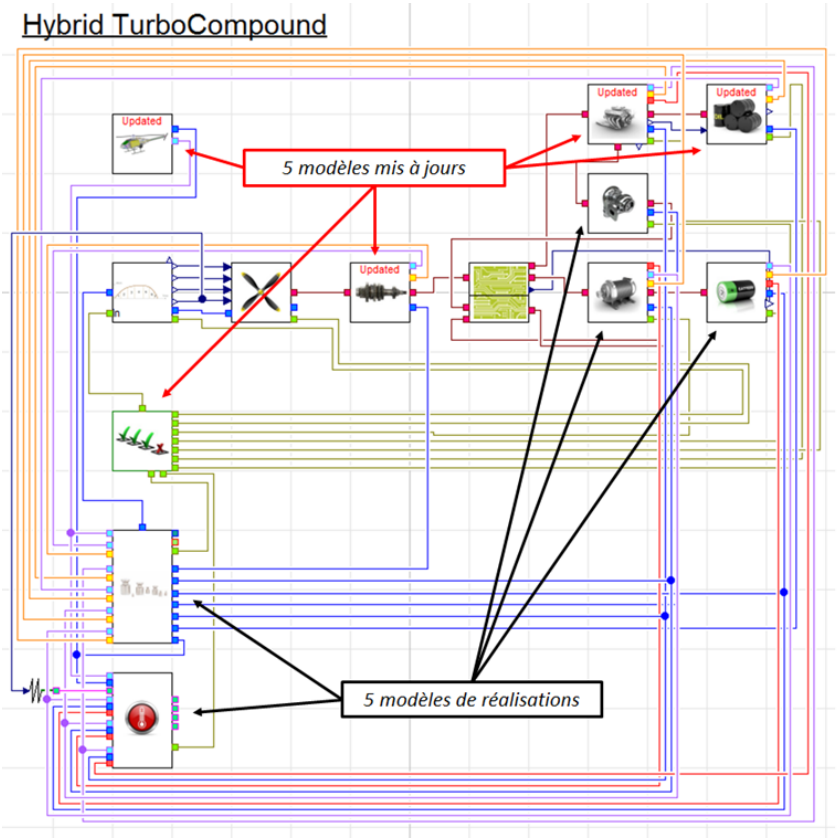


Figure 30 : Modèle d'intention (en vert) constitutif d'un modèle architectural multi-physique exécutable (Retho, 2015)

De façon plus détaillée, ces travaux s'intéressent au lien entre la conception totale conduite par le domaine d'ingénierie système et la conception basée sur la physique du monde réel au moyen de l'analyse des interactions et des impacts multidisciplinaires qui apparaissent dans le système. L'application de cette information pour la construction d'un modèle d'intention permet ensuite la requête de modèles comportementaux auprès de domaines d'ingénieries spécialistes. Afin de concentrer ces modèles d'ingénieries spécialistes autour de la construction du modèle d'intention, (Retho, 2015) fait le choix d'un langage unique d'intégration pour la représentation multidisciplinaire comportementale du système virtuel, à savoir le langage Modelica pour la modélisation des systèmes multi-physiques. A travers la recherche de connaissances vers les domaines d'ingénieries spécialistes pour construire un modèle virtuel du système, cette méthodologie est à la tangente de notre heuristique de co-spécification système développée dans ce mémoire. Néanmoins la modélisation du système virtuel ($\boldsymbol{M_I}$) ne nous apparait pas suffisamment intégrée à une modélisation système récursive pour comprendre le réel ($\boldsymbol{M_H}$). De plus nous estimons que le choix d'un langage unique pour la construction des modèles disciplinaires limite les domaines d'ingénieries spécialistes dans l'expression de leurs connaissances métiers. Une technique d'intégration système permettant de contourner cette problématique est la co-simulation de modèles exécutables.

Co-Simulation

La co-simulation consiste à exécuter l'ensemble des modèles d'ingénierie spécialistes dans chacun de leurs outils spécialistes dédiés. Ces modèles interopèrent les uns avec les autres autour d'un bus de co-simulation en charge d'orchestrer l'échange des données entre modèles. Cela va dans le sens du projet européen ITEA 2 (Information Technology for European Advancement) Modelisar où un consortium d'entreprises conduit par l'éditeur de logiciel Dassault Systems ont initié la création d'un nouveau standard[24] : FMI (Functional Mock-up Interfaces) pour l'échange de modèles et la co-simulation (Figure 31).

> The Functional Mock-up Interface (FMI) standard enables co-simulation which is the ability to dynamically interoperate multiple related models for improving model-based design between *original equipment manufacturer and supplier (Blochwitz, et al., 2012).*

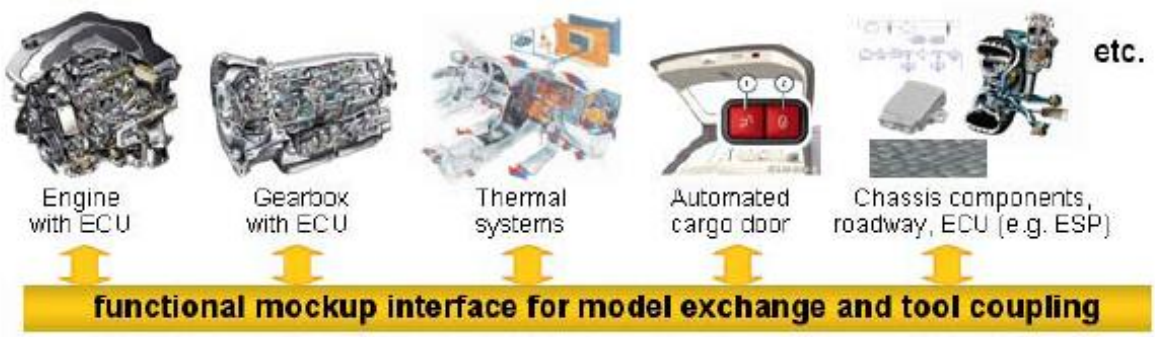


Figure 31 : Evolution interdisciplinaire de l'Ingénierie Système Basée sur des Modèles (Blochwitz, et al., 2012)

[24] https://www.fmi-standard.org/

L'objectif de ce standard est de fournir les règles d'interfaces entre différents environnements d'ingénierie afin d'évaluer par co-exécution de modèles un ensemble d'équations différentielles algébriques ou discrètes décrivant le comportement d'un système multidisciplinaire. Notons que ce standard apparait aujourd'hui comme un candidat prometteur d'un point de vue industriel pour la co-simulation et les collaborations inter-entreprises (Bertsch & Schulmeister, 2014).

2.3. Caractérisation d'une ingénierie de conduite centrée humain

L'ingénierie système s'est historiquement développée pour la conception de systèmes composés en grande partie de constituants techniques.

> *The classical systems (i.e., the system-as-machine paradigm) were small to large scale, multidisciplinary, relatively stable and predictable, without people as component, and were typically from the aerospace and defence industries. The new ones (i.e., the system-as-organism paradigm), which must cope with the global challenges of sustainable development, are large scale, complex, adaptive, interoperable, scalable, technology intensive, human integrative and comprise (Ramos, Ferreira, & Barceló, 2012).*

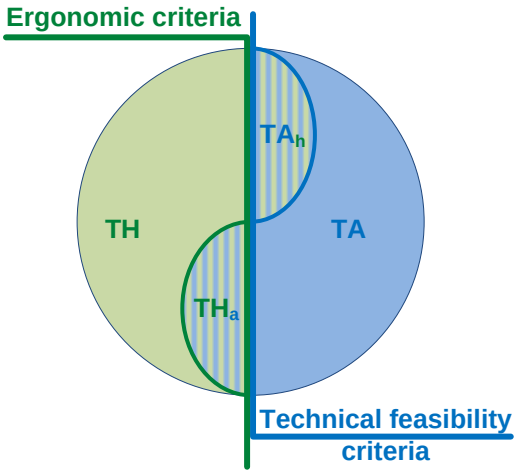
En ce sens, l'ingénierie système traditionnelle ne permet pas d'appréhender toute la complexité itinérante à la prise en compte des constituants humains composant les systèmes socio-techniques. Cependant, cette constituante humaine reste à l'heure actuelle majoritaire dans les systèmes de conduite étudiés, il est alors nécessaire de s'orienter vers des ingénieries centrées humains, afin de tirer profit de leur capacités, sans en subir les effets perturbateurs.

2.3.1. Intégration Machine-Homme

Cela nous amène à nous intéresser au cadre d'ingénierie pour les systèmes$_{\text{Homme-Machine}}$ critiques définit par (G. Boy, 2005): la pyramide AUTOS (Figure 5a) synthétisant un ensemble de travaux d'ingénierie centrée humain (Rasmussen, 1986), (Norman, 1986), (Amalberti, 1996), (Javaux & De Keyser, 1997) et (Vicente, 1999). Ce cadre d'ingénierie définis cinq domaines d'un système$_{\text{Homme-Machine}}$ ainsi que les interactions entre ces domaines à prendre en compte pour la conception centrée humain. Ainsi, appliqué à notre cas d'étude, le cadre AUTOS permet l'étude d'une ***Situation*** système de conduite de l'***Artefact*** de contrôle-commande de la plate-forme CISPI par des ***Utilisateurs*** (opérateurs humains) faisant partie d'une ***Organisation*** et exécutant différentes opérations (***Task***) de conduite. La pyramide AUTOS nous a notamment permis de caractériser la situation système de conduite de la plate-forme CISPI dans le chapitre 1.

Ce cadre d'ingénierie fait cependant apparaitre une dichotomie entre les ingénieries centrée technique (***Artifact***) et les ingénieries centrée humaine (***User***) au cours de la conception globale d'un système$_{\text{Homme-Machine}}$ où l'utilisateur final du système n'est pas perçu comme un constituant du système mais comme un élément en interaction avec ce dernier. Ce constat montre qu'il n'existe pas véritablement d'ingénieries centrées système à même de construire un modèle ($\boldsymbol{M_H}$) du « système-à-faire », mais plutôt un ensemble d'ingénierie (*centrée technique ou centrée humain*) contribuant à la construction d'un ensemble de modèles de « constituants-à-faire » ($\boldsymbol{M_h}$). En ce sens, cette vision système requiert une évolution des pratiques courantes d'ingénieries traditionnelles (Ruault, Vanderhaegen, & Luzeaux, 2012) afin de prendre en compte au plus tôt, des modèles de faisabilités techniques ou ergonomiques en réponse au dilemme de la répartition des fonctions de conduite entre des agents techniques ou humains (Figure 32).

Le dilemme du concepteur peut être résumé comme suit : d'un côté, il est tenté d'encadrer voire limiter l'activité humaine dans les situations connues pour se prémunir d'éventuelles erreurs, mais de l'autre, il ne peut compter sur l'inventivité humaine pour gérer l'inattendu (Millot, 2014).



- **TA** = tâches automatisées pouvant être effectuées par des agents techniques
- **TH** = tâches ne pouvant pas être automatisées pour des raisons techniques, théoriques, ou par manques d'informations et devant être allouées aux agents humains
- **TA_h** = tâches pouvant être effectuées aussi bien par un agent technique que par un agent humain
- **TH_a** tâches ne pouvant pas être effectuées par un agent technique seul ou un agent humain seul

Figure 32 : Répartition des tâches de conduite entre des agents techniques ou humains (Millot, Debernard, & Vanderhaegen, 2011)

Ce dilemme fait émerger le besoin d'un consensus pour concevoir les systèmes socio-techniques comme des systèmes techniques (Kroes, Franssen, Poel, & Ottens, 2006) en unifiant l'ingénierie système (*orientée technique*) avec l'ingénierie centrée humain « *Human Centered Engineering – HCE* » dans le but d'évoluer vers une intégration de l'humain et du système « *Human System Integration – HSI* ».

2.3.2. Virtualisation d'un modèle d'intégration Machine-Homme

A l'instar de l'ingénierie système orientée technique, (G. A. Boy & Narkevicius, 2014) recommande l'utilisation des techniques de modélisation et de simulation au plus tôt de la conception du système pour atteindre ces objectifs d'intégration.

Incorporating HSI into SE resulted in identifying and incorporating human capabilities and limitation in system development. Human Centred Design brings a methodology and philosophy to leading the design process prior to engineering and feeding forward solutions that can be successfully introduced into the society and organizations that will use the system. Requirements should be human centred, but the main asset of modern technology is to provide Modelling and Simulation from the very beginning of a design project (G. A. Boy & Narkevicius, 2014).

Des initiatives dans le domaine de la physiologie permettent d'envisager la construction de modèles exécutables du comportement « physiologique » humain. Cela se concrétise par le développement d'un outil : PhysioMatica™ (Bennani, Chauvet, & Dupont, 2010) pour la modélisation et la simulation des fonctions physiologiques intégrés du vivant, ou bien encore la création d'une bibliothèque Modelica : Physiolibrary (Mateják, et al., 2014) contenant des lois physiques de base régissant la physiologie humaine. Cette librairie est ainsi utilisable pour la circulation cardio-vasculaire, les processus métaboliques, la distribution des nutriments,

la thermorégulation, le transport de gaz, la régulation de l'électrolyte, la distribution de l'eau, la régulation hormonale et la réglementation pharmacologique. Notons cependant que l'ensemble de ces initiatives restent dans le domaine de l'ingénierie centrée humain sans prendre en compte les interactions avec les systèmes techniques contrairement aux travaux de (Lieber, 2013) relatifs à l'ingénierie d'un système support de maintenance aéronautique et auxquels nous avons contribué. Lieber propose ainsi une méthode pour s'assurer que le comportement des $\text{systèmes}_{\text{Machine-Homme}}$ dépendant des synergies entre les différentes interactions prenant place entre les constituants techniques et humains, soit maintenu dans un domaine de performances acceptables. En ce sens, le paradigme exploré se fonde sur la possibilité de faire interopérer des modèles de processus physiologiques avec des modèles de processus techniques en spécification d'interaction Machine-Homme, en combinant un cadre de modélisation système avec celui mathématique et computationnel de la physiologie intégrative (que nous détaillerons dans le chapitre 8 de ce mémoire). L'intégration de ces modèles techniques (*SysML*) et humains (*Simulink*) a été mis en œuvre par une technique de code-generation afin de s'assurer par simulation du respect des performances du système global. Cette technique requiert que l'ensemble des modèles d'ingénieries spécialistes soient transformés en langage de programmation (*C/C++*), pour être ensuite importés dans un outil d'ingénierie système exécutant de façon intégrée l'ensemble du code généré à partir des différents modèles. Notons que cette technique, dépendante des transformations de modèles vers des langages de programmation au sein des différents outils d'ingénieries, ne permet pas d'assurer que le code généré reflète correctement le comportement spécifié.

2.4. Caractérisation d'une ingénierie centrée système

Le dilemme évoqué précédemment soulève la question de déterminer au cours de la conception d'un système qualifié de complexe tel que les systèmes sociotechniques de conduite d'un procédé industriel critique, les critères permettant de trancher sur une alternative d'architecture de conduite plutôt qu'une autre. Il apparait que l'ingénieur centré système ne peut se contenter de la seule connaissance de ces faisabilités techniques ou ergonomiques. Il est alors primordial pour lui de prendre une décision en bouclant sur la réalité d'où proviennent les exigences parties prenantes et dans laquelle sera implémenté le futur système. En ce sens, nous caractérisons une ingénierie centrée système en tant qu'ingénierie bouclée à la fois sur la réalité (domaine opérationnel) et sur ses constituants (domaines spécialistes).

Cette vision centrée système rejoint l'approche DSA[25] (Distributed Situation Awareness) (PM Salmon, Stanton, Walker, & Jenkins, 2009) initialement issue de l'étude des processus cognitifs en environnement opérationnel, que nous transposons à un environnement d'ingénierie collaborative (Yufei & Kazuo, 2005). Ainsi, la DSA considère les systèmes sociotechniques comme un TOUT (une entité) en interaction avec son environnement en termes de phénomènes ou de stimulis. En ce sens, cette approche revisite l'approche SA (Situation Awareness) en la considérant comme une propriété émergente des systèmes complexes sociotechniques plutôt qu'une initiative individuelle (Artman & Garbis, 1998). La conscience de la situation (SA) est considérée comme la connaissance d'une organisation dynamique maintenue grâce à des informations recueillies par un individu à travers la perception de l'environnement de la situation système. La SA est ainsi vu comme le produit résultant d'un processus d'évaluation de la situation comprenant les trois étapes suivantes (Figure 33) (Endsley, 1995):

- Niveau 1 (***Level 1***) : Perception des éléments de l'environnement dans un espace-temps
- Niveau 2 (***Level 2***) : Compréhension de la situation courante

[25] DSA (Distributed Situation Awareness) = conscience de la situation distribuée

- Niveau 3 (***Level 3***) : Projection de leurs états dans le futur

Les approches DSA supposent que les systèmes sociotechniques possèdent des propriétés cognitives (comme la conscience de la situation) qui ne sont pas une partie d'une cognition individuelle mais le résultat de propriétés émergentes d'un système cognitif collaboratif vue comme un TOUT. (Artman & Garbis, 1998) suggère que la SA d'une équipe soit distribuée non seulement aux agents humains (ingénieurs) constituants l'équipe, mais également aux agents techniques (Artefact) utilisés dans le but d'atteindre la finalité du système. Par exemple dans la Figure 34 « quatre agents humain travaillent de façon collaborative mais sont répartis les uns par rapport aux autres. La conscience de la situation (SA) est initialement à un niveau système pour que ce dernier fonctionne de façon efficiente. Cette information est ensuite partagée entre l'ensemble des agents humains assistés d'agents techniques. En ce sens, les approches DSA ne prétendent pas que les perspectives individuels sont redondantes, elles fournissent des approches complémentaires pour décrire une SA dans un environnement collaboratif » (P Salmon, et al., 2008).

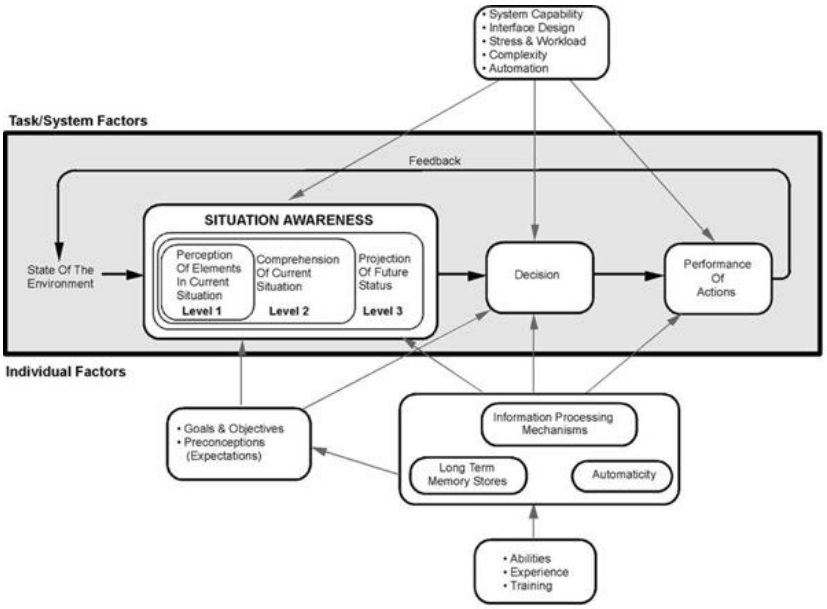


Figure 33 : Modèle d'un processus d'évaluation d'une situation (Endsley, 1995)

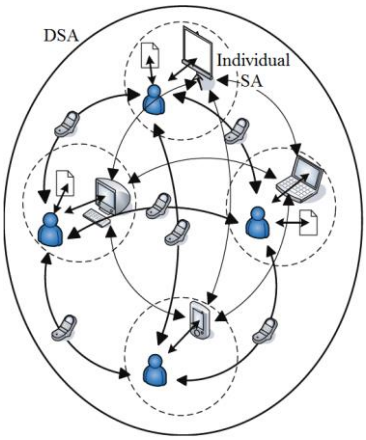


Figure 34 : Exemple d'une DSA (Distributed Situation Awareness) (P Salmon, et al., 2008)

2.5. Conclusion : Environnement de modélisation centré sur le système CISPI constitutif de la situation-système d'ingénierie ciblée

L'objectif de ce chapitre était de présenter le contexte d'ingénierie interdisciplinaire pour la conception d'un système$_{\text{Machine-Homme}}$. En ce sens, nous nous sommes focalisés sur les ingénieries centrées technique et humain afin de montrer leurs limites dans la prise en compte de l'intégration des agents techniques avec des agents humains. Nous avons également statué sur l'apport d'une ingénierie basée sur des modèles exécutables et centrée système afin de pallier la dichotomie courante entre ces ingénieries interdisciplinaires. En ce sens, une ingénierie centrée système s'appuie sur un ensemble de modèles de faisabilité interdisciplinaires ($\boldsymbol{M_h}$) pour construire un modèle virtuel de faisabilité ($\boldsymbol{M_I}$) du système au regard d'un modèle ($\boldsymbol{M_H}$) d'une situation opérationnelle réelle.

Il ne faut pas oublier que derrière ces modèles, se cachent un ensemble d'ingénieries spécialistes et de ce fait un ensemble de connaissances spécialistes. Afin de ne pas contraindre ces ingénieries par un outil, un langage ou une méthode lors de la formalisation de leurs connaissances métiers sous la forme de modèle de faisabilités, nous retenons la technique de co-simulation pour l'intégration de ces différents modèles (Figure 35). En effet, cette technique d'intégration système offre l'avantage aux différentes ingénieries spécialistes de rester dans leurs domaines d'expertises sans avoir à partager leurs méthodes, outils et modèles d'ingénierie des parties , tout en protégeant leur propriété intellectuelle.

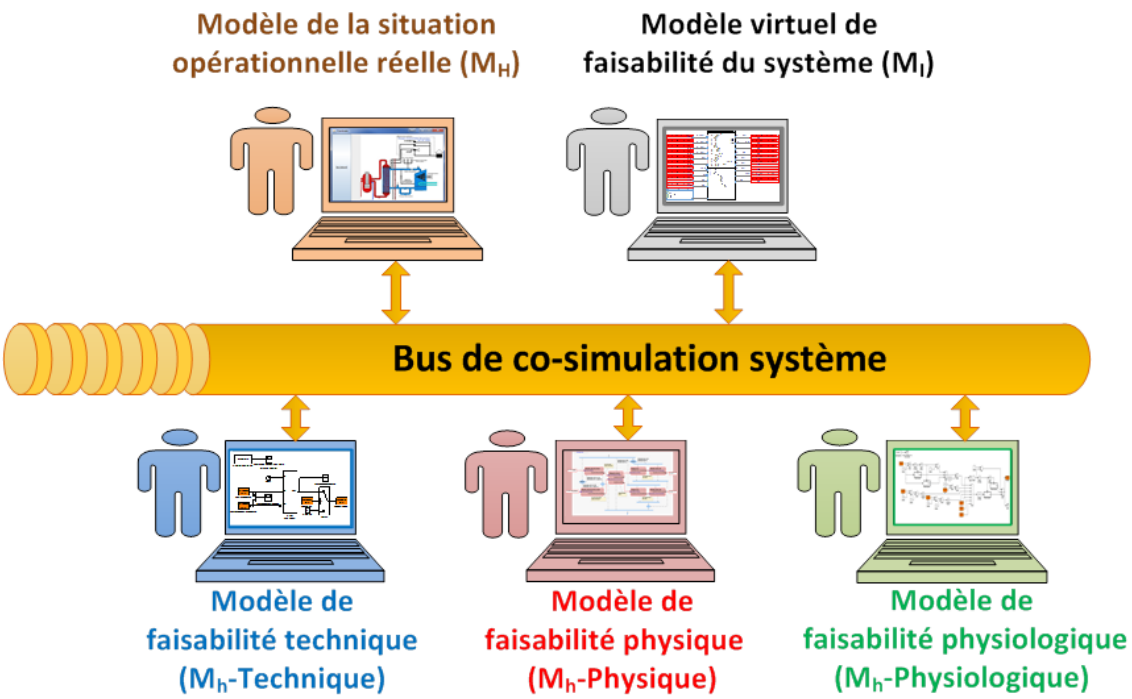


Figure 35 : Environnement collaboratif d'ingénierie centrée système basée sur des modèles exécutables

Conclusion : Environnement collaboratif d'interactions entre domaines d'exploitation et d'ingénierie constitutive de la situation-système d'intérêt ciblée

Dans cette première partie du mémoire, nous venons de contextualiser à la fois le domaine opérationnel de conduite interactive d'un procédé industriel critique en exploitation, ainsi que les domaines d'ingénierie requis pour concevoir ce type de système$_{\text{Homme-Machine}}$.

De manière plus détaillé, nous avons présenté dans le chapitre 1, les caractéristiques essentielles d'une situation système de conduite d'un procédé industriel critique que nous avons réduit à la plate-forme CISPI émulant une fonction élémentaire d'alimentation de secours des générateurs de vapeur (ASG) et implémentant l'architecture visée d'un SIAC. Ainsi, ce système de conduite majoritairement sous le contrôle d'agent humain a pour mission de maitriser l'eau circulante « flowing object » à travers la plate-forme CISPI afin de refroidir les générateurs de vapeurs en situation incidentelle ou accidentelle.

Dans le second chapitre, nous avons statué sur la mise en œuvre d'une ingénierie centrée système basée sur des modèles exécutables afin de construire un modèle virtuel de faisabilité ($\boldsymbol{M_I}$) du système intégrant un ensemble de modèles interdisciplinaires exécutables de faisabilité ($\boldsymbol{M_h}$) afin de satisfaire un modèle ($\boldsymbol{M_H}$) d'une situation opérationnelle de conduite satisfaisant les exigences parties prenantes. Cette ingénierie centrée système contribue notamment à pallier la dichotomie courante entre les métiers d'exploitations et d'ingénieries en exécutant et en couplant, autour d'un bus de co-intégration système, les modèles spécialistes de faisabilité avec l'environnement réel de conduite, au travers du domaine de l'ingénierie centré système tourné à la fois vers le monde réel et le monde de l'ingénierie, afin de converger de manière récursive vers une solution optimale (Figure 36).

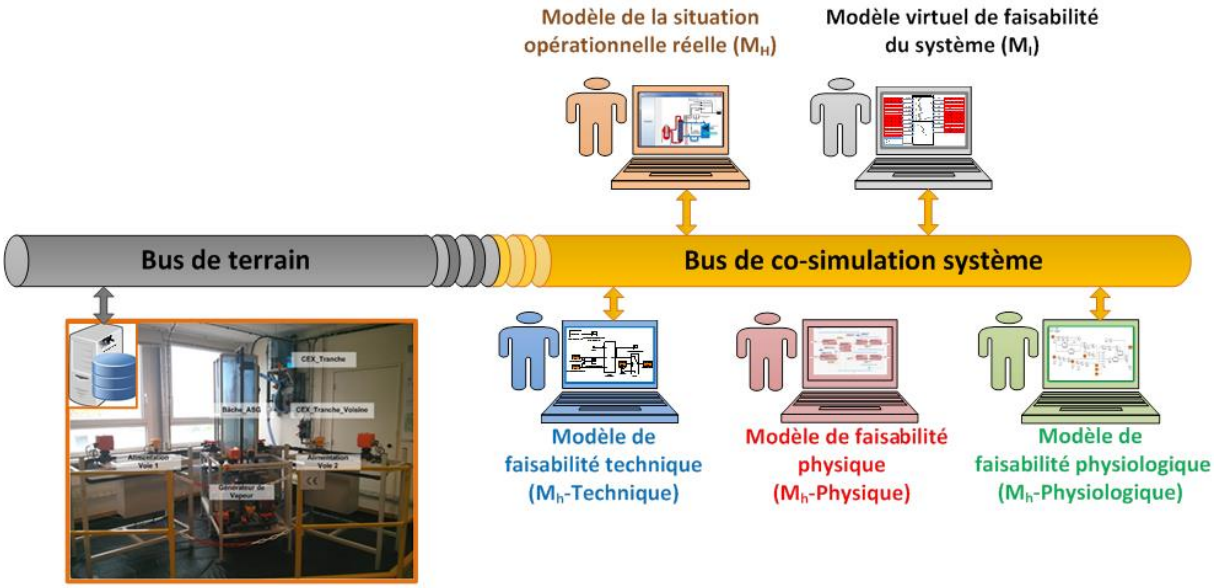


Figure 36 : Environnement collaboratif de co-intégration système

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

Introduction : Interaction collaborative entre métiers d'ingénierie interdisciplinaire constitutive du système d'intérêt ciblé

L'objet des deux chapitres de cette seconde partie est de définir notre approche heuristique de co-spécification holistique d'un SYSTEME et comment la mettre en œuvre à travers un environnement de co-simulation basée sur des modèles, en nous limitant au niveau d'abstraction fonctionnel (Cf. Annexe 3-1) de notre cas d'étude.

L'objectif est de faciliter la construction d'un modèle du **SYSTEME d'intérêt** en mettant l'ensemble des parties prenantes interdisciplinaires en situation itérative de modélisation (« System model in the Loop ») pour contribuer à chacun des éléments formant un ensemble de modèles en interaction.

Comme le montre cette discussion récente (Pyster, et al., 2012) cette notion demeure ambigüe puisqu'elle réfère aussi bien à la situation opérationnelle dont le système d'intérêt est une abstraction construite par le modélisateur système avec un ensemble d'éléments de modélisation (connaissance) systémique (2), qu'au système opérationnel dont le SYSTEME d'intérêt est une concrétisation construite par l'ingénieur système avec un ensemble d'éléments de modélisation (connaissance) dont une partie importante appartient aux domaines de spécialistes.

> ***System-of-interest*** *(1): the system whose life cycle is under consideration (Von Bertalanffy, 1968) a collective set of all elements of any system considered by a lifecycle, this may include both operational and enabling systems. Strictly, this definition is relevant for an engineered system of interest.*
> ***System-of-interest*** *(2): to an observer (Von Bertalanffy, 1968). It is the system science definition and applies to all systems irrespective of whether they have a purpose or lifecycle in an engineering sense.*

Afin de concilier (2) et (1), nous pouvons formuler différemment notre objectif en considérant qu'il s'agit de faciliter la construction d'un ensemble de connaissances d'un **SYSTEME d'intérêt** en mettant les parties prenantes interdisciplinaires en situation itérative de modélisation (« System knowledge in the loop ») pour contribuer à chacun des éléments de connaissances représentés par un ensemble des modèles en interaction.

Il s'agit donc d'une situation d'Ingénierie Système Basée sur des Modèles (ISBM) à laquelle nous conférons l'objectif sous-jacent d'assurer au plus tôt la Vérification/Validation de cette construction itérative d'un SYSTEME d'intérêt par co-exécution de modèles.

Aussi nous proposons dans le chapitre 3 une approche heuristique de co-spécification holistique d'un SYSTEME basé sur un ensemble de boucles (« feedback ») de recherche de ces connaissances (Figure 37) :

- orientées-problème du domaine de l'ingénierie-système vers le domaine « source » d'intérêt et vice versa ;
- orientées-solution du domaine de l'ingénierie-système vers les domaines d'ingénierie-spécialiste « puits » d'intérêt.

La dimension holistique de notre approche est guidée par un raisonnement systémique selon (2) qui s'appuie sur un paradigme holonique de modélisation pour percevoir le SYSTEME d'intérêt étudié comme :

- TOUT : à travers ses interactions avec l'environnement (connaissances sur l'extérieur) ou

- PARTIE : à travers sa structure interne (connaissances des parties requises pour l'architecturer) tout en assurant un continuum de connaissances et modèles vers le développement SYSTEME selon (1).

La dimension itérative de notre approche s'appuie sur l'approche des « problem frames » pour étendre le paradigme de spécification d'exigences en génie informatique à celui d'une co-spécification SYSTEME itérative et interdisciplinaire basée sur des modèles.

En ce sens, nous détaillons dans le chapitre 4 notre environnement collaboratif d'ISBM qui permet de mettre les différents acteurs interdisciplinaires en situation de co-modélisation avec leurs propres environnements autour d'un bus de co-simulation (Figure 37). Nous concluons ce chapitre par la spécification des scénarios qui nous permettent de faire la preuve de concept de notre proposition en parties 3 et 4, au facteur d'échelle considéré.

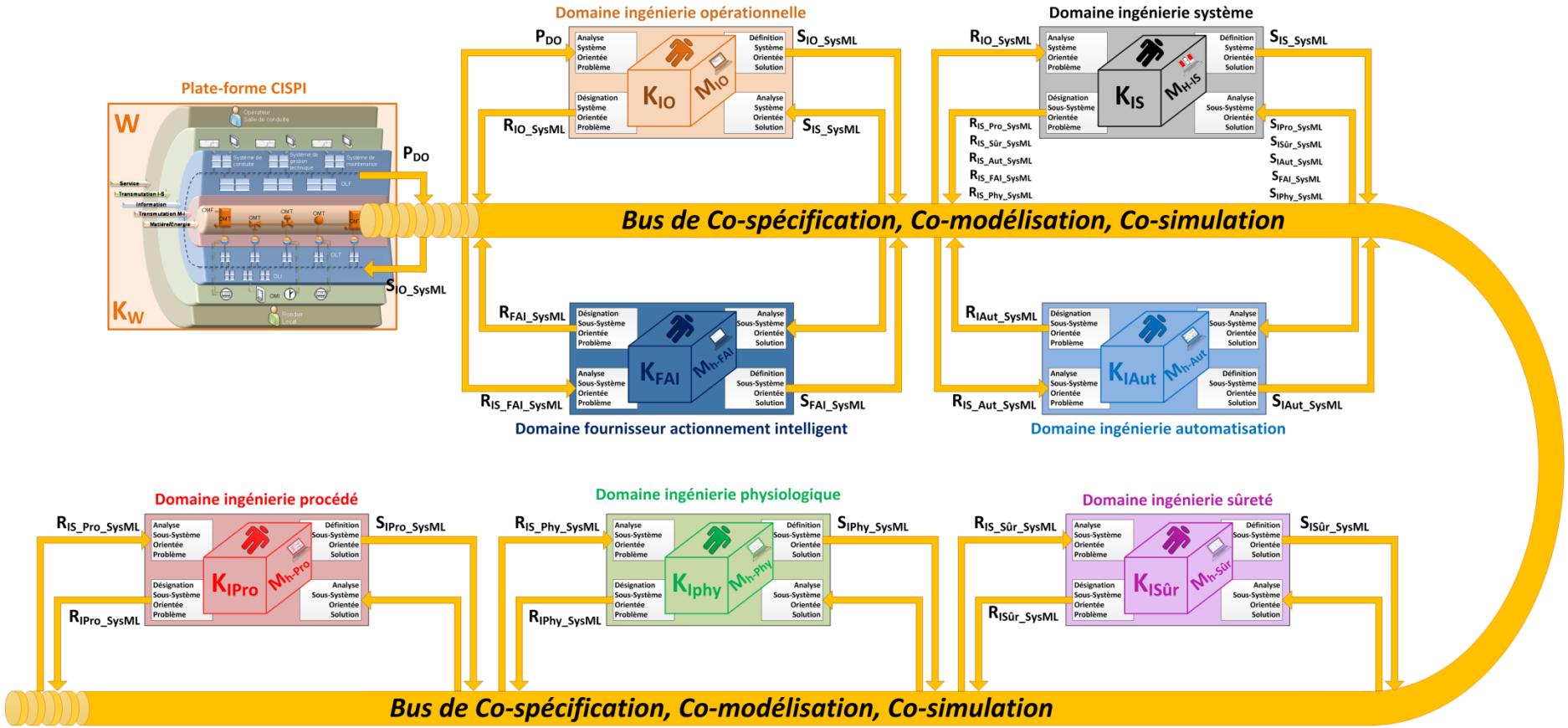


Figure 37 : Environnement de co-spécification exécutable basée sur des modèles

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

3. Chapitre 3 : Heuristique de spécification-système pour faire CISPI

3.1. Introduction : Caractérisation du processus de spécification d'un modèle total de CISPI

Nous fondons dans ce chapitre notre approche heuristique de co-spécification holistique d'un SYSTEME sur un ensemble de travaux formalisant les artefacts essentiels de spécification d'un SYSTEME d'intérêt.

Cette formulation sous-tend une approche systémique qui dépasse la vision classique d'un SYSTEME.

> ***Systems Thinking*** *is a way of thinking that, in contradistinction to traditional analysis, does not break down entities in separate parts, but is concerned with how the parts are interacting such that they reproduce or transform a whole. Thus, it attempts to understand the parts in relation to the whole* (HB Lawson, et al., 2005-2014)

Nous présentons, dans un premier temps, un ensemble de paradigmes qui nous permettent de fonder les artefacts systémiques essentiels de notre processus de co-spécification SYSTEME.

Cependant, encore faut-il savoir orchestrer, au sens de l'ingénieur-système, un ensemble de connaissances interdisciplinaires pour former une vision holistique partagée, un modèle, du SYSTEME à faire.

> ***Systemic Thinking*** *a way of thinking that emphasizes connectedness and enables people to see the bigger picture; one in which owners, solvers, solutions, problem solving methods, and problem descriptions are portrayed as a whole system* (Boardman & Sauser, 2013b)

Nous développons dans un deuxième temps les artefacts essentiels du raisonnement systémique qui nous permet de faire interopérer de façon heuristique l'ensemble des modèles co-spécifiant un modèle du SYSTEME à faire.

3.2. Paradigmes de spécification système

La notion de spécification peut se référer à plusieurs concepts en ingénierie système. Comme le souligne (Van Lamsweerde, 2000), une source de confusion vient du fait que le mot "*spécification*" soit utilisé à la fois pour se référer au processus de spécification, ainsi qu'au résultat de ce processus. Si dans la littérature, la spécification vue comme un résultat a été définie dans de nombreux travaux, comme nous pouvons le constater dans le glossaire en ingénierie système[26] de l'INCOSE, ceci n'est pas le cas du processus de spécification sur lequel nous nous focalisons dans la suite de ce mémoire. Parmi les définitions existantes, nous avons retenu la définition suivante de l'AFIS (Association Française d'Ingénierie Système) pour la spécification (en tant que résultat) :

«*Spécification : Document prescrivant des exigences (ISO 8402). Généralement utilisé lorsque le document est validé et approuvé par les deux parties, client et fournisseur…. La spécification est donc un modèle prescriptif du système (elle prescrit le « quoi » ce que l'on attend du système* » (Fiorèse & Meinadier, 2012).

[26] http://www.incose.org/productspubs/pdf/techdata/ertc/glossarydefnsofterms_1998-10_twg.pdf

3.2.1. Paradigme de spécification orienté automatisation

A l'initiative de la DAST de Renault (Direction de la Recherche des Automatismes), les travaux menés dans le cadre du PTA (Poste de travail de l'Automatisation) (PTA, 1987) visant à assister l'automaticien tout au long du cycle de vie d'une application, ont conduit à définir un corpus scientifique cohérent de concepts, théories, modèles, méthodes, méthodologies, langages et outils en génie automatique (Frachet, 1987) (Lhoste, 1994). Notre équipe actuelle de recherche a contribué dès l'origine à ces travaux et s'est intéressée plus formellement aux fondements du paradigme d'automatisation sur la base des travaux de (Fusaoka, Seki, & Takahashi, 1983). Ce dernier postule qu'automatiser un SYSTEME consiste classiquement à définir des règles de contrôle et de commande (***Unknown Control Rules***) à partir de la connaissance de la dynamique du système physique à automatiser (***Dynamics***) et des objectifs à atteindre globalement pour le système automatisé (***Behaviour Goal***) selon :

$$\textit{Unknown Control Rules} \wedge \textit{Dynamics} \supset \textit{Behaviour Goal} \qquad (1)$$

Par la suite, (Lamboley, 2001) substitue un processus de raffinement à cette approche courante en automatique (Christos Cassandras & Lafortune, 1999) qui réduit la notion de « SYSTEME » à la seule partie à contrôler en se focalisant sur la seule élaboration d'un modèle comportemental de la commande comme souligné récemment par (C Cassandras & Braatz, 2011). En ce sens, Lamboley revisite la formulation du paradigme d'automatisation (1) et l'interprète de manière plus générale selon (2) afin de mieux rendre compte de la nature logique de la relation de spécification (opérateur $\rightarrow$ se substituant à $\supset$ pour traduire aussi bien la nature moins formelle du processus, que formelle du résultat) :

$$\underbrace{\textit{Processus de commande}, \textit{Processus opérants}}_{\substack{\textit{Système concret} \\ (\textit{Lois physiques},\ \textit{comportements})}} \rightarrow \underbrace{\textit{« SYSTEME »}}_{\substack{\textit{Système abstrait} \\ (\textit{missions},\ \textit{objectifs},\ \textit{propriétés})}} \qquad (2)$$

De plus, Lamboley insiste sur la difficulté « *d'assurer à la fois la correction (correctness) intrinsèque à chacun des modèles (*du contrôle et du procédé*) vis-à-vis de leurs règles de construction et de l'application ainsi que leur cohérence globale (consistency) vis-à-vis du comportement attendu du SYSTEME* ». En ce sens, Lamboley propose de différer l'utilisation de représentations orientées métiers dans la phase de spécification par une représentation commune entre toutes les parties prenantes impliquées durant la conception, selon le prédicat :

$$\textit{Spécification des processus de commande},\ \textit{Spécification des processus opérants} \rightarrow \textit{Spécification "SYSTEME"} \qquad (3)$$

Notons que la proposition de (Lamboley, 2001), basée sur la méthode B (Abrial & Hoare, 2005) pour le raffinement formel de modèles, ainsi que pour la vérification de leur cohérence globale vis-à-vis des objectifs, limite le comportement dynamique de processus continus d'une application de type CISPI à une représentation à évènements discrets. La proposition de notre environnement de co-spécification exécutable dans le chapitre 4 vise au contraire à s'affranchir de l'utilisation d'un langage unique pour la spécification holistique du SYSTEME afin de s'appuyer au mieux sur chaque domaine d'ingénierie avec son propre environnement de modélisation.

Les travaux de (Pétin, Morel, & Panetto, 2006) complètent ces travaux pour les étendre à l'ingénierie des systèmes automatisés. Soit ($\boldsymbol{P_C}$) spécification de la partie commande et ($\boldsymbol{P_O}$) spécification du procédé et de la partie opérative, ils interprètent la relation ($\boldsymbol{P_C}$, $\boldsymbol{P_O}$) comme une architecture d'un SYSTEME automatisé

satisfaisant aux exigences systèmes (***S***), et résultant d'un processus de spécification concourant entre domaines requis d'ingénierie selon (4) (Figure 38). Notons que ces exigences systèmes (***S***) dépendent d'exigences parties prenantes R.

$$P_c, P_O \rightarrow S \tag{4}$$

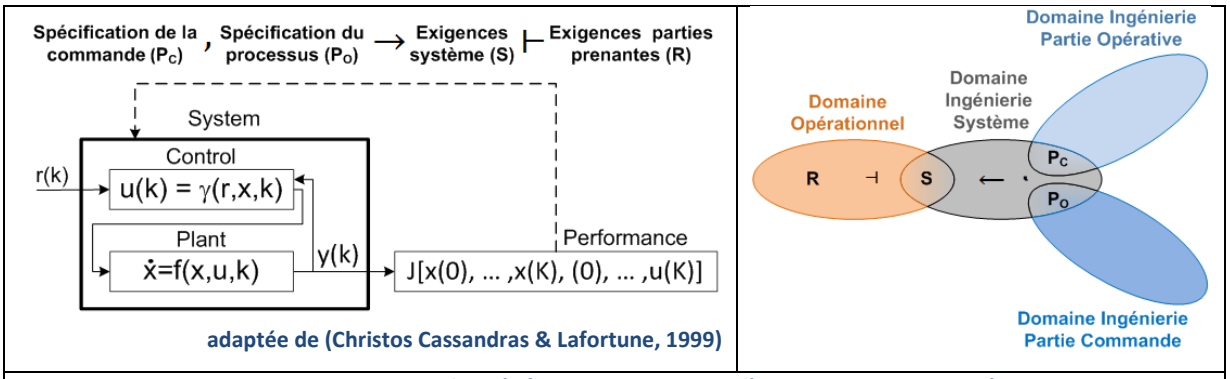


Figure 38 : Processus de spécification concourante d'un SYSTEME automatisé

3.2.2. Paradigme de spécification orienté problème

3.2.2.1. Systèmes techniques

Les travaux de (Dobre, 2010) sur la formalisation du processus de spécification en ingénierie système, ont conduit à réinterpréter le paradigme d'automatisation dans le cadre des « problems frames » de (Jackson, 2001). Parmi les nombreux travaux en génie informatique contingent à cette approche, Dobre revisite ceux de Jackson (Jackson & Zave, 1995) (Jackson, 1997) pour démontrer que c'est la mise en œuvre réelle dans l'environnement (***W***) du SYSTEME réalisé et définit par une architecture (***P_C, P_O***) à partir d'une spécification (***S***) qui garantit la satisfaction des exigences (***R***). Ainsi, d'un point de vue automatisation, il ne s'agit plus de garantir le fonctionnement du SYSTEME automatisé en se focalisant uniquement sur la dynamique et le comportement des sous-systèmes de commande mais en considérant également leurs aspects fonctionnels, organisationnels et informationnels en lien avec l'environnement dans lequel ils évoluent (G. Morel, Panetto, Zaremba, & Mayer, 2003)

Plus précisément, ces différents artefacts font partie d'un modèle de référence proposé par (Gunter, et al., 2000) revisité par Dobre et répartis entre le domaine de l'environnement et celui du SYSTEME (Figure 39) :

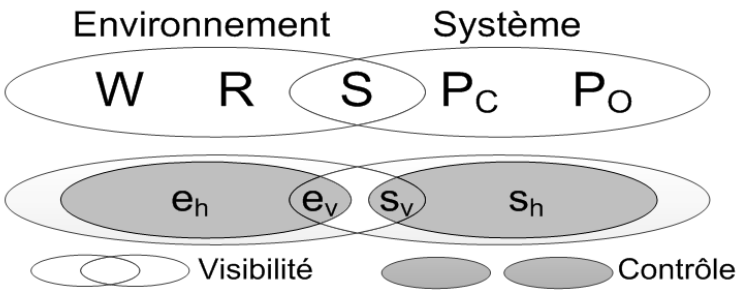


Figure 39 : Modèle de référence pour les exigences et les spécifications adaptée de (Gunter, et al., 2000)

- La connaissance du domaine (***W**orld*) fournit une description indicative de l'environnement, spécifiant les conditions connues sur les phénomènes de l'environnement validés par des spécialistes et ceci quel que soit le comportement et les propriétés du SYSTEME réalisé.
- Les besoins (***R**equirement*) fournissent une description optative de ce qui est requis par l'environnement. Ils spécifient les phénomènes que l'on souhaite voir se manifester une fois le SYSTEME mis en œuvre dans cet environnement.
- La partie commande (P_C) (*Analogie avec le **P**rogramme en informatique*) reçoit les consignes de l'opérateur et comptes rendus de la partie opérative. Elle adresse des ordres à la partie opérative et des signaux à l'opérateur. Son programme gère l'ensemble de ces échanges d'informations.
- La partie opérative (P_O) (*Analogie avec la plate-forme de programmation (**M**achine) en informatique*) reçoit les ordres de la partie commande, effectue les opérations et adresse des comptes rendus à la partie commande.
- La spécification (***S***) fournit des informations suffisantes pour que des ingénieurs puissent concevoir le SYSTEME qui satisfait les exigences. La spécification est exprimée en termes de phénomènes visibles à la fois par le domaine de l'environnement (e_v) et du système (s_v). En effet, la séparation du domaine de l'environnement et du système permet la séparation des phénomènes (*Etats, Evènements, Personnes*) appartenant et contrôlé par l'environnement $\{e_h, e_v\}$ ou par le système $\{s_h, s_v\}$. Ainsi, l'ensemble des phénomènes $\{e_h, e_v, s_v\}$ visibles par l'environnement sont utilisés pour décrire (***W***) et (***R***). L'ensemble des phénomènes $\{e_v, s_h, s_v\}$ visibles par le système sont utilisés pour décrire (P_C) et (P_O).

Ainsi, de façon plus formelle, nous avons la spécification (***S***) selon :

$$\forall e, s. W, S \vdash R \tag{5}$$

Permet de prescrire une solution (P_C, P_O) selon :

$$\forall e. (\exists s. S) \vdash (\exists s. P_c, P_o), (\forall s. P_c, P_o \vdash S) \tag{6}$$

En réponse au besoin (***R***) selon :

$$\forall e, s. W, P_c, P_o \vdash R \tag{7}$$

Les travaux originaux en génie informatique sont une contribution essentielle à la construction d'un schéma mental de spécification d'exigences entre un domaine d'intérêt et un domaine d'ingénierie. La formulation (7) rend compte d'une nécessaire concourance pour satisfaire les exigences (***R***) d'un système technique automatisé entre un domaine d'intérêt requérant (***W***) et deux domaines d'ingénierie spécialiste relatifs à la fois à la partie commande (P_C) (« Software in the loop ») et à la partie opérative (P_O) (« Hardware in the loop »).

3.2.2.1.1. Relation de description-prescription entre espaces problème-solution

(Dobre, 2010) interprète ainsi le modèle de référence de (Gunter, et al., 2000) comme : « *une partition entre domaines où l'environnement représente le domaine d'exploitation visé et où le système représente le domaine d'ingénierie système dans un premier temps et, dans un deuxième temps, comme une partition entre le domaine d'ingénierie système et les domaines métiers des composants du système. Ceci de façon récursive jusqu'à la spécification des constituants du système* » (Figure 40).

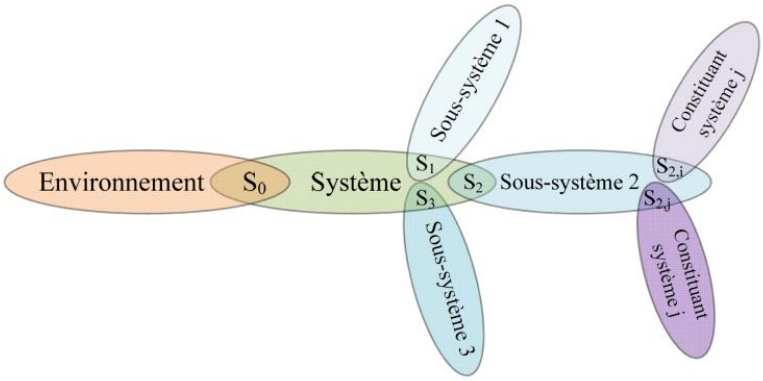


Figure 40 : Partition des domaines autour du domaine du SYSTEME à faire (Dobre, 2010)

Dans le contexte de l'ingénierie système, Dobre considère que chacun des domaines comporte deux rôles en tant qu' **espace-problème** et **espace-solution** comme suggéré dans (Czarnecki, 1998), cela de manière récursive jusqu'aux domaines de réalisation des constituants vus comme des **espaces-solution**. En ce sens, le domaine de l'environnement du modèle de référence est vu comme un **espace-problème** et le domaine du système est vu comme un **espace-solution**.

Cette séparation de l'**espace-problème** et de l'**espace-solution** est un artefact essentiel en ingénierie système. Cet artefact permet d'éviter toute confusion entre les parties prenantes responsables du besoin (maitre d'ouvrage) et de la solution (maitre d'œuvre) comme mis en évidence par (Fiorèse & Meinadier, 2012) : « *toute confusion entre ces deux rôles est préjudiciable tant pour le système à faire (risque de besoin défini par la solution et non l'inverse par exemple) que pour le projet pour le faire (risque de modification incessante de la définition du besoin entrainant de multiples reprises de conception de la solution)* »

Le processus de spécification est alors vu comme une relation logique de **description-prescription** entre un **espace-problème** et un **espace-solution**. La spécification en tant que résultat de ce processus est une solution vérifiée et prescrite par **l'espace-solution** et validée par l'espace-problème en réponse à un problème décrit par ce même espace. Notons que cette relation de **description-prescription** telle que définie en ingénierie système diffère de l'interprétation faite dans le contexte des « problems frames » comme spécifié par (Jackson, 1995). La description se réfère à la spécification de connaissances dans le mode indicatif, statuant sur des faits objectifs, tandis que la prescription se réfère à la spécification d'exigences dans le mode optatif et exprimant un souhait ou un désir (Bjørner, 2009). Dans ce mémoire, nous interprétons la **description** comme un échange de problèmes depuis un **espace-problème** vers un **espace-solution**. La **prescription** est vue comme la relation inverse de la **description** et correspond à un échange d'une solution depuis un **espace-solution** vers un **espace-problème**.

3.2.2.1.2. Récursivité entre domaines orientés problème-solution

En fonction du système étudié, l'**espace-problème** et l'**espace-solution** peuvent appartenir au même **domaine** si le problème peut être résolu dans ce même **domaine**, ou ils peuvent appartenir à différents **domaines** si d'autres domaines d'expertises sont requis pour prescrire une solution. Comme nous le voyons à travers les prédicats (5) (6) et (7), la représentation commune (***S***) partagée entre les domaines spécialistes du procédé et du contrôle est d'importance pour réaliser un SYSTEME automatisé (P_C, P_O) satisfaisant les exigences (***R***). Cela nous a conduit dans (Bouffaron, Gouyon, Dobre, & Morel, 2012) à prendre en compte le

processus interdisciplinaire requis pour concevoir le SYSTEME comme un TOUT, à travers sa spécification (***S***). En ce sens, le domaine de l'ingénierie système apparait comme le domaine pivot entre les ingénieries spécialistes impliquées dans la conception du SYSTEME. Ainsi le domaine d'ingénierie système est en charge d'architecturer les composants systèmes (P_C) et (P_O) pour satisfaire la spécification (***S***) du SYSTEME comme un TOUT (Figure 41). La relation de description-prescription est ainsi vue comme une relation récursive entre domaines d'ingénieries (Herzog, 2004). Notons que nous interprétons le processus de spécification :

- Entre un **espace-solution** et un **espace-problème** comme le processus de définition du besoin (ISO/IEC 15288, 2008) avec comme entrée des besoins non formalisés et en sortie des exigences parties prenantes (***R***).
- Entre un **espace-problème** et un **espace-solution** comme le processus d'analyse des exigences (ISO/IEC 15288, 2008) avec comme entrée des exigences parties-prenantes et en sortie des exigences SYSTEMES (***S***).

Le processus de spécification entre plusieurs domaines est alors qualifié de **processus de co-spécification** car il met en jeu plusieurs domaines : opérationnel et d'ingénieries système et spécialistes pour spécifier de façon collaborative le SYSTEME comme un TOUT.

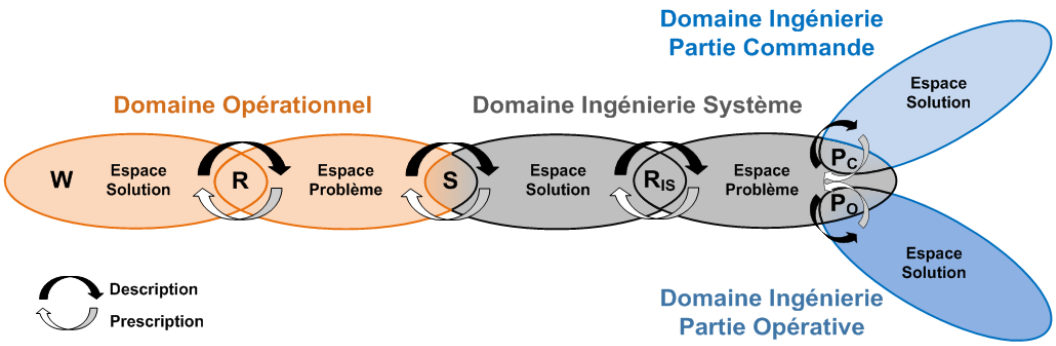


Figure 41 : Processus itératif, récursif et concourant de co-spécification en ingénierie de SYSTEME automatisé

Notons que cette organisation « linéaire » où le domaine d'ingénierie système occupe la position de domaine pivot entre les ingénieries spécialistes présente l'inconvénient de filtrer le domaine opérationnel en regard domaines spécialistes qui ne voient celui-ci qu'à travers le domaine de l'ingénieur système. Elle s'avère cependant très pragmatique en pédagogie pour mettre apprenant et tuteur en situation respective d'espace-solution et d'espace-problème pour un apprentissage itératif à l'ingénierie système (G. Morel, et al., 2014).

3.2.2.1.3. Itération entre espaces problème-solution

Si nous nous référons à la Figure 39, la **description** est exprimée en termes de phénomènes contrôlés par l'**espace-problème**, et la **prescription** en termes de phénomènes visible à la fois par l'**espace-problème** et l'**espace-solution**. Etant donné que cet ensemble de phénomènes sont disjoints, et comme mis en évidence par (Hall, Jackson, Laney, Nuseibeh, & Rapanotti, 2002) la solution prescrite peut être difficilement validée après une seule itération de **description-prescription** entre les deux espaces. En ce sens, ils étendent l'approche des problems frames vers le « Twin Peaks model » pour représenter la relation itérative de spécification entre un **espace problème** et un **espace solution (**Figure 42**)**.

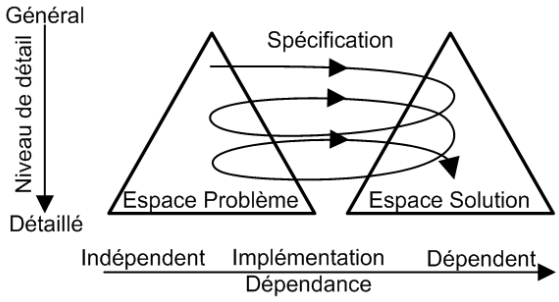


Figure 42 : Twin Peaks model – itération entre les espaces problème et solution. Adaptée de (Hall, et al., 2002)

Durant ce processus itératif de **description-prescription**, les espaces du problème et de la solution s'enrichissent et se détaillent de façon concourante jusqu'à ce qu'ils convergent vers une spécification exprimée entièrement en terme de phénomènes visibles pour les deux espaces.

3.2.2.2. Systèmes sociotechniques

Récemment (Lieber, 2013) a revisité les travaux de (Dobre, 2010) en interprétant le cadre de modélisation de SYSTEMES sociotechniques proposé par (Hall & Rapanotti, 2005) étendant celui des « Problem Frames » en génie informatique (Gunter, et al., 2000) (Figure 43).

(Hall & Rapanotti, 2005) considère que le comportement des utilisateurs du SYSTEME doit être étudié dès l'analyse des exigences. En d'autres termes la machine (***M***) aussi bien que l'humain (***H***) doivent faire l'objet de descriptions optatives (exigences). Par comparaison avec le domaine de la machine, qui est un domaine causal (prédictif) Hall et Rapanotti qualifient le domaine de l'humain de domaine « docile » (*biddable*) pour exprimer le manque de prédiction sur le comportement de l'humain. En ce sens les exigences imposées à l'humain se limitent à des contraintes sur les actions permises entre l'utilisateur et la machine. Nous étendons dans le chapitre 8 de notre mémoire, ce champ de définition des exigences, à des exigences mesurables prescrivant les propriétés requises de la machine lors d'une interaction Homme-Machine.

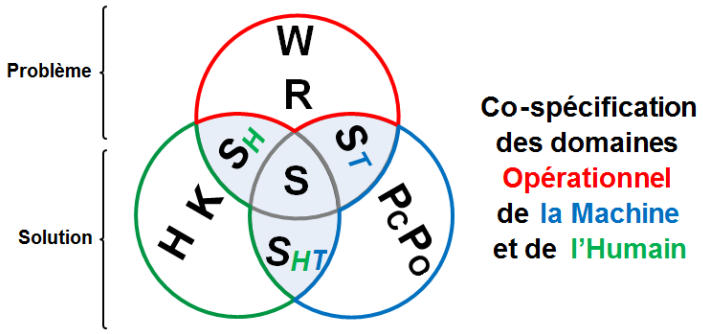


Figure 43 : Modèle de référence d'un SYSTEME sociotechnique. Adaptée de (Hall & Rapanotti, 2005)

Ce modèle de référence d'un SYSTEME sociotechnique (Figure 43) fait apparaitre trois nouvelles spécifications avec l'introduction de l'humain (***H***) et de sa connaissance (***K***) :

- La spécification (S_H) qui spécifie les connaissances et le comportement qui est attendu de l'humain comme un composant du SYSTEME sociotechnique

- La spécification (***S_{HT}***) qui spécifie l'interface Homme-Machine
- La spécification (***S_T***) qui spécifie la partie commande et la partie opérative qui est attendue de la machine comme un composant du SYSTEME sociotechnique. La spécification (***S_T***) est équivalente à la spécification (***S***) dans le modèle de référence de (Gunter, et al., 2000).

Sur la base des travaux de (Lieber, 2013) auxquels nous avons contribué (OMG), nous réinterprétons le prédicat (5) à satisfaire et en proposons une écriture simplifiée :

$$W, S_H, S_T, S_{HT} \vdash R \tag{8}$$

De la même manière que nous avons interprété le prédicat (7), le prédicat (8) rend compte d'une nécessaire concourance pour satisfaire les exigences (***R***) d'un SYSTEME socio-technique entre un domaine d'intérêt requérant (***W***) et deux domaines d'ingénierie spécialiste relatifs à la fois à la partie Technique (***S_T***) (« Hardware-Software in the loop ») mais également à la partie humaine (***S_H***) (« Human in the Loop ») en interaction avec la partie technique (***S_{HT}***) (« Human-Machine in the loop »). Ainsi, ce prédicat (8) met en évidence que trois types de spécifications (notées ***S_H, S_{HT}, S_T***) concourent de façon itérative à la spécification du SYSTEME (***S***) prescrivant le système Homme-Machine comme un TOUT pour satisfaire l'ensemble des exigences (***R***) selon :

$$S_H, S_T, S_{HT} \rightarrow S \tag{9}$$

Dans ce contexte, (Lieber, 2013) propose une nouvelle organisation en ingénierie système et revisite par la même occasion le processus de co-spécification présenté précédemment, en introduisant deux nouveaux architectes (Architecte-Système Humain et Architecte-Système Technique) ainsi que leur domaine d'ingénierie associée. Ces architectes interagissent à la fois avec l'architecte système (responsable du système comme un TOUT) et leurs ingénieries spécialistes respectives, tout au long du cycle de vie du SYSTEME (Figure 44).

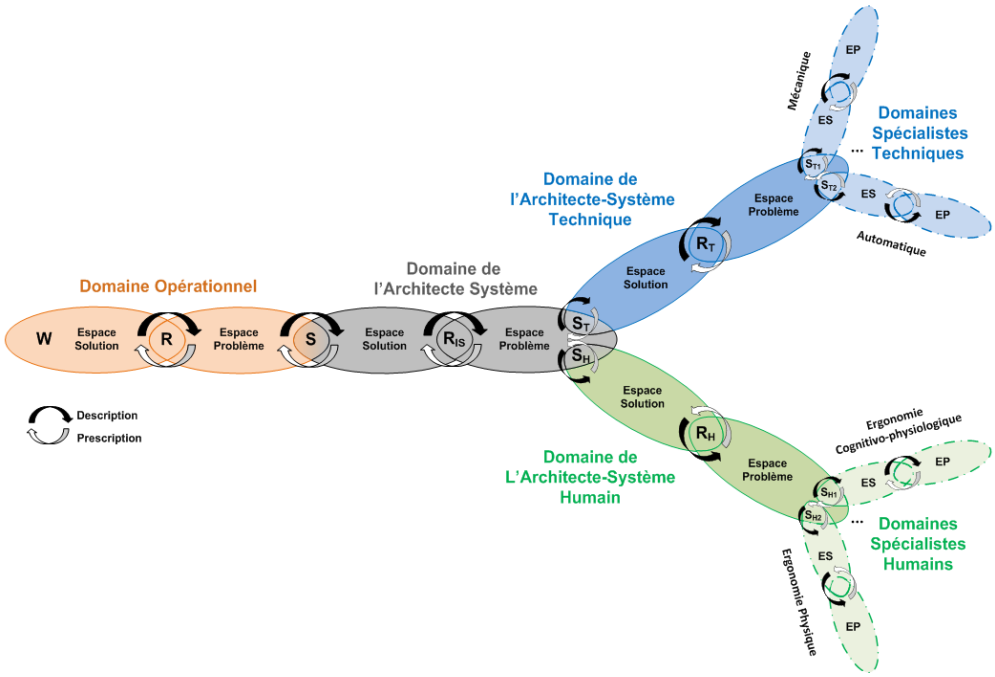


Figure 44 : Proposition d'une organisation d'ingénierie d'un SYSTEME$_{\text{Homme-Machine}}$ au travers du processus de co-spécification SYSTEME. Adaptée de (Lieber, 2013)

3.3. Paradigmes de modélisation système

Alors que les travaux présentés précédemment se sont concentrés sur la **spécification-SYSTEME** en tant que résultat d'un processus, nous nous sommes focalisés dans la suite de ce chapitre sur la connaissance systémique à mettre en œuvre pour **construire-SYSTEME** selon une vision moléculaire acquise au cours de notre formation et de nos travaux de doctorat. (Figure 45 et Figure 46).

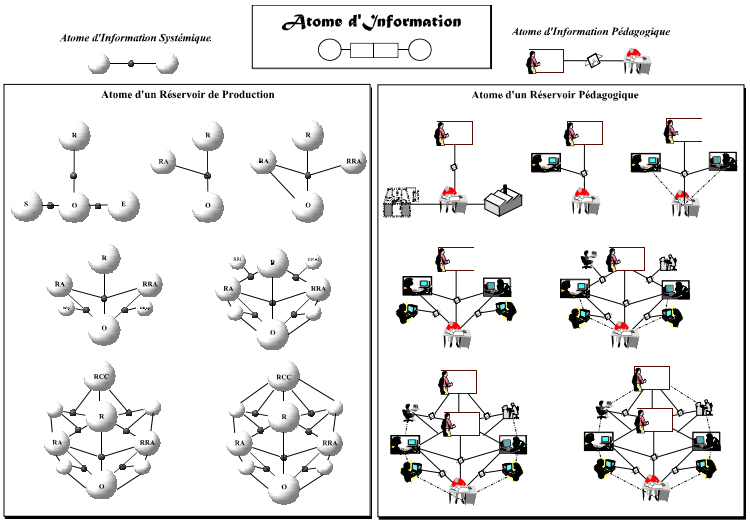


Figure 45 : Vision moléculaire d'une construction SYSTEME (Mayer, 1995)

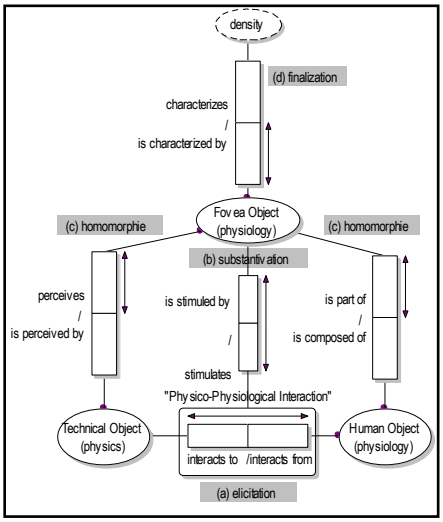


Figure 46 : Patron essentiel en formalisme FOSM (Fact Oriented System Modeling)[27] (OMG) (G. Morel, et al., 2013)

3.3.1. Fondements holoniques

Dans son approche des « problem frames », (Jin, 2006) suggère de prendre en compte certaines propriétés systémiques essentielles que sont :

- la **globalité** (*en. Wholeness*) et l'**ouverture** (*en. Openness*) : Un SYSTEME est une partie de la réalité. Il est séparé de son **environnement** par une **frontière** et peut être observé en **interaction** avec son environnement.
- la **structure** et l'**émergence** (*en. Emergency*) : De manière générale, un SYSTEME est composé de plusieurs **sous-systèmes** en interaction, qui présente des propriétés globales (effets des comportements d'ensemble) non réductibles aux propriétés de ses sous-systèmes ses constituants mais émergeant du réseau d'interactions entre ses constituants. Ces propriétés sont souvent dites propriétés émergentes (Fiorèse & Meinadier, 2012).

Ce faisant, elle fonde principalement son approche holistique d'une spécification d'exigences sur le concept d'holon (Koestler, 1967) afin de percevoir la situation d'intérêt (K_{IO}) comme un système-holon (*que nous noterons s-holon*) à travers deux points de vue étroitement liés :

27 Fact-Oriented System Modeling : cette variante orientée-système (Hall, et al., 2002) complète « ORM » (www.ormfoundation.org*)* et son mécanisme de substantivation en préservant au mieux l'homomorphie entre l'objet émergeant d'un fait et les objets constituant ce fait d'origine. La mesurabilité de propriétés de l'objet émergeant est la condition de finalisation d'une émergence lors de la construction d'un modèle de système.

- le TOUT, à travers les manifestations externes du s-holon en interaction avec son environnement et se traduisant par des effets observables et mesurables sur l'environnement.
- la PARTIE, à travers la structure interne d'un « *super-holon* » dont est dérivée ce s-holon en tant que sous-composant en cohésion hiérarchique avec d'autres composants-holons (*sous-s-holons*) semi-autonomes et indépendants. Ces sous-holons sont eux-mêmes vus comme TOUT et PARTIE décomposables à leur tour, et à un niveau inférieur par d'autres sous-holons.

Sur la base de ce constat, les exigences relatives à un « s-holon » concerne à la fois ses manifestations externes et sa structure interne. Les manifestations externes décrivent le comportement du « s-holon » » tandis que sa structure interne décrit comment les composants « sous-s-holons » sont agencés pour implémenter ce comportement. Ces deux aspects des exigences coexistent à chaque étape du raffinement jusqu'à ce que les ingénieurs soient capables de réaliser un « sous-s-holon ». Cependant, si (Jin, 2006) propose une méthodologie pour percevoir l'environnement ainsi que ses interactions avec le SYSTEME d'intérêt et les manifestations attendues, en phase de spécification en vision externe, l'auteur n'explique pas comment, durant la phase de spécification, en vision interne, elle découvre la structure du SYSTEME. Cela peut s'expliquer par le fait que les systèmes abordés sont des systèmes informatiques dont l'auteur possède déjà la connaissance à priori pour raffiner la solution, ce qui n'est pas le cas en ingénierie système avec des systèmes complexes (interdisciplinaires) sociotechniques, mécatroniques ….

Beaucoup d'autres travaux ont montré la pertinence d'une perception holonique d'une situation d'intérêt lors d'une construction-SYSTEME, notamment certains de notre équipe pour spécifier des « Holonic Manufacturing Execution Systems » (G. Morel, Valckenaers, Faure, Pereira, & Diedrich, 2007) ou des « Enterprise System of Systems » (Baïna, 2006), (Auzelle, 2009) en lien avec les communautés internationales respectivement en « Intelligent Manufacturing Systems »[28] et « Enterprise Interoperability »[29].

Nous retiendrons plus particulièrement certains travaux de notre communauté d'intérêt en ingénierie système. Ceux de (Kuras, 2006), qui font de l'holon ($\boldsymbol{H^v}$) le filtre de l'ensemble $\boldsymbol{\{E_i\,;\,REL_j\}}$ des conceptualisations d'un SYSTEME d'intérêt (***S***) perçu à l'échelle (***v***) pour faciliter, échelle par échelle, sa représentation multi-modèles multi-échelles comme archétype TOUT et PARTIE d'un système selon :

$$s^v \in \{S^v\} \mid S^v \equiv \{E_i; REL_j\}_{I^v J^v}; H^v \tag{10}$$

avec (***E***) élément de (***S***) et (***REL***) les relations entre ses éléments.

Comme rappelé dans la thèse de (Lo, 2013) un SYSTEME (***SYSTEM***) est défini comme « *une abstraction* (***CONCEPTUALIZATION***) *d'un phénomène ou d'une situation perçue comme complexe* (***REALITY***) d*ans le but de le comprendre et de le maitriser »* (Penalva, 1997) (Le Moigne, 1990). Cette abstraction passe par une représentation sous la forme d'un ou plusieurs modèles résultant d'une activité de modélisation de ce phénomène ou de cette situation, moyennant l'absence volontaire de certains détails considérés comme non pertinents au regard des objectifs de compréhension et de maitrise (définis par un niveau d'échelle : ***SCALE***). Dans le contexte de la co-spécification holistique d'un SYSTEME, nous considérons que le niveau d'échelle requis pour modéliser le SYSTEME comme un TOUT (***WHOLE***) est le niveau fonctionnel selon le paradigme systémique. L'ingénieur système (***MODELER***) s'appuie alors sur un ensemble de patrons (***SET OF PATTERNS***)

[28] http://www.ims.org/
[29] http://www.interop-vlab.eu/

pour conduire cette définition multi-échelle d'un SYSTEME. Le systémigramme de la Figure 47 représente quelques relations essentielles entre concepts de la définition multi-échelle d'un SYSTEME de (Kuras, 2006) qui ont fait aussi l'objet d'application par (Bjelkemyr & Lindberg, 2007) pour définir certaines caractéristiques de systèmes de systèmes de production futures.

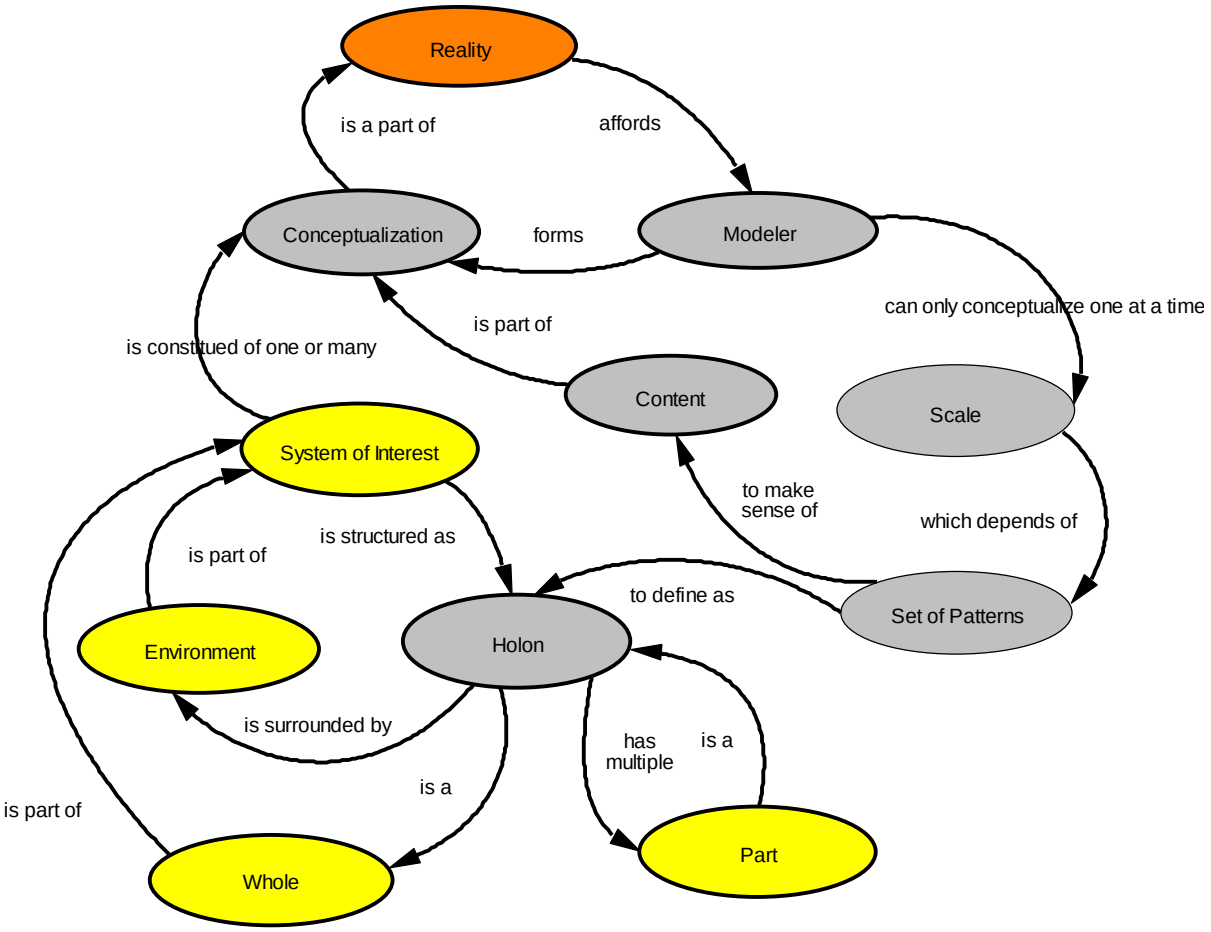
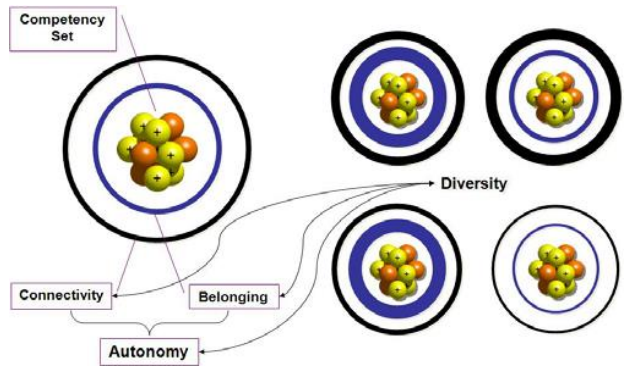


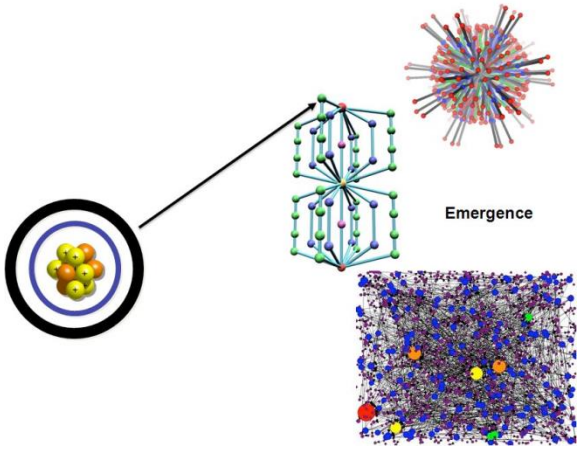
Figure 47 : Systémigramme de définition multi-échelle d'un SYSTEME d'intérêt

Dans ses travaux de recherche en « biology of systems » quant à ce qui constitue l'« ADN » d'un SYSTEME (Boardman & Sauser, 2006), l'équipe de recherche Systomics considère l'holon comme essentiel pour définir le degré d'autonomie de chacun des « Building Block » d'un système de systèmes en regard de la propriété d'**unité** (*en. togetherness*) requise par l'ensemble (Figure 48).

> *« We believe that the essence of a system is « togetherness », the drawing together of various parts and the relationships they form in order to produce a new whole.* » (Boardman & Sauser, 2008)

Un holon est une entité qui est elle-même un TOUT composée de PARTIES, mais en même temps une PARTIE d'un ensemble plus vaste. Le paradigme holonique peut être utilisé de manière récursive ; ainsi chaque PARTIE d'un SYSTEME peut être considéré à la fois comme une entité vue de l'extérieur, et comme un ensemble de PARTIES en interactions vues de l'intérieur. Ce modèle s'applique également de sorte que le SYSTEME d'intérêt original peut être perçu comme une PARTIE en interaction d'un ou plusieurs systèmes plus larges.





Le noyau de l'holon représente l'ensemble de ses compétences correspondant à son autonomie (sa capacité à faire ce qu'il est censé faire, sans surveillance continue)	*La couche externe de l'holon représente l'ensemble de ses compétences d'appartenance, de connectivité mais aussi de diversité par rapport à d'autres holons*

Figure 48 : Le système vu comme un holon est composé d'un noyau interne et d'une couche externe en « communication » permanente (Boardman, Sauser, & Verma, 2009)

3.3.2. Fondements systémiques

Dans son approche systémique des « problem frames », (Jin, 2006) suggère qu'une spécification (***S***) est dérivable à partir d'un ensemble de **connaissances** et de contraintes (***K***) sur l'environnement du SYSTEME d'intérêt et de sa structure interne pour satisfaire aux exigences parties prenantes (***R***) selon :

$$K, R \rightarrow S \tag{11}$$

Aucune précision n'est cependant clairement fournie pour fonder les éléments de cette connaissance systémique, si ce n'est quelques principes articulés autour du concept d'holon.

Les travaux de (Kuras, 2006) approfondissent ce que sont les éléments de définition d'un SYSTEME d'intérêt, mais à l'instar des préceptes de l'approche systémique en générale, la logique d'articulation des différents éléments de définition pour construire-SYSTEME n'est pas explicite.

Ce questionnement relatif à « *'how-to' guide for systemigrams* » (Boardman & Sauser, 2013a) a conduit l'équipe Systomics à proposer un cadre de modélisation SYSTEME reposant sur un ensemble de vingt et un concept: le « conceptagon » (Figure 49). Non seulement ce cadre de modélisation à travers une représentation tangible, guide l'ingénieur système à penser système, mais aussi il représente un pont avec les autres domaines d'ingénieries spécialistes étant donné que les concepts manipulés trouvent leurs pertinences dans chacun de ces domaines pouvant les interpréter à travers leurs champs de compétences.

Plus précisément, le « conceptagon » guide la perception du SYSTEME d'intérêt par le domaine d'ingénierie système à travers sept triades de concepts. Ces sept triades représentent chacune des points de vue différents et nous assure une complétude systémique dans la spécification d'un SYSTEME d'intérêt. Parmi ces triades on

retrouve notamment la triade (« ***Communication, Command, Control »***) qui permet d'étudier le SYSTEME-holon à travers son rôle de **contrôle** (« feedback ») de sa part intérieure pour maintenir sa finalité propre, mais aussi son rôle de **commande** en regard des services requis par sa part extérieure, avec la **communication** pour assurer la relation d'interopérabilité entre ces parties d'un TOUT.

Notons que les auteurs ne préconisent pas de méthodes normatives pour utiliser le « conceptagon », chaque ingénieur est libre d'étudier un SYSTEME en commençant par n'importe quelle triade.

3.4. Processus de modélisation holonique

Nous retenons le « conceptagon » comme patron de pensée systémique pour notre heuristique de co-spécification SYSTEME que nous adaptons pour prendre en compte la dimension holistique du SYSTEME d'intérêt (« ***Relationships, Wholes, Parts*** ») selon le paradigme holonique (Figure 49). Chacune des triades du « conceptagon » interconnectés les unes avec les autres, guide l'ingénieur système dans la perception du SYSTEME d'intérêt. En ce sens, leurs études sont sources de nouvelles itérations de spécification entre le domaine d'ingénierie système et les domaines d'ingénieries spécialistes.

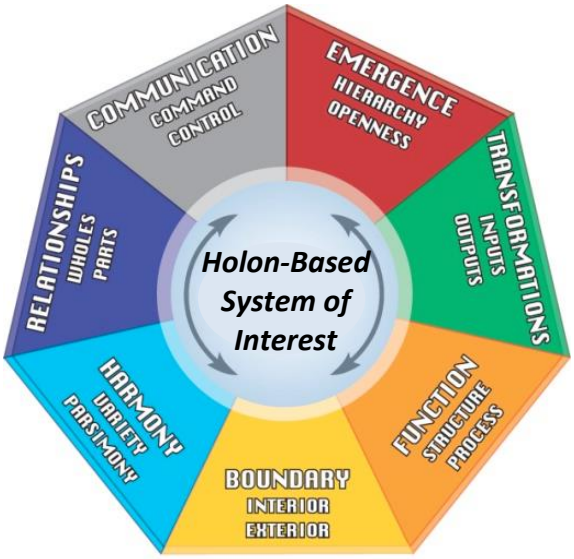


Figure 49 : Interprétation du « Conceptagon » (Boardman, Sauser, John, & Edson, 2009) selon le paradigme holonique

Pour notre part, nous recommandons de spécifier les SYSTEMES en commençant par la triade (***Transformations, Input, Output***) en étudiant les flux physiques circulant à travers ces systèmes. Ce flux physique est l'attracteur de notre raisonnement de co-spécification SYSTEME, ainsi chacune des itérations successives de spécification doivent s'assurer de maintenir les propriétés physiques de ce flux. Nous montrons dans la partie 3 de ce mémoire que l'étude de ce flux à travers la triade (***Transformations, Input, Output***), guide la perception du SYSTEME d'intérêt et fait émerger des réflexions sur les autres triades du conceptagon. De plus, afin d'éviter la proposition de solution « extravagante » nous postulons que l'étude des transformations du flux d'entrée-sortie selon les lois de la physique, permet de contenir et contraindre la modélisation fonctionnelle du SYSTEME en se rattachant ainsi à la réalité. Cela nous assure de converger vers des solutions réalisables à terme.

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

Dans la mesure où, dans nos travaux, le raffinement de l'ensemble (***R***) des exigences originelles se propage de façon itérative entre espaces Problème/Solution de divers domaines de co-spécification, il nous est apparu opportun de combiner le raffinement de cette **connaissance** systémique (***K_H***) avec le paradigme holonique dans la cadre de l'approche des « problem frames » afin de supporter un raisonnement heuristique pour construire TOUT et PARTIE d'un SYSTEME d'intérêt.

Cela nous conduit à interpréter l'approche des « problem Frames » selon un ensemble de prédicats (Figure 50). En ce sens, nous revisitons le prédicat (10) de (Jin, 2006) afin de formaliser le processus de spécification globale selon deux prédicats de spécifications d'exigences (***R***) et de solutions (***S***) selon un point de vue holonique. Notons que pour représenter le SYSTEME-holon en tant que TOUT nous le représentons par la notation « ***H*** » et en tant que PARTIE par la notation « ***h*** ».

Les quatre premiers prédicats formalisent ainsi le processus de définitions des exigences (ISO/IEC 15288, 2008) mis en œuvre dans les **espaces-problèmes**. Nous suggérons qu'une spécification d'un ensemble d'exigences (***R_S***) est dérivable d'un ensemble de problèmes (***P_S***) à partir d'un ensemble de **connaissances** et de contraintes (***K_S***) sur la situation système d'intérêt (environnement) selon :

$$K_S, P_S \rightarrow R_S \tag{12}$$

Notons que la flèche d'implication traduit également la relation de description des exigences vers un espace solution. De plus, ce prédicat (11) est le résultat d'une modélisation SYSTEME qui consiste à percevoir le SYSTEME requis (***R_S***) comme un holon en tant que :

- TOUT en regard des **connaissances** systémiques (***K_H***) requises pour le spécifier selon :

$$K_H, R_S \rightarrow R_H \tag{13}$$

- PARTIE en regard des **connaissances** et contraintes interdisciplinaires (***K_I***) requises pour l'architecturer selon :

$$K_I, R_S \rightarrow R_I \tag{14}$$

Les prédicats suivants formalisent les processus de définition d'un système mis en œuvre dans les **espaces-solutions**. La spécification SYSTEME consiste alors à dériver une solution totale (***S_H***) d'un ensemble d'exigences (***R_S***) à partir d'un ensemble de **connaissances** systémique (***K_H***) selon :

$$K_H, R_S \rightarrow S_H \tag{15}$$

Notons que la flèche d'implication traduit également la relation de prescription de la solution vers un espace problème. Cette solution (***S_H***) est ensuite validée par l'**espace-problème** en situation opérationnelle**,** à partir d'un ensemble de connaissances sur la situation système (***K_S***) (*commun à l'ensemble des ingénieries impliquées dans le projet*) afin de s'assurer qu'elle satisfait l'ensemble des exigences (***R_S***) selon le prédicat (5) que nous adaptons ici au concept holonique:

$$K_S, S_H \vdash R_S \tag{16}$$

La relation d'entailment ($\vdash$) exprime alors une relation contractuelle de spécification entre **espace-problème** et **espace-solution**.

Notons que la connaissance systémique ($\boldsymbol{K_H}$) évolue au cours du temps en fonction de la spécification système ($\boldsymbol{S_I}$) architecturant un ensemble de spécifications de faisabilités ($\boldsymbol{S_h}$) à partir des **connaissances** et contraintes interdisciplinaires ($\boldsymbol{K_I}$) pour satisfaire l'ensemble des exigences ($\boldsymbol{R_H}$) selon :

$$K_I, S_h, R_H \rightarrow S_I \tag{17}$$

Pour être intégré en tant que **connaissance** systémique ($\boldsymbol{K_H}$), la spécification de faisabilité ($\boldsymbol{S_I}$) doit être validée par l'**espace-solution** systémique afin de s'assurer qu'elle satisfait l'ensemble des exigences ($\boldsymbol{R_H}$) selon :

$$K_H, S_I \vdash R_H \tag{18}$$

La spécification de faisabilité ($\boldsymbol{S_I}$) validée étend alors les **connaissances** systémiques ($\boldsymbol{K_H}$) de **l'espace-solution** systémique requises pour spécifier la solution totale ($\boldsymbol{S_H}$)

$$K_H = \{K_{H1}, K_{H2}, \dots, K_{Hn}, S_I\} \tag{19}$$

De façon récursive, les spécifications de faisabilités ($\boldsymbol{S_h}$) sont le résultat d'un processus d'ingénierie dans des **espaces-solutions** disciplinaires consistant à dériver une solution de faisabilité ($\boldsymbol{S_h}$) d'un ensemble d'exigences ($\boldsymbol{R_I}$) à partir d'un ensemble de **connaissances** disciplinaires ($\boldsymbol{K_h}$) selon :

$$K_h, R_I \rightarrow S_h \tag{20}$$

Cette solution ($\boldsymbol{S_h}$) est considérée comme faisable, si au regard d'un ensemble restreint de **connaissances** ($\boldsymbol{K_S}$) de la situation système, elle satisfait des exigences opérationnelles ($\boldsymbol{R_s}$) selon :

$$K_s, S_h \vdash R_s \tag{21}$$

Une fois la faisabilité de la solution ($\boldsymbol{S_h}$) vérifiée, cette dernière est prescrite à l'**espace-problème** d'architecture ($\boldsymbol{K_I}$) pour valider que cette solution satisfait les exigences ($\boldsymbol{R_I}$) selon :

$$K_I, S_h \vdash R_I \tag{22}$$

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

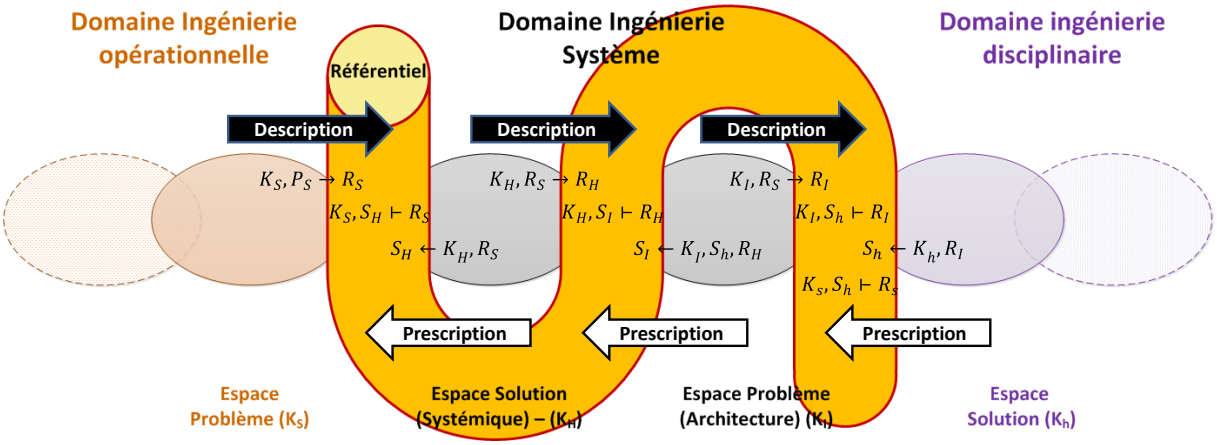


Figure 50 : Formalisation du processus de spécification

Nous étendons ce processus de spécification, à l'échelle d'une co-spécification SYSTEME multidomaines tout en conservant que ce processus itératif et récursif soit orchestré par le domaine de l'ingénierie système comme proposé par (Dobre, 2010). En revanche, nous proposons de distribuer les domaines d'ingénierie autour d'un bus de co-spécification afin de pallier aux problèmes de filtrage de la **connaissance** du domaine opérationnel (***K_{IO}***) par le domaine de l'ingénierie système, apparu dans les organisations linéaires proposés dans Dobre et Lieber (Figure 51). En effet, le prédicat de validation (5) proposé par (Jackson, 1997) requiert que chacun des domaines d'ingénieries aient accès à la **connaissance** du domaine opérationnel perçu comme l'environnement du SYSTEME, pour valider les différentes spécifications. Ainsi, le bus de co-spécification facilite les échanges d'informations, de modèles, entre les ingénieries spécialistes. De plus, cette organisation distribuée autour d'un bus de co-spécification permet à chacune des ingénieries de conserver leurs **connaissances** métier (***K***) dans leur domaine, et de partager uniquement le résultat de leur travail sur le bus.

Par rapport aux travaux de (Lieber, 2013) nous proposons d'introduire un nouveau domaine, qui est le domaine de l'ingénierie opérationnelle. Ce domaine a trouvé ses fondements à travers différentes applications aussi bien d'un point de vue technique que d'un point de vue pédagogique. Ainsi dans le cadre de notre mémoire, le domaine d'ingénierie opérationnelle représente le service plate-forme de notre laboratoire, en tant que domaine possédant les connaissances liée à l'usage, l'évaluation de la plate-forme CISPI mais également le recueil et la formalisation des problèmes liés à l'exploitation. Ainsi, il assure la vision ingénierie du système en exploitation (domaine opérationnel), vis-à-vis du domaine d'ingénierie système. D'un point de vue pédagogique, et dans le cadre de mes activités d'enseignements à l'Université de Lorraine, nous retrouvons ce domaine d'ingénierie opérationnel au niveau de l'AIPL [30] et représenté par l'équipe d'ingénierie de l'établissement. Cette équipe constitue une source de connaissances pour décliner un ensemble de scénario pédagogiques au sein du domaine opérationnel de l'AIPL qui a pour vocation de fournir des situations pédagogiques reflétant au mieux la réalité industrielle. En ce sens, le domaine d'ingénierie opérationnelle de l'AIPL effectue le lien entre le domaine opérationnel de l'AIPL et le domaine d'ingénierie système assurant l'intégration des différents domaines d'ingénierie spécialistes fournis par l'atelier (G. Morel, et al., 2014). Notons, que cette organisation se retrouve au facteur d'échelle industrielle, dans le domaine de l'aéronautique, l'automobile de la production d'énergie électrique. A titre d'exemple, au niveau d'EDF nous

30 Atelier Inter Etablissements de Productique Lorrain ; www.aip-primeca.net

interprétons la DIN (Direction Ingénierie Nucléaire) comme un domaine d'ingénierie opérationnelle. En effet, la DIN, composée de six centres d'ingénieries agit aux cotés des exploitants (domaine opérationnelle) pour recueillir et centraliser les problèmes provenant des 58 sites d'exploitation nucléaire en France, les formaliser, et tenter d'y apporter une solution en faisant appel aux compétences interdisciplinaires de ses ingénieurs.

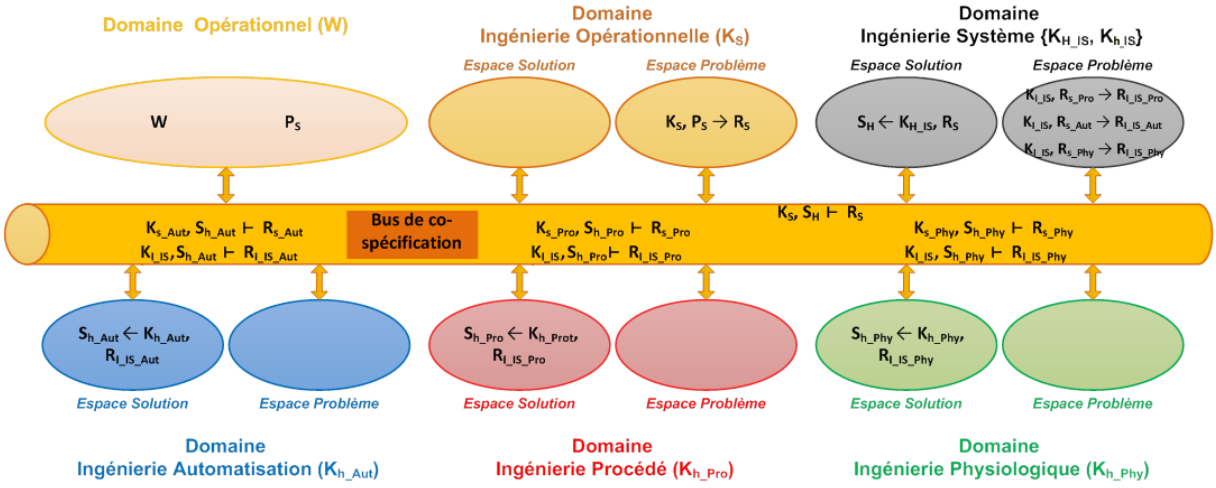


Figure 51 : Processus de co-spécification SYSTEME distribué autour d'un bus

La Figure 51 représente le processus de co-spécification mis en œuvre dans le cas d'application développé dans les parties 3 et 4. En ce sens, les domaines d'architecture-système humain et technique n'apparaissent pas, étant donné que nous avons spécifié la plate-forme CISPI avec uniquement un domaine spécialiste du technique et de l'humain représenté respectivement par le domaine d'ingénierie en automatique, et le domaine d'ingénierie en physiologie.

Plus concrètement, les itérations de spécification se propagent entre les différents domaines tant que la **connaissance** d'un **espace-solution** n'est pas suffisante pour satisfaire les exigences d'un **espace-problème** requérant. Cette dynamique de recherche de **connaissances** est à la base du processus holonique de co-spécification SYSTEME puisque le domaine de **connaissances** se construit au fur et à mesure de la perception des interactions du SYSTEME à contrôler à la fois vers son environnement et à l'intérieur de sa structure. La distribution des domaines autour du bus de co-spécification évite que cette recherche de connaissances ne soit contrainte par une organisation trop procédurale, facilitant ainsi cette dynamique par notification ou diffusion entre les domaines d'ingénieries impliqués.

- La notification (Simão & Stadzisz, 2009) apparait pertinente principalement dans la phase de concrétisation-système dans le cas où le domaine d'ingénierie qui décrit le problème cible les domaines d'ingénieries spécialistes en mesure de résoudre ce problème. Ce principe offre l'avantage de converger plus rapidement vers une solution que par le principe de diffusion. Cependant, elle restreint les possibilités d'explorer d'autres solutions pouvant répondre au problème posé.
- La diffusion apparait pertinente dans la phase d'abstraction-système afin que le domaine d'ingénierie qui décrit le problème soit en mesure d'identifier un domaine d'ingénierie spécialiste pouvant résoudre ce problème. Ce principe est particulièrement recommandé en innovation, pour qu'un maximum de domaines d'ingénieries spécialistes prescrive des variantes de solutions à un problème commun.

3.5. Conclusion : Heuristique holonique constitutive du processus de spécification-système ciblé

Comme souligné par (Cloutier, Sauser, Bone, & Taylor, 2015) , un défi fondamental en ingénierie système, qui s'avère également le nôtre dans ce mémoire, est de capturer l'essence du problème avec un modèle. Cela sous-tend de construire un modèle SYSTEME d'une situation d'intérêt (« System model in the Loop »).

Cette phase de définition du SYSTEME exige une application de la pensée système avec des outils et des méthodes de modélisation adéquates. A la différence (Cloutier, et al., 2015) qui propose de passer par une représentation du problème sous la forme d'un « systemigram[31] », puis de transférer les informations représentés par le « systemigram » en langage SysML (System Modeling Language) (OMG, 2012a) (Figure 52) ; nous proposons de construire directement le modèle du SYSTEME (**System Model in the Loop**) via SysML pour faciliter un continuum de représentation entre phases d'abstraction et de concrétisation en mettant l'ensemble des domaines d'ingénieries interdisciplinaires en situation itérative de co-spécification. En ce sens, notre heuristique de co-spécification système montre que dans une approche d'ISBM, la spécification des parties d'un SYSTEME repose sur un ensemble de **connaissances** (**Knowledge in the Loop**) provenant de différents domaines d'ingénieries spécialistes et représentés par un ensemble de modèles interdisciplinaires (**Part-Model in the Loop**), architecturés les uns par rapport aux autres pour construire un modèle du SYSTEME (**System Model in the Loop**). Tandis que ce chapitre s'est focalisé sur le raisonnement de co-spécification, le prochain chapitre s'intéresse à comment mettre en œuvre ce raisonnement de co-spécification SYSTEME basée sur des modèles.

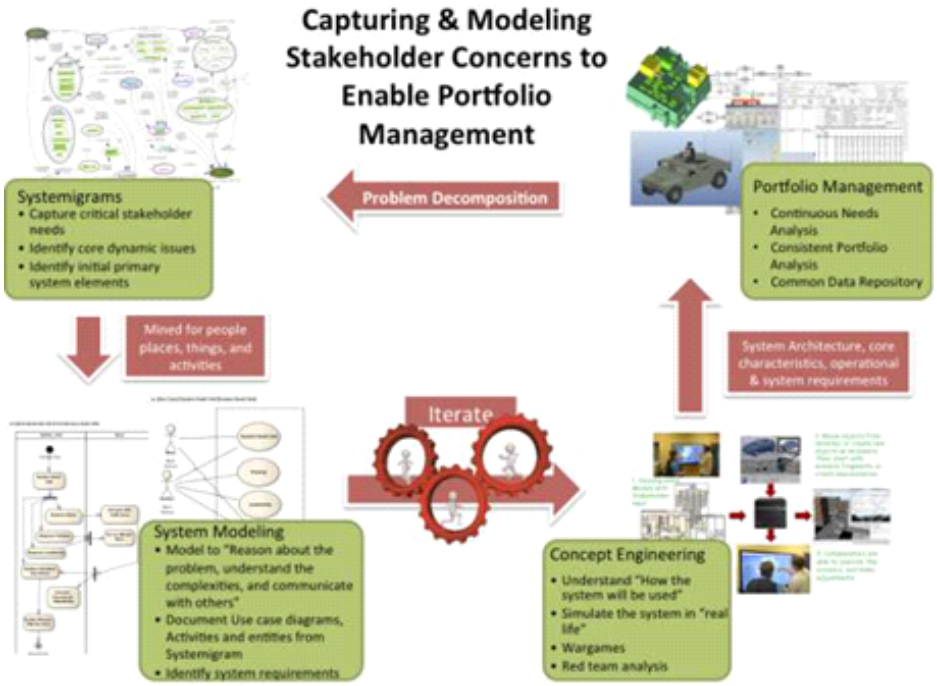


Figure 52 : Méthodologie pour la définition du problème selon une vision systémique (Boardman & Sauser, 2008)

31 Systemigram : est un mot valise provenant de systémique et de diagramme. Le systemigram est une représentation graphique qui encapsule un problème à travers les relations entre les différentes expressions des besoins des parties-prenantes.

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

4. Chapitre 4 : Environnement de spécification-système exécutable pour faire CISPI

4.1. Introduction : Caractérisation de l'environnement de spécification-système exécutable CISPI

Dans le chapitre précédent, notre heuristique de co-spécification a mis en évidence la nécessité de faire interopérer en « système » chacun des domaines de connaissances contribuant à une solution globale du SYSTEME d'intérêt visé. Notre objectif est que cette interopération de connaissances permette :

- D'une part, à chacun des domaines d'ingénierie spécialiste de représenter sa **connaissance** (K_h) sous forme d'un modèle ((M_h) d'une partie du système) vérifié dans son propre environnement de travail (E_h) et validé de façon collaborative avec l'environnement de travail d'ingénierie système (E_H),
- D'autre part, à ce dernier domaine d'architecturer l'ensemble de ces modèles (M_h) à partir de sa **connaissance** d'interdisciplinaire (K_I) et de sa **connaissance** systémique (K_H) pour construire un modèle du SYSTEME d'intérêt visé ((M_H) du SYSTEME vue comme un TOUT) pour satisfaire les exigences des parties prenantes utilisatrices (R_S).

De façon à faciliter cette co-construction itérative de (M_H) selon un agencement de (M_h), nous proposons dans un premier temps un environnement collaboratif (Figure 37) autour :

- D'un bus de co-spécification qui complète notre heuristique de raisonnement par un patron architectural et un langage pivot de modélisation système,
- D'un bus de co-simulation qui permet de Vérifier / Valider chacune des spécifications par co-exécution de modèles.

Nous illustrons dans un deuxième temps la mise en œuvre de cet environnement de travail collaboratif par un scénario opérationnel de co-spécification SYSTEME basée sur des modèles de la plate-forme CISPI, que nous développons dans les parties 3 et 4 de ce mémoire. Pour ce cas d'application, nous nous sommes placés dans une situation plausible d'interactions itératives entre le domaine de l'ingénierie système, sous notre responsabilité pour construire (M_H), et des domaines interdisciplinaires d'ingénieries auxquels des experts aussi bien industriels qu'universitaires ont accepté d'apporter leur contribution pour chacun des **(M_h)**.

4.2. Cadre de modélisation architecturale

De façon générale, un patron de conception permet de réutiliser l'expérience acquise et de la faire évoluer afin d'adapter une solution standard à une situation particulière d'ingénierie, notamment en génie logiciel.

> *« A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems. It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples. The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context » (Gamma, Helm, Johnson, & Vlissides, 1994)*

A un facteur d'échelle plus large, il en est de même en ingénierie système où ce type de modèles de référence, adapté à un contexte d'utilisation, assure une compréhension commune du SYSTEME en définition, développement, en déploiement, …, par l'ensemble des parties prenantes impliquées, par exemple sous forme de cadre de modélisation d'architecture d'entreprise.

As systems become more complex, the need for a method to capture and manage the implicit knowledge that exists within corporations and within the systems engineering discipline grows. This implicit knowledge must be documented, creating objective evidence, which can be applied over and over again to solve similar challenges. Patterns may be a partial solution to both of these needs (Cloutier, 2006).

Il en est de même dans nos travaux où la construction du modèle total du SYSTEME d'intérêt ($\boldsymbol{M_H}$) est guidée par une adaptation du « conceptagon » comme « system pattern » en regard de la situation d'intérêt mais doit aussi être guidé par un patron en regard de l'ensemble des ($\boldsymbol{M_h}$) partiels à architecturer. En regard de notre cible d'application des systèmes de conduite de procédé industriel, nous revisitons l'intérêt du cadre de modélisation architectural CMMS-IAMS[32] et de sa variante SIAC en tant que patron architectural dans la mesure où diverses adaptations et applications au cours du temps permettent de lui conférer, de fait, une dimension « pattern ».

4.2.1. Cadre de modélisation architecturale CMMS-IAMS

Ce cadre de modélisation architectural est issu de divers travaux antérieurs au laboratoire en collaboration avec la DER (Direction des Etudes et Recherches) d'EDF dans le cadre d'importants programmes européens de R&D[33]. Ces derniers ont montré l'intérêt d'architectures techniques interopérables, distribuant au plus près du procédé une forme d'« intelligence technique » afin d'intégrer, en un système de conduite, de maintenance et de gestion technique (CMMS), des systèmes d'actionnement et de mesure intelligents (IAMS) autour d'un bus de terrain (Figure 53) (Iung, 1992), (Pétin, 1995), (Neunreuther, 1998), (Leger, 1999) et (Liu, 2002).

Le patron architectural IAMS repose sur le patron architectural du module fonctionnel d'automatisme (MFA), proposé de façon similaire par (Galara, 1986) (Belhimeur, 1989)dans le domaine des systèmes de production continue. Quel que soit le niveau fonctionnel de la structure hiérarchique du système de conduite, l'activité interne du MFA est un ensemble de tâches organisées : validation des objectifs, validation des observations, élaboration des rapports et élaboration des actions. Ce concept architectural est proche de celui de « filtre de comportement » proposé dans le domaine de la production discrète (Lhoste, 1985) qui vise aussi à structurer la « partie contrôle-commande » à partir de la structuration de la « partie opérative » par composition d'objets « software $\wedge$ hardware » réutilisables parce que standardisés.

Dans les deux cas, l'« intelligence technique » embarquée au cœur des composants instrumentés vise aussi à filtrer des rétroactions positives dans le cas de consignes ou de mesures fausses par rapport à ce qui est attendu de chaque objet. Il s'agit là d'un concept essentiel dans notre approche puisque cette « intelligence technique » embarquée ($\boldsymbol{I_h}$) n'est que la partie implémentable de la **connaissance** ($\boldsymbol{K_h}$) d'un domaine d'automatisation centrée technique ou humaine.

[32] Paradigme CMMS-IAMS (Integrated Control Maintenance and technical Management System distributed on Intelligent Actuation and Measurement System) (AFIS - GT Ingénierie Système, 2005)

[33] ESPRIT II-DIAS (*Distibuted Intelligent Actuation and Sensors),* ESPRIT III PRIAM (*Prenormative Requirements for Intelligent Actuation and Measurement*), ESPRIT III EIAMUG (*European Intelligent Actuation and Measurement User Group*), ESPRIT IV IAM-PILOT (*Intelligent Actuation and Measurement Pilot*) , ESPRIT IV-REMAFEX (REmote Maintenance for Facility EXploitation)

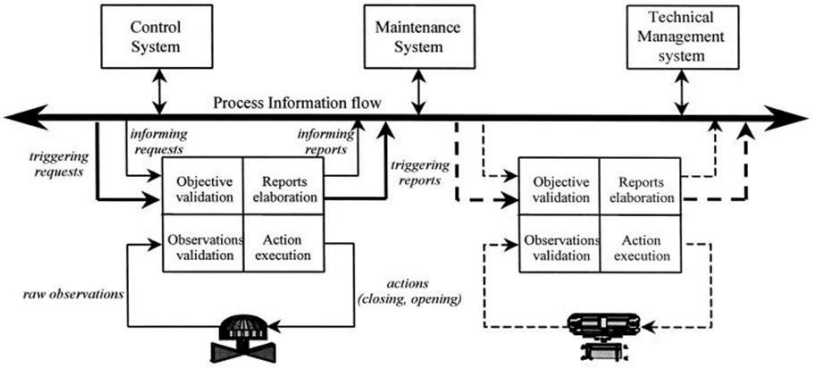


Figure 53 : Le Module Fonctionnel d'Automatisme dans le cadre CMMS-IAMS. (Pétin, Iung, & Morel, 1998)

Une démarche de spécification exécutable basée sur des modèles de ce type d'architecture a fait l'objet d'un développement méthodologique et d'un outil dédié dans le cadre du projet européen PRIAM (« Prenormative Requirements for Intelligent Actuation and Measurement »). Comme l'indique l'intitulé du projet, l'objectif était de capturer au plus tôt les exigences de chacune des parties prenantes avec pour objectif de faire converger le modèle fonctionnel CMMS attendu des utilisateurs et les modèles technologiques AMI développés par des offreurs dans un cadre commun d'interopérabilité système pour les intégrateurs. Dans cette optique, plusieurs niveaux de classification des exigences ont été proposés afin de permettre à un modélisateur d'exprimer de manière systématique les services attendus d'un système d'actionnement et de mesure intelligent selon deux niveaux d'abstractions : fonctionnel indépendant de toute répartition dans des équipements et de toutes contraintes organisationnelles de l'entreprise, et organisationnel tenant compte de la répartition géographique des fonctions propres à chaque site de production. Ces exigences sont spécifiées selon les points de vue contrôle-commande, maintenance et gestion technique, mais également selon un regroupement des fonctions d'actionnement et de mesure intelligents en sept classes plus génériques (documentation, configuration, paramétrage, test, état opérationnel, historiques, modes d'utilisation) (Figure 54) (Pétin, 2007)

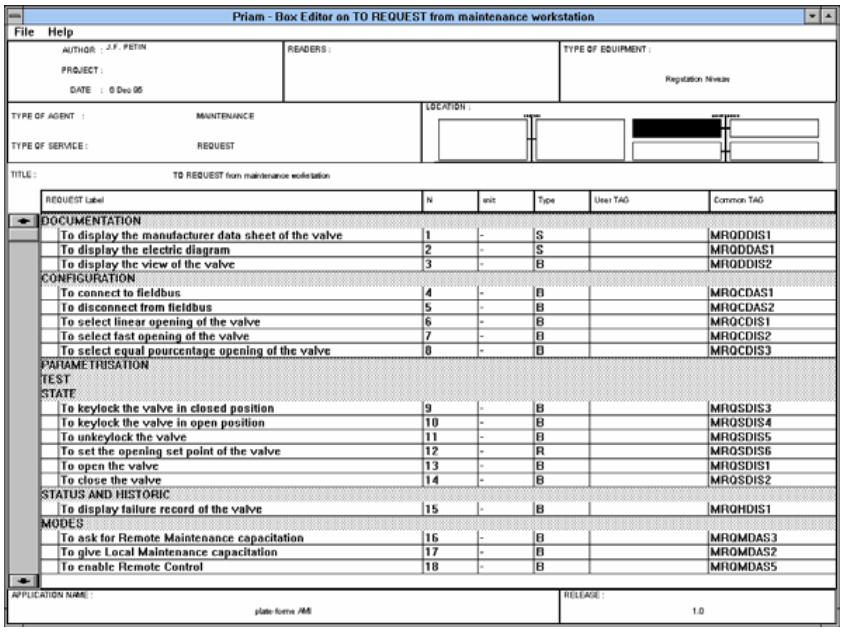


Figure 54 : Exemple d'expression des exigences pour un agent de maintenance avec l'outil PRIAM

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

Afin de permettre la description du comportement d'un système AMI mais également de simuler son fonctionnement, l'outil de spécification comportementale exécutable SPEX[34] (Tixador, 1989) (Panetto, 1991)a été intégré à l'outil PRIAM (Iung, 1992). Cet outil permet de modéliser des comportements discrets dans des formalismes tels que le Grafcet, le langage à relias, le langage à blocs fonctionnels (Cf. Annexe 4-1, Figure 182). Ces modèles sont ensuite simuler de manière interactive par l'intermédiaire d'un tableau de bord, ou de manière automatique par scénario de test et analyse de traces (Cf. Annexe 4-1, Figure 183) pour s'assurer du respect des exigences exprimées par les différentes parties prenantes aux différents niveaux de modélisation dans l'outil PRIAM. Afin de réaliser une simulation dans un contexte émulé d'exploitation, l'outil SPEX permet de coupler les modèles comportementaux spécifiés à des modèles d'émulation des constituants (vannes, pompes, transmetteurs, …) du système AMI.

Cet ensemble d'outil (PRIAM & SPEX) constitue un environnement de co-spécification exécutable basée sur des modèles. Cependant, au regard de notre SYSTEME d'intérêt relatif à la conduite d'un procédé industriel critique, cet environnement trouve ses limites dans la modélisation des comportements continus. En effet, l'outil SPEX permet de modéliser uniquement les comportements à évènements discrets. En ce sens, nous proposons un environnement de co-spécification exécutable basée sur des modèles autour d'un bus de co-simulation afin de pouvoir intégrer l'ensemble des outils spécialistes les plus adaptés pour représenter et simuler les différentes facettes d'un SYSTEME.

4.2.2. Cadre de modélisation architecturale SIAC

Alors que les travaux présentés précédemment s'étaient limités à l'interopérabilité entre fonctions techniques CMMS et IAMS à des niveaux opératoires différents, (Dobre, 2010) les a revisités récemment pour accroitre l'interactivité technique avec les opérateurs humains, en exploitation (Figure 55a) et en ingénierie (Figure 55b).

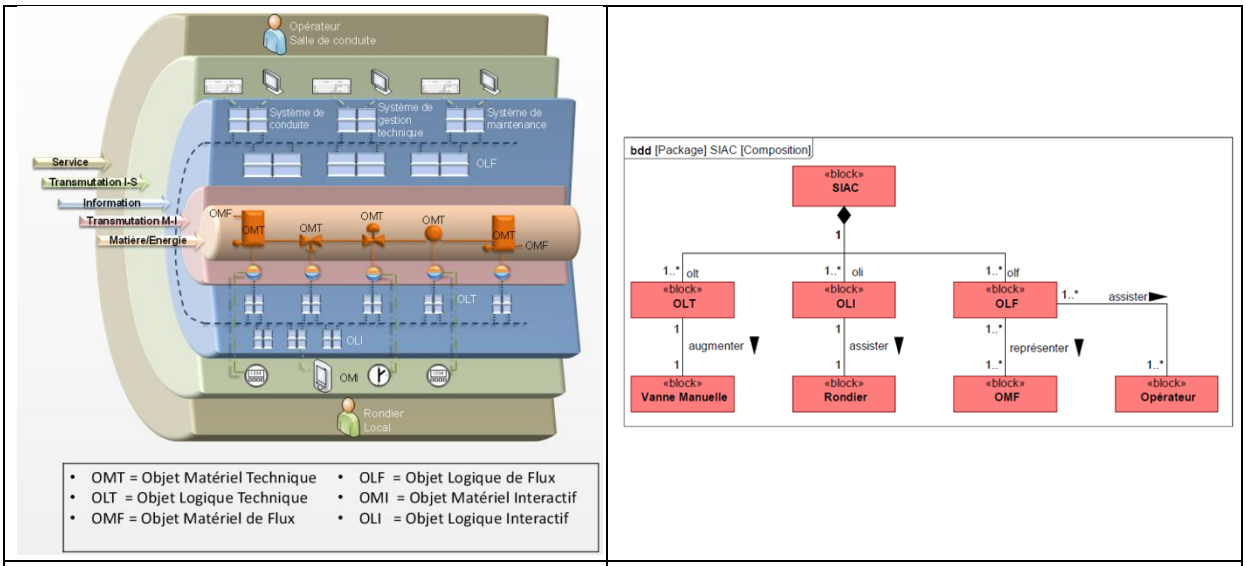


Figure 55a : en exploitation (interactivité architecturale)	Figure 55b : en ingénierie (Modélisation SysML)

Figure 55 : Paradigme architectural de Système Interactif d'Aide à la Conduite (Dobre, 2010)

[34] Les développements ultérieurs de SPEX ont conduit à l'outil ControlBuild (http://www.3ds.com/products-services/catia/capabilities/embedded-systems/controlbuild/) faisant partie désormais de l'environnement 3DS de Dassault System

L'objectif est d'assurer un continuum d'informations pour permettre aux opérateurs en local d'interopérer numériquement en tout lieu, à tout instant et pour toutes actions de conduite via des objets interactifs « logiques » (fournissant au rondier des services d'assistance au cours de ses activités) et « matériels » (fournissant l'interface Homme-Machine).

D'un point de vue ingénierie, la spécification du SIAC repose sur une ingénierie système basée sur des modèles en langage SysML (System Modeling Language) et guidée par des processus de la norme (ISO/IEC 15288, 2008). Dans la mesure où SysML n'a pas de règles de preuves, par exemple de raffinement d'un élément de modélisation en préservant des invariants, qui en ferait un langage formel de spécification (Bjørner & Henson, 2007), (Dobre, 2010) propose un processus itératif de spécification système, en spécialisant la syntaxe et la sémantique de SysML pour supporter le processus proposé.

> *« It is important to note that the UML and SysML are not software or systems methodologies but rather visual modelling languages that are agnostic to any one specific methodology. » (Estefan, 2007)*

En ce sens, Dobre effectue une correspondance entre les éléments du méta-modèle de SysML et les artefacts clefs extraits de certaines bonnes pratiques en ingénierie système, formalisées sous la forme de processus. Cela le conduit à s'intéresser aux travaux de Jackson en génie informatique, présenté dans le chapitre 3, afin de formaliser le processus de spécification, vu alors comme un processus central en ingénierie système.

Notre formation en ingénierie système, ainsi que notre expérience en enseignement nous ont montrés les limites et les difficultés de mettre en place une approche SYSTEME seulement guidée par les processus standardisés ou supportée par des techniques diverses de modélisation (langage, pattern, …) sans compréhension en préalable de leurs fondements systémiques comme précisé par Meinadier dans la construction de l'ouvrage de référence « Découvrir et Comprendre l'Ingénierie Système » (Fiorèse & Meinadier, 2012).

4.2.3. Cadre de modélisation architecturale Système

Dans la mesure où, par son usage multiple, le paradigme CMMS-IAMS fait œuvre de « patron » en ingénierie, nous en avons revisité les fondements systémiques sans en développer sa représentation syntaxique.

En effet, le paradigme architectural CMMS rompt avec l'approche cartésienne courante consistant à spécifier séparément le sous-système de production, le sous-système de maintenance et le sous-système de gestion de l'information d'un sous-système technique. Chacun de ces éléments a pour finalité de contrôler la transformation dans la forme de l'objet finalisant, de la maintenir dans le temps et à travers l'espace. L'action de gestion organisatrice (Morin, 1980) a en plus pour rôle de réfléchir le comportement SYSTEME de l'ensemble à l'environnement. En ce sens, cette action de gestion complète la vision système courante en ingénierie système consistant à partitionner le SYSTEME d'intérêt en sous-système principal (« End Products ») et en sous-système de soutien (« Enabling Product »), sans expliciter comment ces deux parties sont reliées entre elles et à leur environnement commun.

Ainsi, l'organisation d'un SYSTEME selon le paradigme architectural CMMS lui permet de se contrôler, se maintenir et se relier à son environnement (Le Moigne, 1994), confirmant ainsi l'analogie système proposée par (Mayer, 1995) avec celle des travaux synthétisés en chapitre 3 autour du paradigme holonique. Cette

vision holonique d'un SYSTEME est également à la base du concept architectural « Holonic Manufacturing System » (Valckenaers, Van Brussel, Bongaerts, & Wyns, 1997)et de ses variantes de « contrôle par le produit »[35] (G. Morel & Grabot, 2003).

En ce sens, nous intégrons les travaux de (Fliess & Join, 2013) relatifs à la commande sans modèle (CSM) pour spécifier le comportement « générique » d'un agent de conduite. De par sa formulation, la CSM permet de relier partiellement les sous-systèmes de production, gestion de l'information et maintenance, en adaptant le comportement du procédé à conduire aux variations mesurables de son environnement tout en fournissant une information relative à la détection d'une dérive de ce comportement à des fins de surveillance ou de gestion.

De plus, nos travaux revisitent le paradigme architectural du MFA, structurant les activités de conduite d'un agent technique ou humain selon le paradigme holonique. Ainsi, un agent-holon est spécifié à travers son rôle (Figure 56) de :

- **Commande** en regard des services de conduite requis par sa part extérieure à laquelle il doit rendre compte,
- **Contrôle** en regard des actions et observations établies avec sa part intérieure dont il doit aussi préserver l'intégrité pour assurer ces services.

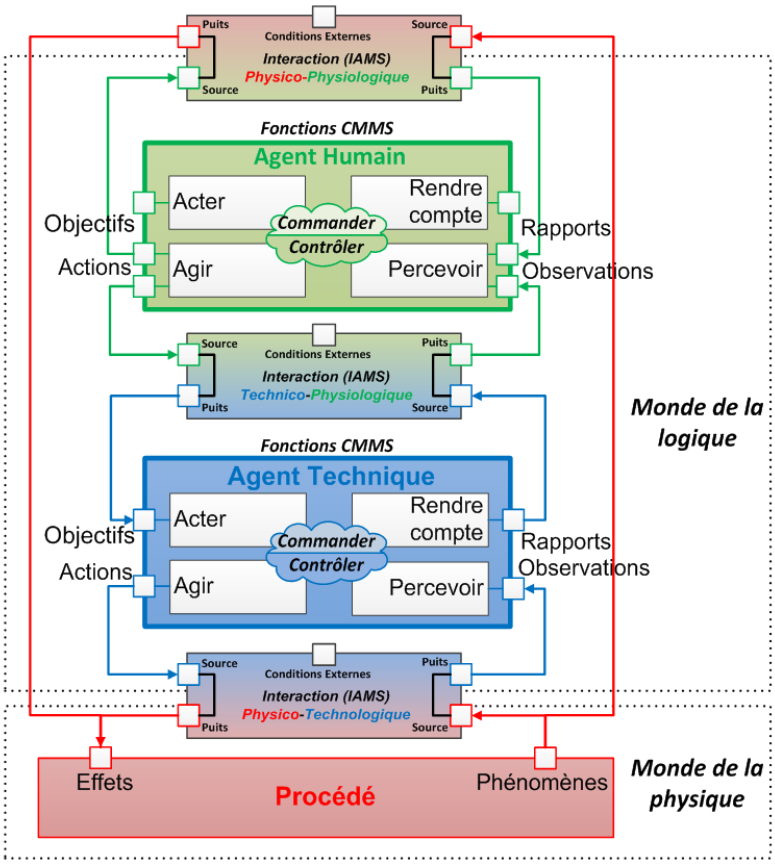


Figure 56 : Patron architectural logico-fonctionnel CMMS-IAMS proposé

[35] http://www.univ-valenciennes.fr/gdr-macs/node/4416

Notons que nous retrouvons cette structuration chez EDF au niveau des spécifications du contrôle-commande d'une centrale de production d'énergie électrique sous la forme de :

- Diagrammes Fonctionnels Logiques (DFL) en tant que spécification des traitements logiques et séquentiels, relatifs aux fonctions de commande.
- Diagrammes Fonctionnels Analogiques (DFA) en tant que spécification des traitements analogiques et des régulations, relatifs aux fonctions de contrôle.

Notre proposition de patron architectural système (Figure 56) précise également les activités composant le MFA en se focalisant d'avantage sur les rétroactions mise en œuvre au cours de la conduite, et en les généralisant à un système sociotechnique. Ainsi quel que soit le type d'agent, ce dernier est en mesure d'« ***Acter*** » le service requis et de « ***Rendre–compte*** » à l'environnement des rétroactions de commande menées pour effectuer la mission assignée. Ces rétroactions de commande supervisent des rétroactions de contrôle visant à « ***Agir*** » et « ***Percevoir*** » pour maintenir le système dans un état donné en fonction de phénomènes émergeants de façon générale, et pour satisfaire les rétroactions requises par la commande de façon plus spécifique.

Notons que cette intelligence distribuée (embarquée dans) sur des agents techniques ou humains en exploitation n'est qu'une partie implémentable des connaissances des domaines d'ingénierie spécialistes impliqués dans la co-spécification du SYSTEME de conduite, sous la forme de programmes, algorithmes,… d'un point de vue technique, ou sous la forme de procédures, de réglementations,… d'un point de vue humain. En ce sens, ce patron architectural permet d'explorer le SYSTEME d'intérêt selon la triade ***« Communication, Commande, Contrôle »*** du conceptagon. L'aspect **communication** se concrétise à travers trois types d'interactions différentes qui dépendent de la nature des agents et du procédé en interactions (Bouffaron, Dupont, et al., 2014) :

- **Interaction Physico-Technologique** : pour assurer l'interface entre le procédé et le niveau des automatismes. Elle comprend les actionneurs et les capteurs pouvant intégrer des traitements (IAMS) et permettant de construire une représentation partielle (*dans le monde de la logique*) du procédé (*dans le monde de la physique*) et restreinte aux performances des capteurs et du problème récurrent de leur localisation.
- **Interaction Physico-Physiologique** : pour assurer l'interface entre la composante humaine du SYSTEME de conduite et le procédé. Les opérateurs interagissent directement avec les équipements du procédé (vannes, pompes, robinet, afficheur numérique, disjoncteur) à travers leur appareil musculo-squelettique pour actionner les équipements et leur appareil sensoriel pour observer les phénomènes, les traiter et remonter l'information concernant l'exploitation du procédé industriel (Galara, 2006). Nous étudions plus en détail cette interaction dans le chapitre 8 de notre mémoire.
- **Interaction Technico-Physiologique** : pour assurer l'interface entre la composante technique et la composante humaine du SYSTEME de conduite, et peut être associée aux interfaces Homme-Machine aussi bien en salle de commande que sur le terrain.

La co-spécification holistique du SYSTEME de conduite implique de bien définir le contexte où le SYSTEME va évoluer et interagir avec les différents objets de contexte avant de décrire sa structure interne. Ainsi, une attention particulière doit être portée sur la modélisation des manifestations externes que le SYSTEME requis doit manifester dans son environnement. L'ensemble des phénomènes contenus dans une interaction peuvent changer dynamiquement son comportement, c'est pourquoi en cours de co-spécification il est important de considérer les interactions entre des éléments comme un objet à part entière (Ducroq, 1996). En ce sens, le

patron proposé (Figure 56) vise à spécifier les éléments de cet objet-interaction entre un objet source et un objet puits. L'objectif étant de bien définir les propriétés physiques du flux de matière/énergie provenant de l'objet source et propagé à travers l'interaction (*pouvant être perturbés par son environnement (conditions externes)*) pour que l'objet puits puisse percevoir cette interaction. Cela permet de définir un premier niveau d'exigence pour les interfaces entre les agents techniques et/ou humains et le procédé.

4.3. Environnement de co-modélisation et co-simulation exécutable

Nous détaillons dans ce paragraphe les éléments de notre proposition d'environnement de travail collaboratif pour échanger, co-spécifier, co-exécuter l'ensemble des modèles à architecturer en « système » afin de valider la cohérence globale du modèle du SYSTEME ($\boldsymbol{M_H}$) au regard des exigences parties prenantes.

4.3.1. Langage d'interopération système

Une telle organisation (Figure 37) nécessite la mise en œuvre d'une ingénierie collaborative. En effet, il ne suffit pas de réunir des spécialistes dans un même espace mais d'assurer une coordination technique et un lien sémantique entre ces ingénieries.

> *« Dans une équipe pluridisciplinaire, la **difficulté majeure**, consiste à **dialoguer entre spécialistes**, du fait de l'utilisation de langages, et de modèles de représentation différents, correspondant aux différentes propriétés étudiées et aux différents métiers concernés. Une partie de la réponse à cette problématique réside dans l'acquisition par chacun d'un **langage commun d'ingénierie minimal** standardisé. Une autre partie de la réponse réside dans la constitution d'un **référentiel de définition partagé**, mis à jour aussi souvent que possible, accessible par chaque membre de l'équipe » (Faisandier, 2014).*

Il convient alors de traduire l'essentiel des informations contenues dans les modèles spécialistes ($\boldsymbol{M_h}$), dans un langage d'interopération (SysML System Modeling Language) pour que le domaine d'ingénierie système puisse agencer l'ensemble de ces modèles comme modèle du SYSTEME d'intérêt visé ($\boldsymbol{M_H}$).

En effet, SysML est le langage do-facto de l'ingénierie système, ayant pour objectif de supporter les approches d'ISBM tout au long de cycle de vie du SYSTEME comme la spécification des exigences, la modélisation des différents constituants avec leur structure et leur comportement, l'intégration, ainsi que la spécification de scénarios pour la vérification et la validation. Il trouve son origine dans l'intérêt des ingénieurs système pour la conception orientée objet et le langage UML (Unified Modeling Language) dans le domaine d'ingénierie informatique et fût initié par une collaboration, en 2001, entre l'INCOSE (International Council of Systems Engineering) et l'OMG (Object Management Group). Aujourd'hui la dernière version de SysML est la version 1.4 adoptée en mars 2014. Notons que la version de SysML sur laquelle nous nous sommes référés est la version 1.2, qui était déjà en vigueur au commencement de notre thèse. De plus, à l'heure actuelle peu d'outils SysML intègre totalement la dernière version de SysML v1.4.

Ainsi, SysML est un profil UML version 2.0 qui réutilise des éléments de ce dernier (« *UML4SysML* ») et ajoute certaines extensions propres à l'ingénierie système sous la forme de deux nouveaux diagrammes : le diagramme d'exigence (*permet de collecter et d'organiser toutes les exigences textuelles du système*) et le diagramme paramétrique (*permet de modéliser les paramètres physiques du système, servant ainsi à tester les performances physiques et quantitative*). En ce sens SysML est organisé en trois catégories (diagrammes structurels, comportementaux et transversaux) regroupant neuf diagrammes au total (Figure 57).

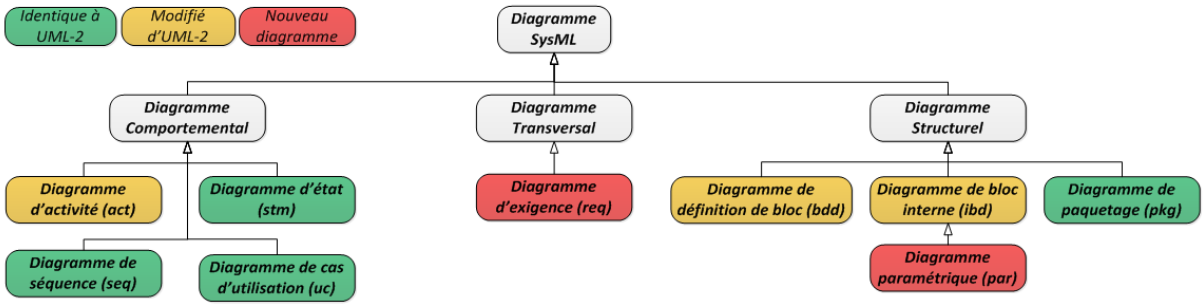


Figure 57 : Types de Diagrammes SysML) (OMG, 2012a)

Divers rapports ont montré un taux d'usage très différent des différents diagrammes SysML. Notamment, celui de (Cloutier & Bone, 2010) pour l'OMG présente le diagramme paramétrique comme étant le diagramme SysML le moins utilisé alors que c'est l'un des deux diagrammes différenciant « système ». Si un diagramme d'exigences permet un usage linguistique plus ou moins bien fondé des exigences, le diagramme paramétrique moins facilement compris et mis en œuvre puisque les équations représentées n'en font pas un diagramme exécutable, et que les liens avec les autres diagrammes SysML ne sont pas clairement définis. Nous apportons des éléments de réponses à ces problématiques dans le chapitre cinq de notre mémoire, et montrons également l'intérêt essentiel du diagramme d'activité pour décrire le comportement global du procédé à conduire.

Notons, que SysML est destiné à supporter deux standards d'interopérabilité qui sont : le standard XMI 2.1 pour l'échange de modèles entre les outils de modélisation supportant UML-2 et le standard ISO 10303 STEP AP233 pour l'échange de données entre les outils d'ingénierie système. Ainsi, ces standards vont fournir des mécanismes d'échange de données, qui pourront éventuellement être représentées dans d'autres langages de modélisation (propres aux ingénieries spécialistes).

Ainsi, nous prescrivons que l'ensemble des domaines d'ingénieries spécialistes traduisent le résultat de leur travail en langage SysML, soit de manière automatisée par échange de diagrammes entre les outils, soit de manière manuelle en appliquant des règles de traduction. Dans la suite du mémoire, nous explicitons un certain nombre de règles d'interopérabilité (« **transitioning** ») entre les langages de modélisation des domaines d'ingénieries spécialistes et le langage SysML. Ces spécifications doivent être traduites à minima en vision boite noire en prescrivant les interfaces d'entrée-sortie, pour que le domaine d'ingénierie système puisse architecturer l'ensemble des modèles des PARTIES du système pour construire un modèle du SYSTEME comme un TOUT. La vision boite blanche peut être requise dans le cas où l'ingénieur système a besoin de valider des solutions plus en détails par rapport à des choix architecturaux, des performances, …

En complément de (Dobre, 2010), nos travaux nous ont également conduit à spécialiser la syntaxe et la sémantique SysML pour supporter notre heuristique de co-spécification SYSTEME. Nos travaux se sont principalement focalisés sur les artefacts de modélisation du diagramme d'exigences SysML (Tableau 5).

Processus/ Artefact IS	Source	Puits	Artefact SysML
Description	Exigence Partie Prenante *Espace Problème*	Exigence Partie Prenante *Espace Solution*	Dépendance « Trace »
Prescription	Exigence Système *Espace Solution*	Exigence Système – *Espace Problème*	Dépendance « Trace »
Analyse des exigences	Exigence niveau d'abstraction n	Exigence niveau d'abstraction n+1	Dépendances « Refine, Derive, Requirement containment relationship »
Vérification	Cas Test	Exigence	Dépendance « Trace » stéréotypée « Verify »
Validation	Cas Test	Exigence	Satisfy
Connaissance (K)			Rationale
Problème (P)			Problem
Exigence (R)			Requirement
Specification (S)			« Requirement » stéréotypée « System_Requirement »

Tableau 5 : Spécialisation du diagramme d'exigences SysML, supportant notre heuristique de co-spécification SYSTEME (Gouyon, et al., 2014)

4.3.2. Bus de co-spécification système

Le bus de co-spécification tel que présenté dans la Figure 37, doit être perçu ici entant que concept opérationnel du système pour faire. Ce dernier distribue les ingénieries spécialiste autour d'un bus afin de partager et de faciliter l'accès à leurs résultats de spécifications sous formes de modèles dans un référentiel commun de définition du SYSTEME. Ce bus fonctionnel de co-spécification SYSTEME peut être mis en œuvre à travers différentes technologies de travail collaboratif telles que :

- **Les plate-formes de travail collaboratif**, qui centralisent tous les outils liés à la conduite d'un projet, la gestion des connaissances et les mets à disposition des acteurs. Ce type de plate-forme intègre les fonctionnalités suivantes : base de connaissances structurées, service de messagerie, système de partage de ressources et de fichiers, outils de type forum, chat, calendrier, index, système de vote… Parmi les outils existants sur le marché nous pouvons citer : *IBM Collaboration Solutions*[36], *SharePoint*[37], *eRoom*[38]…
- **Les plate-formes PLM** (Product Life cycle Management), qui visent à gérer et partager l'ensemble des données d'ingénieries et d'exploitation d'un produit depuis sa conception jusqu'à sa fin de vie. Ces informations sont partagées entre toutes les entités d'une entreprise de la conception, à la vente en passant par la production. Ce type de plate-forme intègre les fonctionnalités suivantes : gestion des exigences, gestion des données techniques, visualisation et maquettes numériques, gestion des modifications et de la configuration, gestion des processus de fabrication, gestion de la qualité, gestion de projets. Parmi les outils existants sur le marché nous pouvons citer : *Enovia - Dassault System*[39], *IBM Rational Jazz*[40], *Siemens PLM Software*[41], *Windchill*[42], *Audros*[43]… Notons

[36] http://www-01.ibm.com/software/fr/lotus/
[37] http://products.office.com/en-us/sharepoint/collaboration
[38] https://www.eroom.net/
[39] http://www.3ds.com/products-services/enovia/
[40] https://jazz.net/
[41] http://www.plm.automation.siemens.com/fr_fr/

également l'initiative industrielle menée initialement par IBM en 2008 et ayant conduit à la spécification du standard OSLC (Open Services for Lifecycle Collaboration) (Workgroup OSLC, 2010), dans le but de simplifier l'intégration des outils tout au long du cycle de vie des logiciels et systèmes. Ce standard a pour objectif de permettre à des équipes interdisciplinaires d'utiliser leurs propres outils, de partager et de tracer les données d'ingénieries entre ces outils.

Parmi les solutions proposées, nous avons mis en œuvre un espace collaboratif avec l'outil « Quickplace » qui est une solution intégrée à l'offre IBM Collaborations Solutions. « Quickplace » permet la création d'espaces collaboratifs de travail pour le web à travers internet. C'est une solution de gestion de projet qui offre la possibilité d'organiser des contenus, des tâches, des évènements et des documents. Cette application intègre également un forum, un agenda et un index.

L'organisation que nous proposons pour la structuration de l'espace « Quickplace » est dérivée de notre heuristique de co-spécification entre domaines de connaissances, eux-mêmes partagés entre espace-solution et espace-problème (Figure 58). Notons que cette organisation a été éprouvée à travers différents projets pédagogiques relatifs à nos activités d'enseignement (G. Morel, et al., 2014).

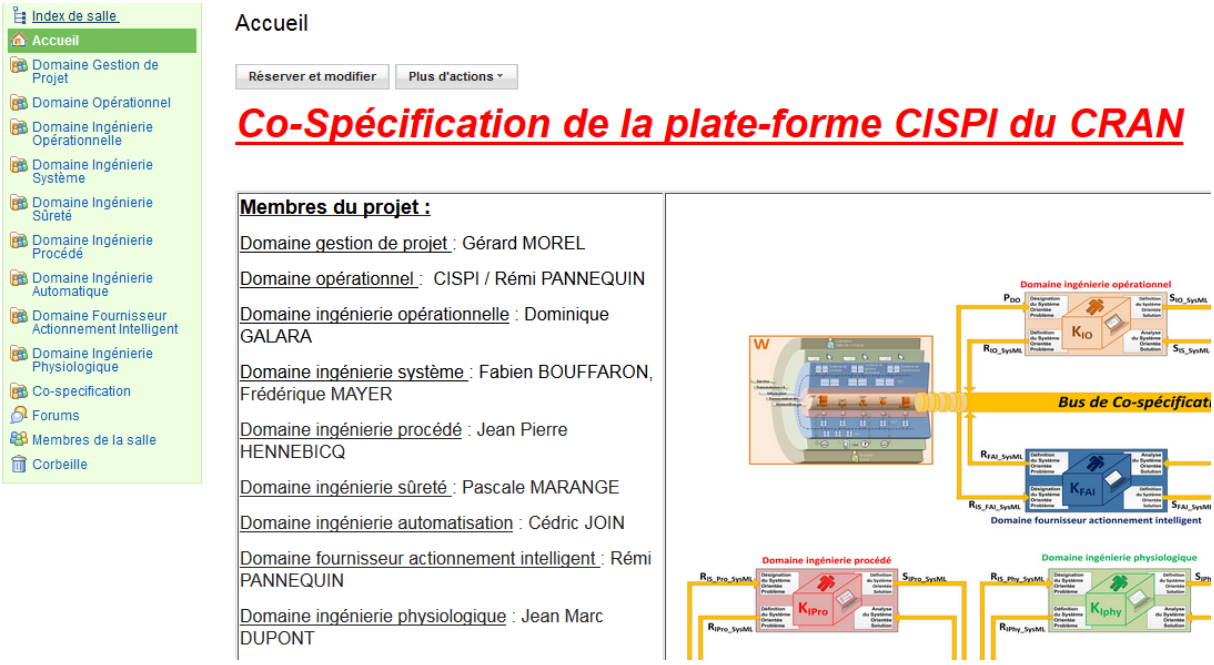


Figure 58 : Organisation de l'espace "Quickplace" par domaines d'ingénieries spécialistes

Chacun des domaines d'ingénierie est concrétisé par une salle portant le nom du domaine d'ingénierie concerné. Chacune des salles comporte un dossier « Membres » définissant les utilisateurs autorisés à y accéder. Un intérêt de séparer les domaines d'ingénieries par salles est que nous pouvons définir les utilisateurs autorisés à y accéder. Le savoir-faire et les connaissances métier sont ainsi conservés à l'intérieur

[42] http://fr.ptc.com/product-lifecycle-management/windchill
[43] http://www.audros.fr/

du domaine d'ingénierie sans être partagés avec les autres membres du projet. Chacune de ces salles sont-elles mêmes structurés en trois dossiers représentant la connaissance appartenant aux domaines d'ingénieries ainsi que les espace-problème et espace-solution composant ce domaine. Le dossier de connaissances contient l'ensemble des connaissances du domaine d'ingénierie et sont instanciés sur la situation d'intérêt d'un point de vue problème ou solution dans les dossiers correspondant aux espaces problème ou solution (Cf. Annexe 4-2, Figure 185).

De plus, nous créons une nouvelle salle que nous nommons « Co-spécification ». Cette dernière concrétise le bus de co-spécification, c'est-à-dire l'espace d'échange des spécifications descriptives et prescriptives entre domaines. Cette salle est structurée en dossiers correspondant aux relations de spécification entre les différents espaces problèmes et solutions des domaines d'ingénierie spécialiste impliqués dans le projet de co-spécification. Ces dossiers de spécification sont eux-mêmes divisés en deux sous-dossiers : *« Spécification Descriptive* et *Spécification Prescriptive* » où les différents espaces d'ingénierie y déposent le résultat de leurs travaux (Cf. Annexe 4-2, Figure 186). En ce sens, cette salle « Co-spécification » constitue un référentiel de définition partagé entre l'ensemble des domaines d'ingénierie spécialiste ainsi que le domaine opérationnel.

4.3.3. Bus de co-simulation système

Comme décrit dans le chapitre 2, la maturité croissante des environnements de travail autour des ateliers de modélisation système permet d'envisager la co-exécution de modèles autour d'un bus de co-simulation. La construction itérative du modèle du SYSTEME architecturant un ensemble de modèles des parties sous-tend de mettre en place une démarche méthodologique itérative dans l'activité de vérification et de validation, en relation avec les connaissances des différents domaines d'ingénierie spécialiste nécessaires pour l'analyse des modèles. Cette étape est primordiale pour s'assurer que le modèle du SYSTEME satisfait l'ensemble des exigences parties prenantes.

Notre postulat de travail privilégie que les domaines d'ingénierie spécialistes utilisent leurs propres environnements de modélisation, méthodes et outils au cours de la co-spécification SYSTEME. Il nous a donc fallu trouver un outil permettant de relier l'ensemble de ces outils spécialistes autour d'un bus de co-simulation. Notre choix s'est porté sur l'outil CosiMate®[44]. En effet, au commencement de notre thèse, cet outil était le seul à notre connaissance à pouvoir interconnecter des outils de modélisation SYSTEME avec le langage SysML (*En l'occurrence IBM® Rational® Rhapsody®*) avec d'autres outils d'ingénieries spécialistes. Aujourd'hui, le standard FMI ouvre des perspectives d'un point de vue industriel pour la co-simulation entre de nombreux outils (Bertsch & Schulmeister, 2014). Récemment des outils de modélisation SysML ont intégré ce standard, cependant les travaux initiés ne sont pas encore suffisamment matures selon nous pour changer d'environnement de co-simulation en cours de thèse.

De plus l'outil CosiMate® offre la possibilité de développer nos propres coupleurs entre le bus et des outils soit en langage Java, soit en C++. Cette flexibilité s'est avérée très utile dans nos travaux, puisqu'elle nous a permis de développer un coupleur entre le bus de co-simulation et le serveur OPC (OLE for Process Control) de la plate-forme CISPI, ce qui nous a permis d'effectuer une transition des simulations « Model in the Loop » vers des simulations «Software - Hardware in the loop » (Figure 59). Cela nous a permis d'évaluer et d'éprouver nos spécifications dans des environnements de plus en plus matures, comme présenté dans le chapitre sept de ce mémoire.

[44] http://site.cosimate.com/

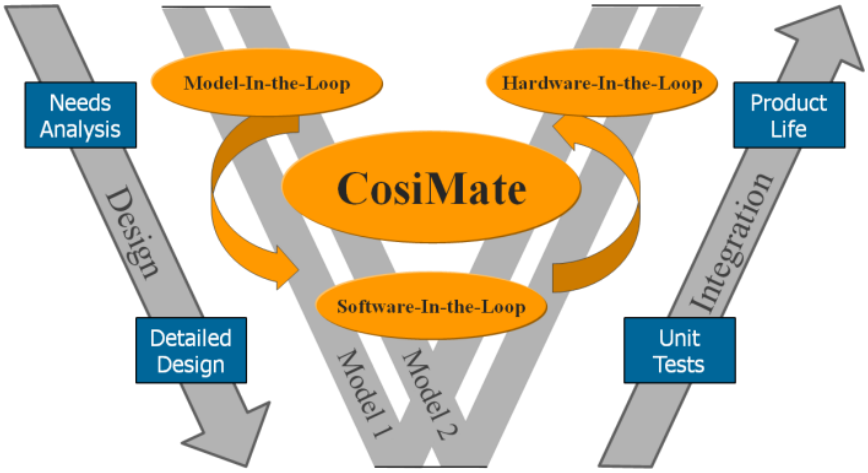


Figure 59 : Utilisation du bus de co-simulation tout au long du cycle de vie SYSTEME

D'un point de vue technique, CosiMate® offre un environnement de simulation globale liant dynamiquement des simulateurs hétérogènes (Figure 60) représentant des systèmes physiques et mathématiques à différents niveaux d'abstraction (Figure 59). Le bus de co-simulation CosiMate® agit comme un contrôleur de communication entre chacun des couples modèle/simulateur communiquant avec le bus à travers des ports d'entrée/sortie intégrés aux modèles.

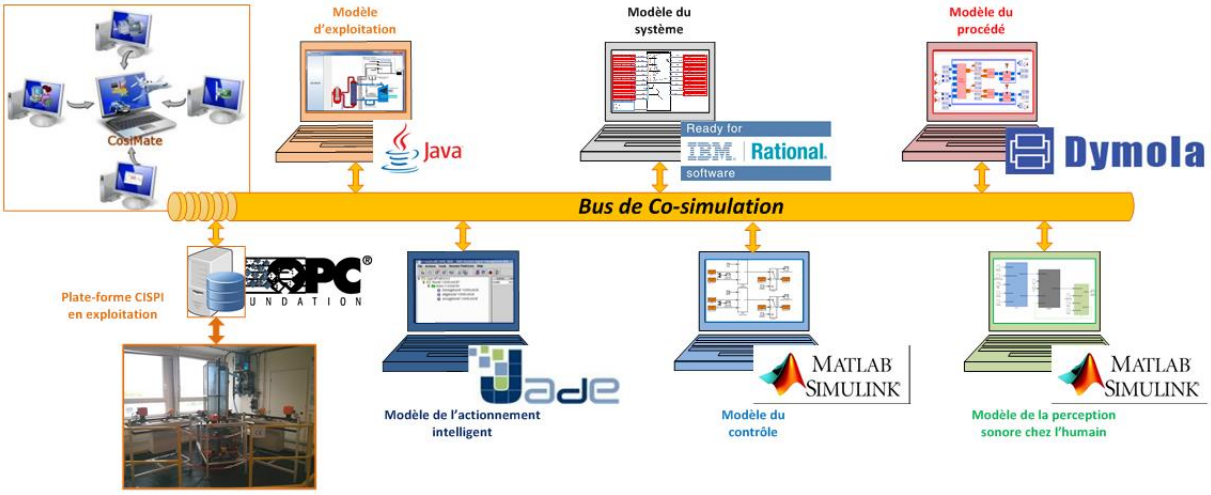


Figure 60 : Environnement de co-modélisation de la plate-forme CISPI

Notre environnement de co-modélisation de la plate-forme CISPI (Figure 60) est ainsi composé :

- Au niveau du domaine opérationnel, de la plate-forme CISPI existante reliée au bus de co-simulation à travers un programme que nous avons développé en langage Java couplant le serveur OPC de la plate-forme au bus de co-simulation. Le serveur OPC fournit une interface permettant à différents logiciels d'accéder aux données des dispositifs de contrôle-commande de la plate-forme comme les automates.

- Au niveau du domaine d'ingénierie opérationnelle, d'un modèle de l'environnement de notre SYSTEME d'intérêt développé en langage Java. Ce modèle est encapsulé dans un programme permettant d'effectuer le couplage de ce dernier au bus de co-simulation Figure 61.

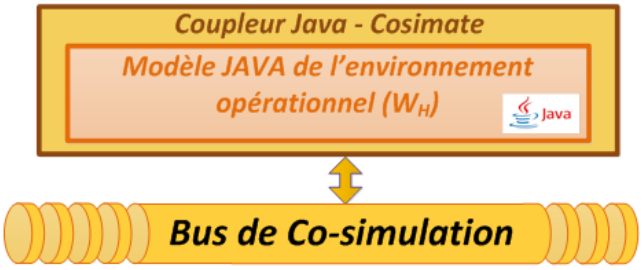


Figure 61 : Couplage du modèle de l'environnement au bus de co-simulation

- Au niveau du domaine d'ingénierie système, d'un modèle du système d'intérêt mais également de sa partie commande. Les modèles sont développés avec l'outil IBM® Rational® Rhapsody®[45] en langage SysML. Cet outil est un environnement d'ingénierie système basée sur des modèles qui permet d'utiliser des prototypes pour accélérer la conception. Cet outil a été retenu par le domaine d'ingénierie système que nous représentons, pour ses capacités à simuler et exécuter les diagrammes SysML permettant de valider au plus tôt le comportement du SYSTEME spécifié.
- Au niveau du domaine d'ingénierie du procédé, d'un modèle du procédé de la plate-forme CISPI spécifié en langage Modelica avec l'outil Dymola®[46]. Dymola est un outil de modélisation et de simulation de systèmes multi-physiques (électrique, thermodynamique, pneumatique, hydraulique et système de commande) basé sur le langage Modelica.
- Au niveau du domaine d'ingénierie d'automatisation, d'un modèle de la partie contrôle du procédé, spécifié dans un diagramme fonctionnel avec l'outil Matlab®/Simulink®[47]. Cet outil est un environnement d'ingénierie dirigé par les modèles, de conception et de simulation multidomaines. Il prend en charge la conception et la simulation au niveau système, la génération automatique de code, ainsi que le test et la vérification en continu des systèmes embarqués.
- Au niveau du domaine fournisseur d'actionnement intelligent, d'un agent JADE[48] (Java Agent DEvelopment framework) d'actionnement intelligent et couplé au bus de co-simulation à travers le serveur OPC de la plate-forme CISPI. JADE est un logiciel codé dans le langage Java et simplifiant la mise en œuvre de systèmes multi-agents à travers un intergiciel conforme aux spécifications du FIPA (Foundation for Intelligent Physical Agents) et d'un ensemble d'outils graphiques qui prennent en charge les phases de débogage et de déploiement.
- Au niveau du domaine d'ingénierie physiologique, d'un modèle de l'interaction physico-physiologique sonore par un agent humain spécifié également dans un diagramme fonctionnel avec l'outil Matlab®/Simulink®.

[45] http://www-03.ibm.com/software/products/fr/ratirhap
[46] http://www.3ds.com/fr/products-services/catia/capabilities/modelica-systems-simulation-info/dymola
[47] http://fr.mathworks.com/products/simulink/CMMS-IAMS.
[48] http://jade.tilab.com/

4.4. Scénario d'une co-spécification exécutable d'un modèle total du système de conduite CISPI

Au niveau d'abstraction logico-fonctionnel où se situent nos travaux, notre scénario de co-spécification SYSTEME exécutable vise à spécifier et valider un modèle global (M_H) de notre SYSTEME d'intérêt satisfaisant des exigences (R_S) de notre environnement d'étude (K_{IO}).

Notamment, nous avons tenu compte d'expertises métier de notre environnement de travail pour que la plate-forme CISPI reflète au mieux certaines situations de conduite. En l'absence de consensus sur ce que pourrait être une échelle SRL (« System Readiness Level ») (Sauser, Ramirez-Marquez, Verma, & Gove, 2006) d'évaluation d'un niveau de maturité « SYSTEME », nous pouvons ainsi considérer que notre environnement de preuve de concept couvre en partie les niveaux 1 à 4 d'une échelle conventionnelle TRL (« Technologie Readiness Level ») de niveau de maturité d'une technologie d'ingénierie système basée sur des modèles.

Ce scénario de co-spécification illustre la construction itérative d'un modèle holistique (M_H) du SYSTEME d'intérêt, vérifié et validé, par co-exécution de modèles, avec des domaines d'ingénieries spécialistes fournissant chacun des modèles partiels (M_h) du SYSTEME. Nous le présentons selon deux grandes étapes.

Une première étape vise à spécifier et valider à un niveau logico-fonctionnel un modèle (M_H) de la plate-forme CISPI, en interaction avec un modèle de l'environnement (K_{IO}), par construction itérative d'une causalité circulaire entre les parties physique (M_{hP}) et logique (M_{hL}) du SYSTEME d'intérêt (Figure 62).

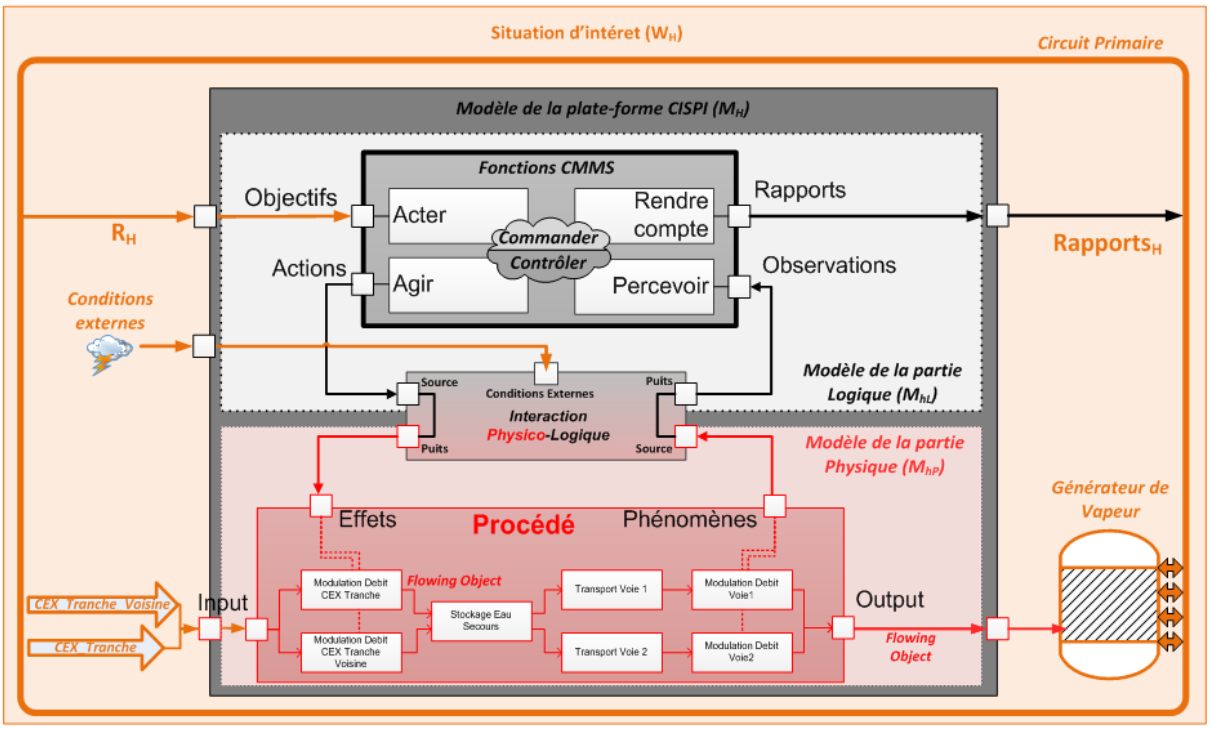


Figure 62 : Patron Architectural CMMS logico-fonctionnel visé

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

Une seconde étape vise à spécifier et valider à un niveau logico-fonctionnel un modèle (***M_{hAMI}***) d'un sous-système d'actionnement et de mesure intelligent de la plate-forme CISPI, dans son environnement opérationnel (***K_{IO}***), qui préserve la causalité circulaire précédente mais prend en compte le changement de nature physico-logique de l'interaction AMI (Figure 63).

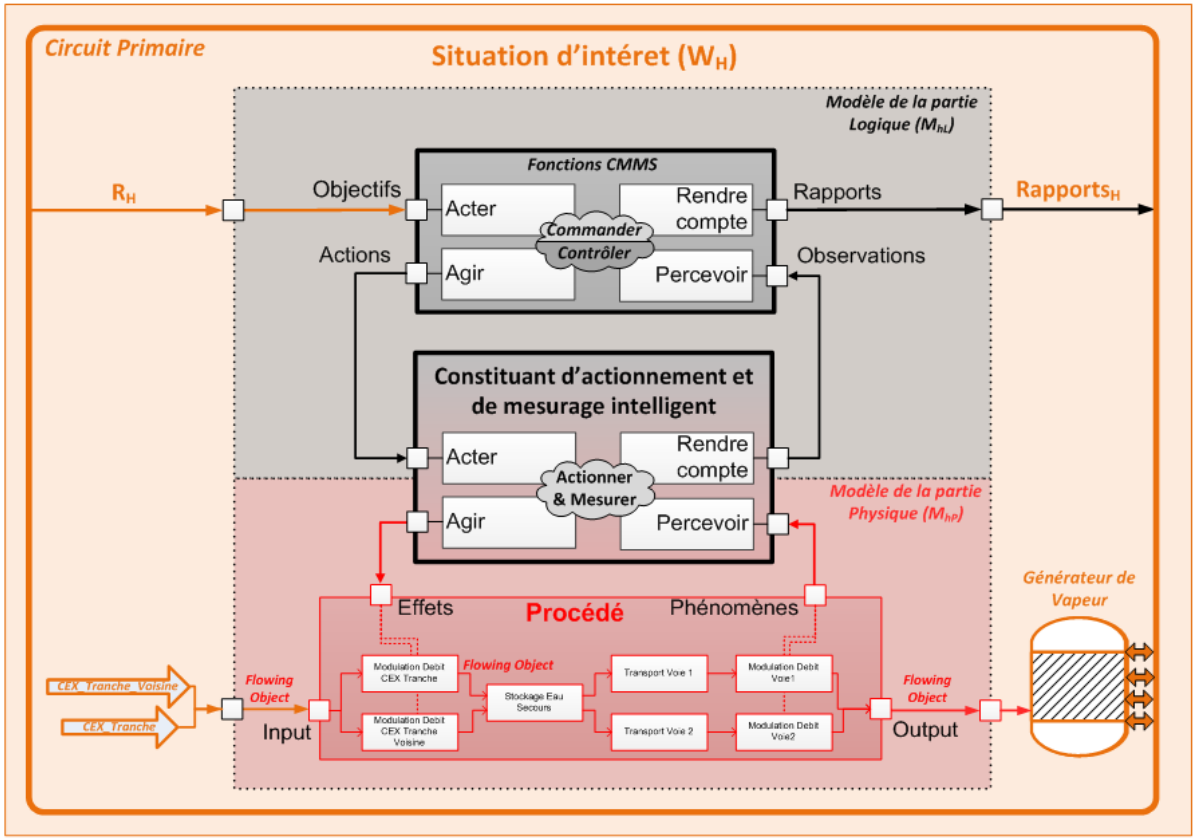


Figure 63 : Patron architectural CMMS-IAMS logico-fonctionnel visé

Nous percevons les fonctions IAMS comme des opérateurs de nature (Le Gallou & Bouchon-Meunier, 1992), définissant l'interface entre le monde de la logique (représenté par une partie « *Software* » : l'intelligence (***I***)) et le monde de la physique (représenté par une partie « *Hardware* » : les actionneurs (***A***) et mesureurs (***M***)). Nos travaux se sont intéressés à la spécification de la partie logique des fonctions IAMS, la partie physique étant prescrite par défaut (actionneurs et mesureurs de la plate-forme CISPI existante).

De manière récursive, ce changement de nature est spécifié et validé à un niveau logique par deux modèles (***M_{hIA}***) et (***M_{hIM}***) de constituants d'actionnement et de mesurage intelligent qui préserve la causalité circulaire primitive mais prend en compte une répartition homme-machine des fonctions de conduite (Figure 64).

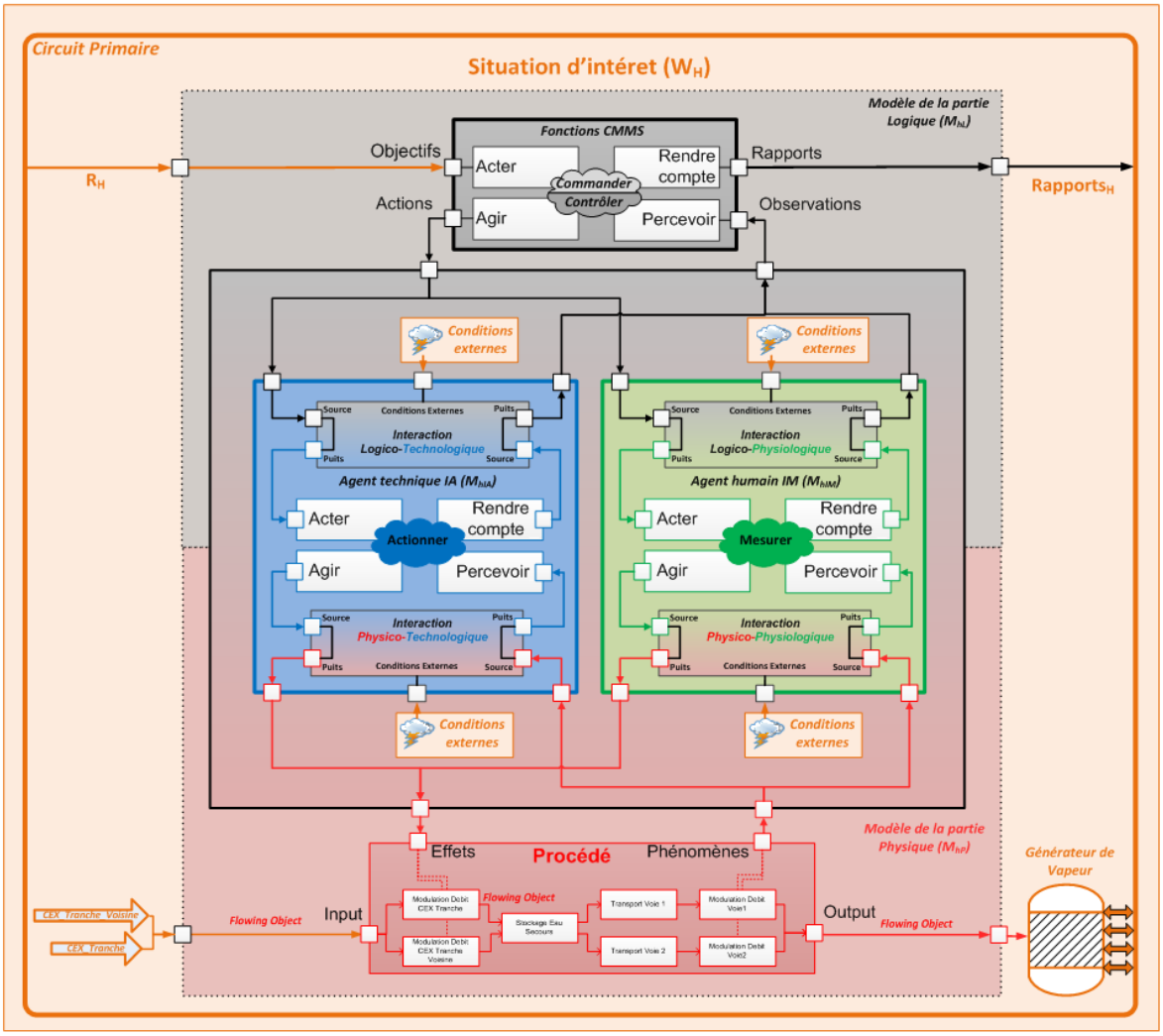


Figure 64 : Patron architectural CMMS-IAMS logique visé

La construction du modèle global (***M_H***) de la plate-forme CISPI repose sur un ensemble architecturé de modèles (***M_h***) des parties du SYSTEME. Ces modèles sont le résultat d'un processus de co-spécification itératif et récursif entre des domaines d'ingénieries spécialistes différents. En ce sens, nous développons dans les paragraphes suivants notre scénario de co-spécification SYSTEME en formalisant l'ensemble des relations de spécifications entre ces domaines d'ingénieries spécialistes.

4.4.1. Co-Spécification exécutable d'un modèle partiel d'un constituant physique

Considérons le domaine opérationnel représenté par le service plate-forme du CRAN qui occupe deux rôles en tant que domaine d'exploitation des plate-formes du laboratoire, mais aussi en tant que domaine d'ingénierie support durant la conception de nouvelles plate-formes expérimentales. Dans ce domaine opérationnel, nous délimitons notre **environnement** opérationnel (***K_IO***) (figure 12) à la salle où se situe la plate-forme CISPI du CRAN. D'un point de vue ingénierie, les exigences de conduite en exploitation (***R_H***) sont exprimées dans un premier temps en termes de phénomènes désirés par l'environnement sous la forme d'un problème opérationnel (***P_DO***).

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

Notre scénario de co-spécification débute dans ce contexte, le **problème** (P_{DO}) ne pouvant pas être résolu par le domaine opérationnel, celui-ci est décrit par notification à travers le bus de co–spécification à **l'espace-problème** d'ingénierie opérationnelle. Ce dernier est en charge de recueillir les problèmes liés à l'exploitation et de les formaliser sous la forme d'un ensemble d'**exigences** (R_{IO_SysML}) sur la base de ces **connaissances** opérationnelles (K_{IO}) (Figure 65) selon :

$$K_{IO}, P_{DO} \rightarrow R_{IO_SysML} \tag{1}$$

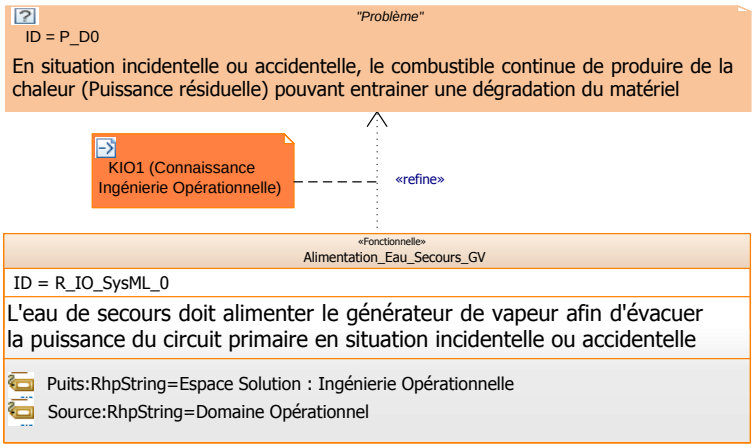


Figure 65 : Formalisation du problème (P_S) sous la forme d'exigence (R_{S_SysML}) selon la connaissance (K_S) du domaine d'ingénierie opérationnelle

Il apparait que le SYSTEME à spécifier est un système complexe, nécessitant un environnement de co-spécification SYSTEME interdisciplinaire. L'**espace-problème** d'ingénierie opérationnelle n'ayant pas les connaissances pour spécifier ce type de système, décrit par notification à l'**espace-solution** d'ingénierie système l'ensemble des exigences (R_{S_SysML}) (Figure 66). En se basant sur ses **connaissances** systémiques (K_H) notamment sur le conceptagon, l'ingénieur système étudie le système d'intérêt selon différents points de vue. La première triade explorée par l'**espace-solution** d'ingénierie système est la triade « ***Transformation, Input, Output*** » et requière la connaissance du domaine d'ingénierie du procédé.

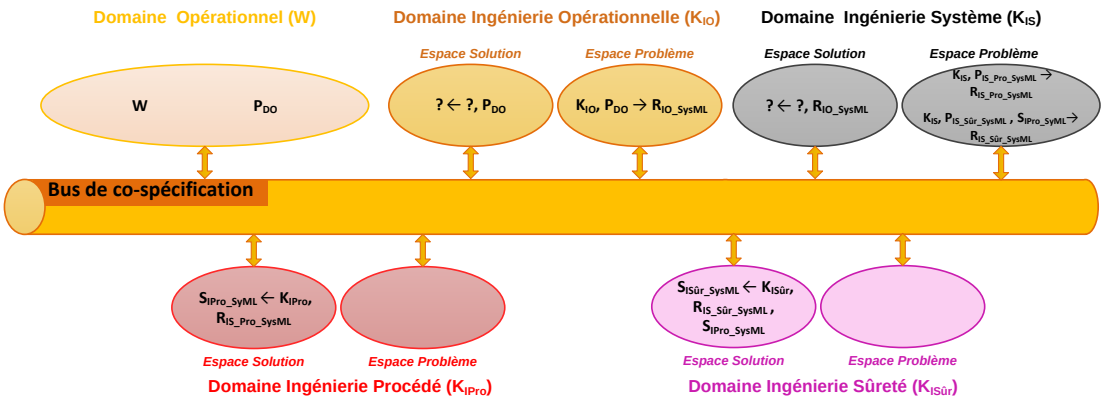


Figure 66 : Environnement de co-spécification du procédé de la plate-forme CISPI selon le patron architectural de la partie physique (Figure 64)

Ce choix d'une première triade s'appuie sur une connaissance experte qui vise à repérer le « flowing-object » traversant le SYSTEME d'intérêt, d'une entrée « source » à une sortie « puits » de l'environnement et faisant émerger les phénomènes requis au sein de cet environnement. Ce flux traversant va alors stimuler l'ensemble des interactions du SYSTEME d'intérêt. Ce faisant, au fur et à mesure du raffinement, les spécifications des parties « intérieure », « extérieure » et frontière du SYSTEME vont se peupler. Ces trois parties sont très importantes car, si certaines interactions contrôlables sont spécifiables en « partie intérieure » et d'autres non contrôlables en « partie extérieure », c'est la possibilité de relier causalement ces interactions en « partie frontière » qui permet ultérieurement de les considérer dans l'une ou l'autre partie. Ceci montre en quoi ($\boldsymbol{K_{IS}}$) se compose de connaissances systémiques à appliquer de façon systématique comme le « conceptagon », mais aussi d'autres connaissances qui font de n'importe quelle ingénierie un art qui rend très sensible aux conditions initiales l'émergence d'une solution globale. Ce constat justifie notre choix de faciliter l'expression experte (créatrice) de cette connaissance pour toute ingénierie, tout au moins dans cette construction préliminaire d'un modèle de connaissance d'ensemble en amont des phases de conception plus standardisée.

En ce sens, l'**espace-solution** d'ingénierie système décrit un nouveau problème ($\boldsymbol{P_{IS_Pro_SysML}}$), relatif à la spécification du procédé de la plate-forme CISPI, à l'**espace-problème** d'ingénierie système en charge de le formaliser sous la forme d'exigences ($\boldsymbol{R_{IS_Pro_SysML}}$) selon :

$$K_{IS}, P_{IS_Pro_SysML} \rightarrow R_{IS_Pro_SysML} \tag{2}$$

Ces exigences ($\boldsymbol{R_{IS_Pro_SysML}}$) sont ensuite décrites par notification à l'**espace-solution** d'ingénierie du procédé responsable de la spécification du procédé CISPI ($\boldsymbol{S_{IPro_SysML}}$) à partir de ses **connaissances** ($\boldsymbol{K_{IPro}}$) selon :

$$S_{IPro_SysML} \leftarrow K_{IPro}, R_{IS_Pro_SysML} \tag{3}$$

Une fois la spécification ($\boldsymbol{S_{IPro_SysML}}$) vérifiée par l'**espace-solution** d'ingénierie du procédé, cette dernière est prescrite à l'**espace-problème** d'ingénierie système pour validation. Le SYSTEME d'intérêt étant un système critique, les premières exigences à satisfaire sont des exigences de sûreté. Le domaine d'ingénierie système n'ayant pas les compétences pour valider formellement la satisfaction de la spécification du procédé vis-à-vis des exigences de sûreté, se pose un nouveau problème ($\boldsymbol{P_{IS_Sûr_SysML}}$). Ce problème est formalisé par **l'espace-problème** d'ingénierie système ($\boldsymbol{K_{IS}}$) sous la forme d'exigence ($\boldsymbol{R_{IS_Sûr_SysML}}$).

$$K_{IS}, P_{IS_Sûr_SysML}, S_{IPro_SysML} \rightarrow R_{IS_Sûr_SysML} \tag{4}$$

Ces exigences ($\boldsymbol{R_{IS_Sûr_SysML}}$)**,** ainsi que la spécification du procédé ($\boldsymbol{S_{IPro_SysML}}$) sont ensuite décrites par notification à l'**espace-solution** d'ingénierie sûreté, pour s'assurer que la spécification du procédé satisfasse les exigences de sûreté. L'espace-solution d'ingénierie sûreté effectue la vérification à parti de sa **connaissance** ($\boldsymbol{K_{ISûr}}$) puis prescrit le résultat ($\boldsymbol{S_{ISûr_SsML}}$) à l'espace-problème d'ingénierie système selon :

$$S_{ISûr_SysML} \leftarrow K_{ISûr}, R_{IS_Sûr_SysML}, S_{IPro_SysML} \tag{5}$$

En se basant sur la spécification de l'**espace-solution** d'ingénierie de sûreté ($\boldsymbol{S_{ISûr_SsML}}$), l'espace-problème d'ingénierie système valide la spécification du procédé ($\boldsymbol{S_{IPro_SysML}}$) par co-simulation pour s'assurer que les phénomènes attendus se manifestent dans le modèle de l'environnement opérationnel ($\boldsymbol{K_{IO_SysML}}$) selon :

$$K_{IO_SysML}, \{S_{IPro_SysML}, S_{ISûr_SysML}\} \vdash R_{IS_Pro_SysML} \tag{6}$$

4.4.2. Co-spécification exécutable d'un modèle partiel d'un constituant logique

De façon concourante, **l'espace-solution** d'ingénierie système explore les autres triades du conceptagon, notamment en se focalisant sur la triade « **Communication**, **Contrôle, Commande** » (Figure 67). Nous notons cependant que l'étude de la partie « Contrôle » sous-tend une relation d'ordre dans l'étude des triades. En effet, la partie « Contrôle » du SYSTEME d'intérêt spécifie, d'un point de vue fonctionnel, comment agir pour maintenir les propriétés du « flowing object » requises par l'environnement et spécifié lors de l'étude de la triade « **Transformation, Input, Output** ».

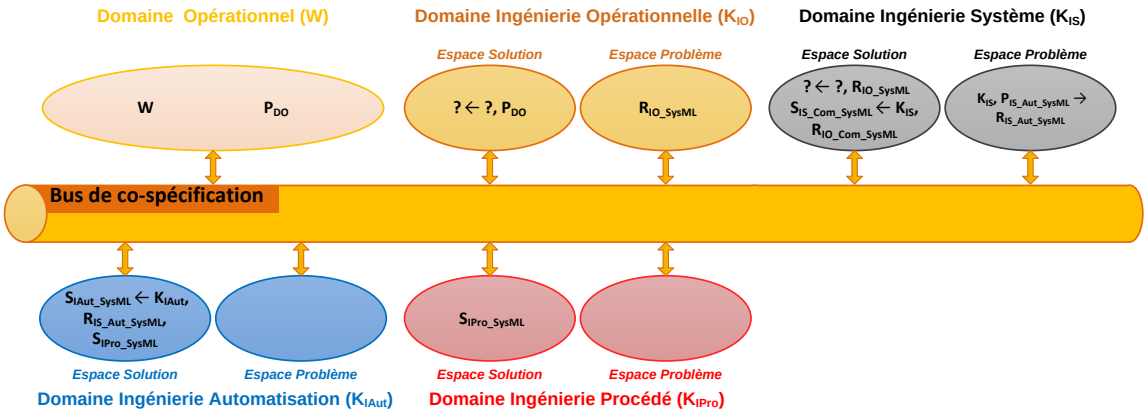


Figure 67 : Environnement de co-spécification du contrôle-commande de la plate-forme CISPI selon le patron architectural de la partie logique (Figure 64)

Ainsi, l'**espace-solution** d'ingénierie système n'ayant pas les connaissances pour spécifier les lois de contrôle de la plate-forme CISPI, se pose un nouveau problème ($\boldsymbol{P_{IS_Aut_SysML}}$) qui est formalisé sous la forme d'exigences ($\boldsymbol{R_{IS_Aut_SysML}}$) par l'espace problème d'ingénierie système ($\boldsymbol{K_{IS}}$) selon :

$$K_{IS}, P_{IS_Aut_SysML}, S_{IPro_SysML} \rightarrow R_{IS_Aut_SysML} \tag{7}$$

Le problème ainsi formalisé est décrit par notification à l'**espace-solution** d'ingénierie d'automatisation, en charge de spécifier et de prescrire la partie « **contrôle** » de la plate-forme CISPI ($\boldsymbol{S_{IAut_SysML}}$) à partir de sa **connaissance** ($\boldsymbol{K_{IAut}}$) selon :

$$S_{IAut_SysML} \leftarrow K_{IAut}, R_{IS_Aut_SysML} \tag{8}$$

Une fois la solution d'ingénierie d'automatisation ($\boldsymbol{S_{IAut_SysML}}$) prescrite, l'**espace-problème** d'ingénierie système valide la spécification par co-exécution de modèles en s'assurant que les phénomènes requis ($\boldsymbol{R_{IS_Aut_SysML}}$) se manifestent dans le modèle de l'environnement opérationnel ($\boldsymbol{K_{IO_SysML}}$).

$$K_{IO_SysML}, S_{IAut_SysML} \vdash R_{IS_Aut_SysML} \tag{9}$$

De façon concourante, **l'espace-solution** d'ingénierie système spécifie la partie « commande » ($\boldsymbol{S_{IS_Com_SysML}}$) du SYSTEME d'intérêt. Dans ce cas, le domaine d'ingénierie système que nous représentons ne nécessite pas une connaissance experte. En effet, notre formation en automatique et système à évènements discrets nous

fournit les connaissances nécessaires à la spécification de la partie « commande ». A ce stade de la co-spécification SYSTEME de la plate-forme CISPI, l'espace problème d'ingénierie système s'assure de la cohérence globale de l'ensemble des spécifications spécialistes. En ce sens, il fusionne (« merge ») l'ensemble des spécifications spécialistes pour construire un modèle du TOUT (***S_{IS_SysML}***) **(**Figure 68**)** selon :

$$S_{IS_SysML} \leftarrow K_{IS}, R_{IO_SysML}, \{S_{IPro_SysML}, S_{ISûr_SysML}, S_{IAut_SysML}, S_{IS_Com_SysML}\} \quad (10)$$

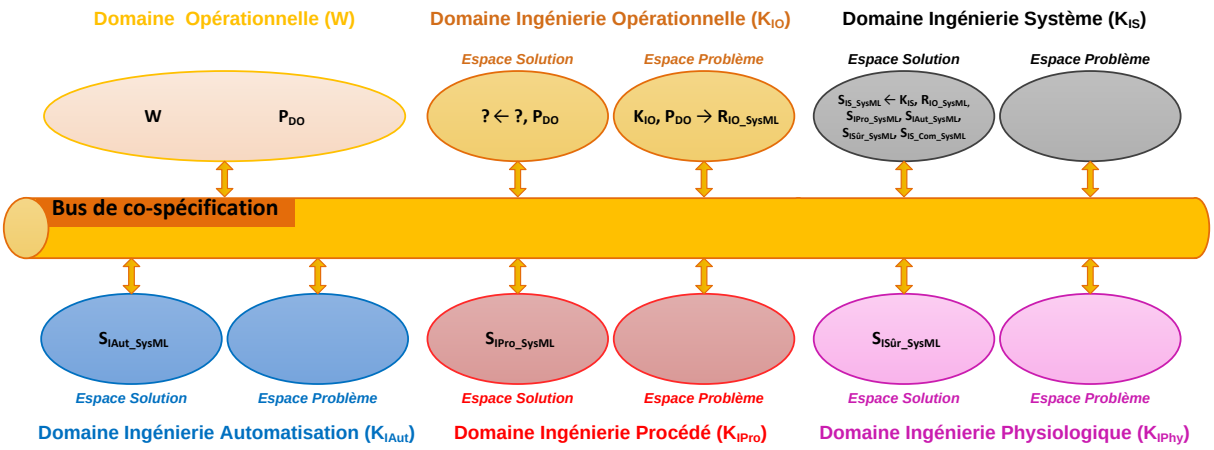


Figure 68 : Environnement de co-spécification de la plate-forme CISPI selon le patron architectural du système CMMS-IAMS visé (Figure 64)

La spécification holistique du SYSTEME (***S_{IS_SysML}***) peut ainsi être prescrite au domaine d'ingénierie opérationnelle (***K_{IO}***) pour s'assurer qu'elle satisfasse aux exigences parties prenantes (***R_{IO_SysML}***) selon :

$$K_{IO_SysML}, S_{IS_SysML} \vdash R_{IO_SysML} \quad (11)$$

La validation s'effectue par co-exécution de modèles en couplant l'ensemble des modèles des parties {***S_{IPro_SysML}***, ***S_{IAut_SysML}***, ***$S_{IS_Com_SysML}$***} avec le modèle de l'environnement opérationnel (***K_{IO_SysML}***) autour d'un bus de co-simulation. Ainsi, l'**espace-problème** d'ingénierie opérationnelle contrôle que le comportement global du SYSTEME, immergé dans son environnement, manifeste les phénomènes requis par (***R_{IO_SysML}***).

4.4.3. Co-spécification exécutable d'un modèle partiel d'un constituant technique d'actionnement

Cette étape du scénario de co-spécification s'intéresse à la co-spécification d'un sous-système d'actionnement et de mesure intelligent. Cette « intelligence » n'a de signification qu'en regard du modèle (***M_H***) en spécification et ne reflète que la part de **connaissance** (***K_h***) de chaque modélisateur représentée sous forme de modèles (***M_h***) qui sont eux même traduits ensuite sous la forme d'un algorithme, un programme et d'un point de vue humain sous la forme de procédure. En ce sens, nous spécifions de manière concourante un agent technique d'actionnement intelligent et un agent humain de mesure intelligent.

Partie 2 : Environnement de co-spécification exécutable basée sur des modèles du système de conduite CISPI

Nous nous intéressons dans un premier temps, à la co-spécification d'un agent technique d'actionnement intelligent, entre le domaine de l'ingénierie système et le domaine fournisseur d'actionnement intelligent (Figure 69) selon le patron architectural IAMS (Figure 64).

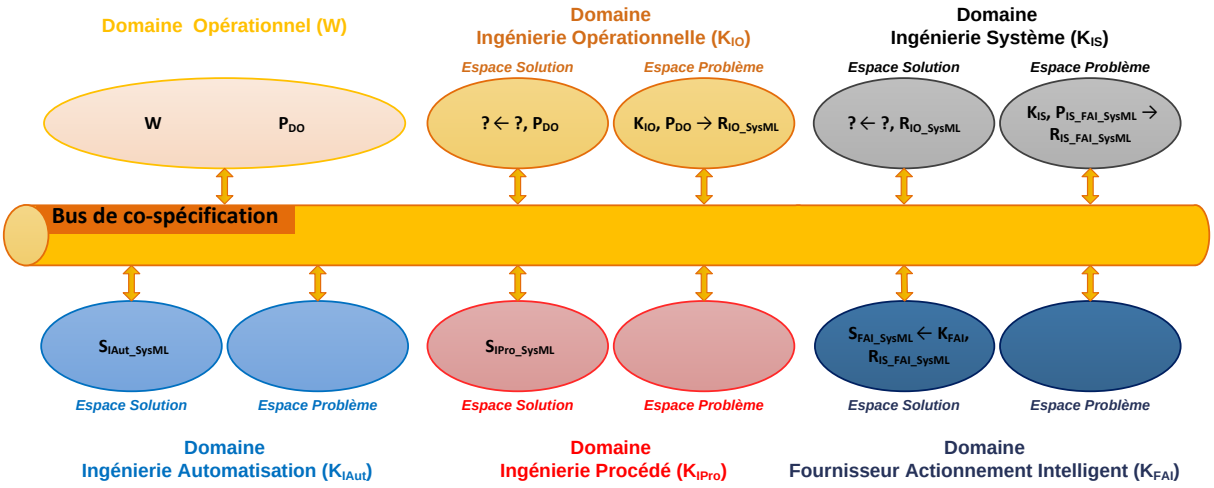


Figure 69 : Environnement de co-spécification d'une fonction d'actionnement allouée à un agent technique selon le patron architectural d'un actionneur intelligent (Figure 64)

La spécification d'un agent technique d'actionnement intelligent requière un ensemble de connaissance technique sur les sous-systèmes d'actionnement que le domaine d'ingénierie système ne possède pas. En ce sens, l'**espace-problème** d'ingénierie système formalise un nouveau problème (***$P_{IS\text{-}FAI\text{-}SysML}$***), à partir de sa **connaissance** (***K_{IS}***) sous la forme d'un ensemble d'exigences (***$R_{IS_FAI_SysML}$***) selon :

$$K_{IS}, P_{IS_FAI_SysML} \rightarrow R_{IS_FAI_SysML} \tag{12}$$

Ce problème est décrit par notification au domaine fournisseur d'actionnement intelligent (FAI), qui est représenté par le service plate-forme de notre laboratoire. Ce dernier joue alors un rôle d'assistance technique pour la mise en œuvre de notre cas d'application. En se basant sur sa **connaissance** technique (***K_{FAI}***) l'espace solution FAI prescrit une solution (***S_{FAI_SysML}***) spécifiant un agent technique d'actionnement intelligent pour répondre à l'ensemble des exigences (***$R_{IS_FAI_SysML}$***) selon :

$$S_{FAI_SysML} \leftarrow K_{FAI}, R_{IS_FAI_SysML} \tag{13}$$

Cette spécification (***S_{FAI_SysML}***) est ensuite validée par l'**espace-problème** d'ingénierie système, par co-exécution de modèle en couplant le modèle spécialiste au modèle de l'environnement, afin de s'assurer qu'elle satisfasse aux exigences (***$R_{IS_FAI_SysML}$***) selon :

$$K_{IO_SysML},\ S_{FAI_SysML} \vdash R_{IS_FAI_SysML} \tag{14}$$

Notons que la validation a été effectuée, de manière itérative, en évoluant vers un modèle de l'environnement opérationnel (***K_{IO_SysML}***) de plus en plus plausible. L'objectif était d'éprouver la spécification (***S_{IFAI_SysML}***) de l'agent technique d'actionnement intelligent selon les niveaux de maturité définis par l'échelle TRL. En ce sens,

nous avons intégré le modèle de l'agent technique d'actionnement intelligent dans un premier temps avec un modèle de l'environnement (représentant le procédé), puis effectuer la transition vers la plate-forme CISPI du laboratoire émulant le procédé physique à piloter.

4.4.4. Co-spécification exécutable d'un modèle partiel d'un constituant physiologique de mesurage

Notons que notre scénario de co-spécification se limite uniquement à la spécification d'une interaction de perception sonore entre un agent technique « source» et cet agent humain « puits » que nous étudions selon l'activité « Percevoir » (Figure 64).

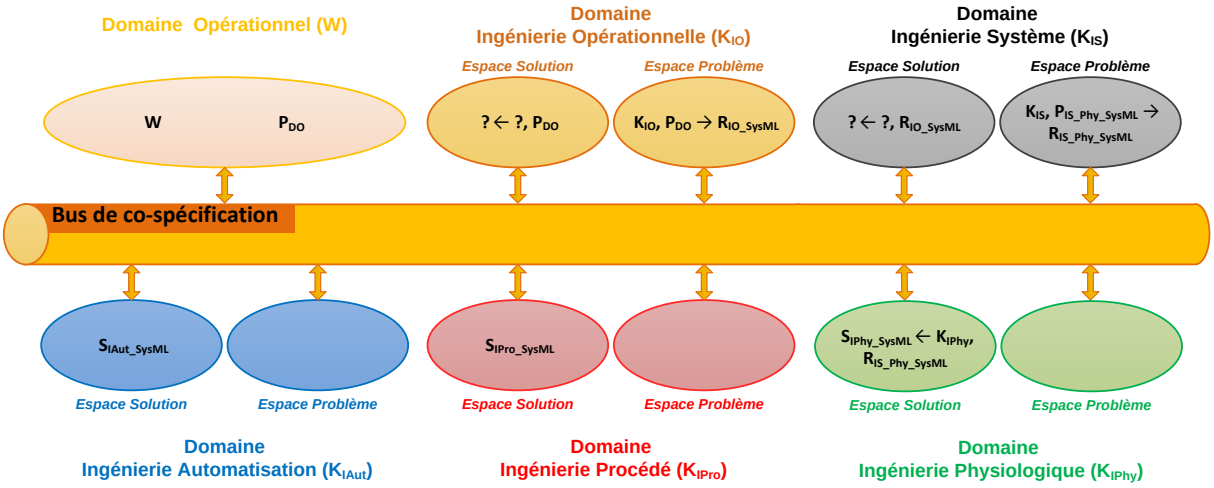


Figure 70 : Environnement de co-spécification d'une fonction de mesure allouée à un agent humain le patron architectural d'un mesureur intelligent (Figure 64)

Tout comme pour l'agent technique, le domaine d'ingénierie système requière un ensemble de connaissance pour spécifier l'interaction technico-physiologique de perception sonore. En ce sens, l'**espace-solution** d'ingénierie système décrit un nouveau problème (***$P_{IS_Phy_SysML}$***), formalisé par l'**espace-problème** sous la forme d'un ensemble d'exigences (***$R_{IS_Phy_SysML}$***) selon :

$$K_{IS}, P_{IS_Phy_SysML} \rightarrow R_{IS_Phy_SysML} \tag{15}$$

Ces exigences sont ensuite décrites par notification à l'**espace-solution** d'ingénierie physiologique, en charge de spécifier et de prescrire l'agent humain de mesure intelligent (***S_{IPhy_SysML}***) en se basant sur ses **connaissances** spécialistes (***K_{IPhy}***) selon :

$$S_{IPhy_SysML} \leftarrow K_{IPhy}, R_{IS_Phy_SysML} \tag{16}$$

Finalement, cette spécification (***S_{IPhy_SysML}***) est validée par l'**espace-solution** d'ingénierie système par co-exécution de modèles avec un modèle de l'environnement opérationnel (***K_{IO_SysML}***) pour s'assurer qu'elle satisfasse aux exigences (***$R_{IS_Phy_SysML}$***) selon :

$$K_{IO_SysML}, S_{IPhy_SysML} \vdash R_{IS_Phy_SysML} \tag{17}$$

4.5. Conclusion : Environnement collaboratif constitutif du processus de spécification-système exécutable ciblé

L'objectif de ce chapitre était de présenter notre environnement de co-spécification SYSTEME exécutable basée sur des modèles et de l'illustrer à travers un scénario de co-spécification de la plate-forme CISPI. Comme nous venons de le voir à travers ce scénario, ce sont les ingénieurs avec leurs connaissances respectives qui délivrent les ($\boldsymbol{M_h}$) composant le modèle global ($\boldsymbol{M_H}$) de compréhension du système en co-spécification. Au-delà, ce sont ces ingénieurs parties prenantes de notre environnement de co-spécification qui doivent être architecturés en **système pour faire** « en charge d'analyser le besoin, et d'étudier et concevoir un **système à faire** répondant aux mieux au besoin, compte tenu des potentialités et contraintes de son propre environnement » (Fiorèse & Meinadier, 2012). En ce sens, notre environnement vise à supporter le comportement global de ce « système pour faire » en facilitant la requête itérative de connaissances interdisciplinaires pour spécifier le « système à faire » en partageant à l'essentiel des éléments de raisonnement, de modélisation, de langage et de travail communs tout en préservant la spécificité de chaque métier.

Notons également que cet environnement de co-spécification exécutable basée sur des modèles soulève des questions relatives à la co-simulation concernant la configuration du bus de co-simulation et des différents outils de simulation, afin de garantir le comportement global du SYSTEME en simulation. Si nous avons fait œuvre d'architecte de co-simulation dans notre cas d'étude, ce rôle d'interaction avec l'ingénieur système au facteur d'échelle industriel peut faire l'objet d'un nouveau métier (Retho, 2015) nécessitant le développement d'une heuristique de co-simulation le guidant dans son travail.

Conclusion : Co-spécification d'un modèle total constitutif du système d'intérêt ciblé

Nous avons proposé dans ces deux chapitres un environnement collaboratif de co-spécification basée sur une heuristique de modélisation système. Cette dernière permet d'orchestrer le comportement global de ce « système pour faire » par la requête itérative de connaissances spécialistes élémentaires mais aussi par la requête de connaissances sur la situation d'intérêt à contrôler. Alors que beaucoup de travaux se focalisent sur la conception d'un système d'un point de vue de son architecture technique, il nous est apparu important d'axer notre contribution sur la compréhension holistique (« systemic thinking ») du SYSTEME d'intérêt à faire. Cette dimension holistique est étudiée selon le paradigme holonique à travers le conceptagon et le patron architectural du CMMS-IAMS (Figure 71) qui applique au « système à faire » la caractéristique essentielle de tout système qu'est la causalité circulaire entre ses éléments afin de les faire fonctionner collectivement comme un ensemble en regard de rétroactions aussi bien internes qu'externes.

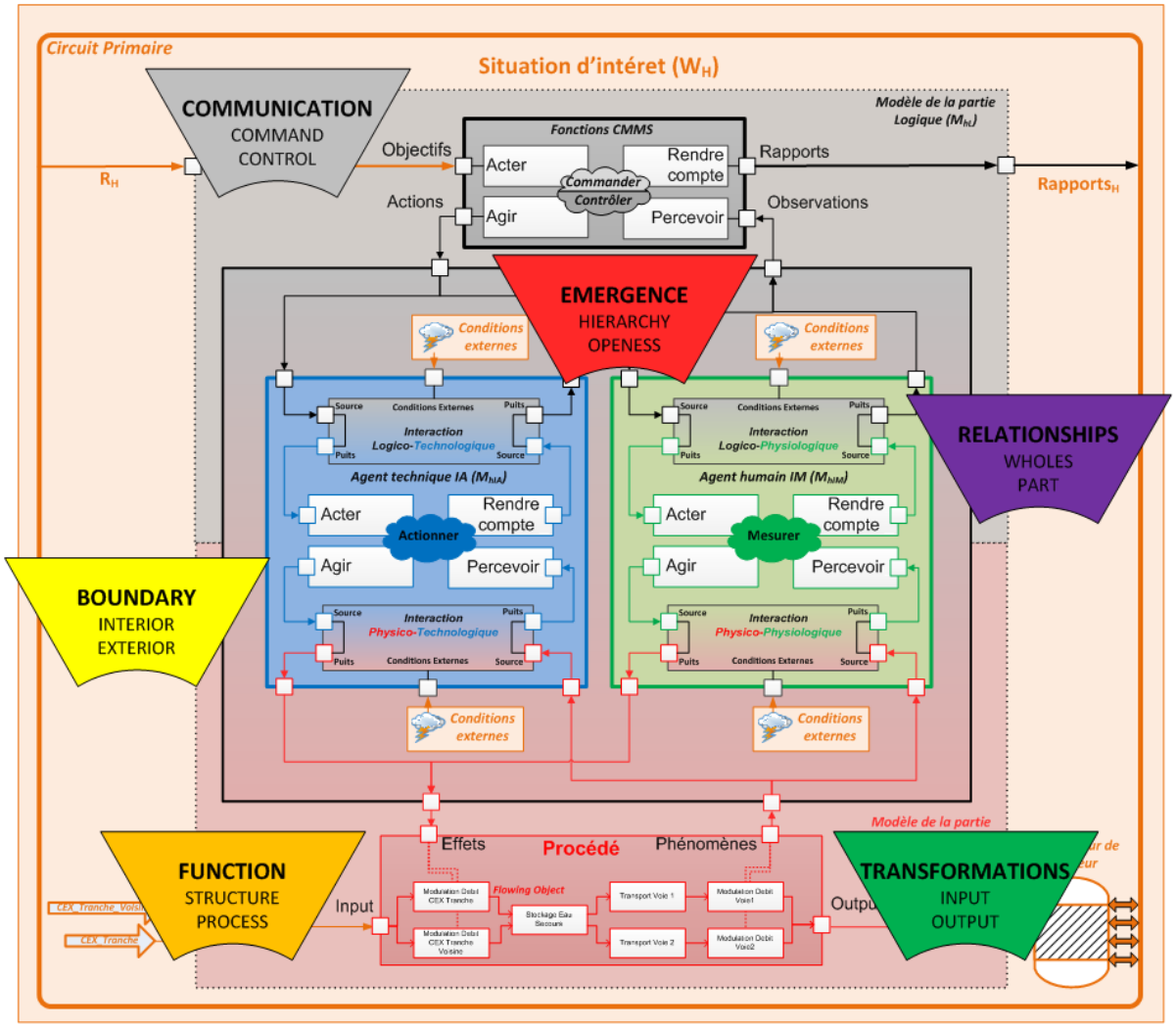


Figure 71 : Impact du patron de modélisation systémique « Conceptagon » sur le patron architectural CMMS-IAMS visé

Ce raisonnement de co-spécification système est mise en œuvre dans un environnement de co-modélisation dont l'objectif est de faciliter la construction d'un ensemble de connaissances représentées par un ensemble

de modèles spécialistes (M_h) en interaction et architecturés par l'ingénieur système en un modèle SYSTEME (M_H), afin de satisfaire les exigences parties prenantes. Cet environnement de co-modélisation profite des évolutions récentes des technologies numériques notamment des bus de co-simulation afin de vérifier/valider le comportement global du SYSTEME par co-exécution de modèles.

Dans la suite de ce mémoire, nous illustrons la mise en œuvre de notre heuristique de co-spécification SYSTEME pour la définition de la plate-forme CISPI du CRAN tel que définis par le scénario précèdent. Nous mettons en évidence l'apport de cette heuristique pour la construction itérative du modèle holistique de notre SYSTEME d'intérêt selon un ensemble de modèles des parties prescrit par les domaines d'ingénieries spécialistes (procédé, d'automatisation, sûreté, physiologique, fournisseur d'actionnement intelligent).

Partie 3 : Co-spécification exécutable de constituants de conduite du procédé du système CISPI

Introduction : Interaction collaborative entre domaines d'ingénieries interdisciplinaire centrée physique et centrée conduite

L'objectif des deux chapitres de cette troisième partie est de mettre en œuvre notre heuristique de co-spécification système pour définir, à un niveau d'abstraction fonctionnel, un modèle initial constitutif d'un comportement global de notre plate-forme d'expérimentation CISPI. Pour cela, l'ingénieur système se base sur le patron architectural CMMS (Figure 72) défini dans le chapitre 4 pour modéliser l'interaction logico-physique de conduite de la transformation du flux matière-énergie entre CISPI et un contexte environnant afin de satisfaire des requêtes externes de services.

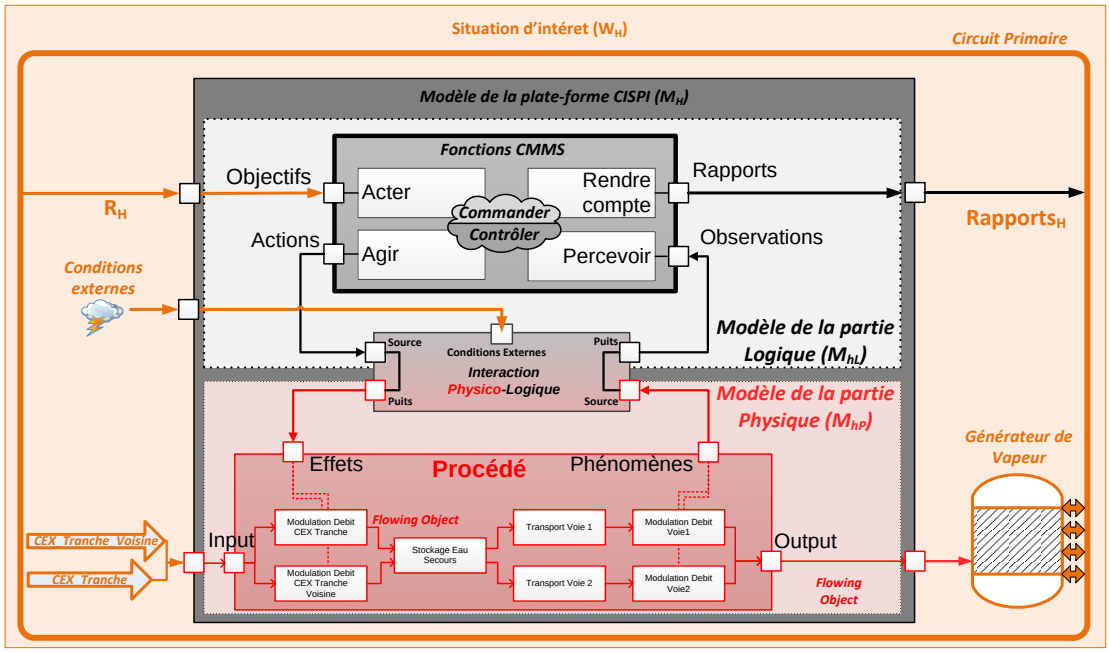


Figure 72 : Patron Architectural CMMS visé en partie 3

Nous illustrons la construction de ce modèle global en 3 étapes majeures faisant l'objet chacune d'itérations de co-spécification :

- descriptive d'un modèle d'exigences système (***R_H***) à satisfaire,
- prescriptive de modèles partiels exécutables (***M_{hP}, M_{hL}***) vérifiant des modèles respectifs d'exigences,
- prescriptive d'un modèle global exécutable (***M_H***) validant une intégration-système en plate-forme des modèles précédemment vérifiés,

entre un **espace-problème** d'ingénierie-système et plusieurs **espace-solution** spécialistes d'ingénierie du procédé et d'ingénierie d'automatisation ainsi que de sûreté de fonctionnement :

De façon plus précise, notre scénario (Figure 73) développe successivement en chapitres 5 et 6 un ensemble de co-spécifications itératives et exécutables :

- (S_{IPro_SysML}) guidée d'un point de vue système par la triade « **Function, Structure, Process** » et constitutive d'un comportement sûr de la partie physique de la plate-forme CISPI
- ($S_{IAut\text{-}Cont_SysML}$) et ($S_{IAut_Comm_SysML}$) guidées d'un point de vue système par la triade « **Communication, Control, Command** » et constitutives d'un comportement discret-continu de la partie logique de conduite de la plate-forme CISPI.

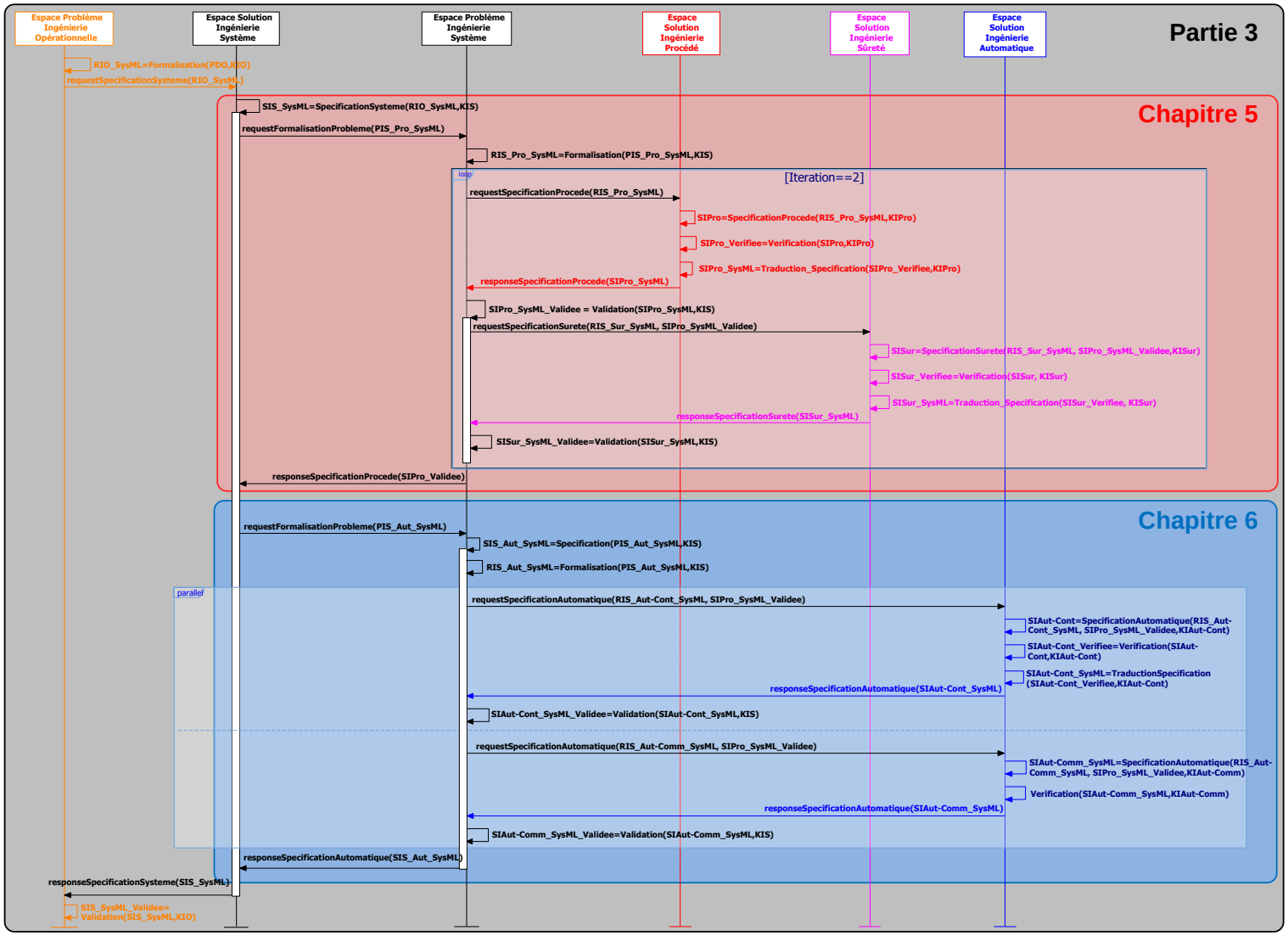


Figure 73 : Diagramme de séquence SysML descriptif du scénario de co-spécification système visé en partie 3

Partie 3 : Co-spécification exécutable de constituants de conduite du procédé du système CISPI

5. Chapitre 5 : Co-spécification exécutable d'un constituant physique du système de conduite CISPI

Dans notre scénario de co-spécification, l'**espace-solution** d'ingénierie du procédé est en charge de prescrire une **spécification** exécutable du procédé (S_{IPro_SysML}) à l'**espace-problème** d'ingénierie système en réponse à une spécification descriptive d'exigences-système ($R_{IS_Pro_SysML}$) en regard de l'interaction physique ciblée. Cette spécification partielle est ensuite validée par exécution de modèles en tant que constituant de la spécification système ($S_{IS,\ SysML}$) globale visée.

5.1. Introduction : Interaction physique ciblée du système CISPI

L'Ingénieur-système est guidé par la triade « ***Function, Structure, Process*** » en regard de la situation-système d'intérêt. La **fonction** du système d'intérêt est décrite par le procédé comme une « suite d'opérations chimiques, physiques ou biologiques nécessaires pour la transformation, le transport et le stockage de matériaux ou d'énergie » (ISO 10628, 2014) en vue d'obtenir un résultat déterminé. Les **processus** représentent les séquences d'opérations à mettre en œuvre de manière structurées (**structure**) et le plus indépendamment possible (**fonction**) des ressources opérantes pour faire circuler à travers l'espace-temps l'« **objet d'intérêt Eau** » constitutif de l'interaction ciblée. Cela conduit l'ingénieur-système à interopérer avec l'ingénieur du procédé afin de raffiner sa spécification système selon la triade « ***Transformation, Input, Output***» afin de prescrire une solution exécutable en termes de modulation (**transformation**) d'un débit d'eau entre une source (**Input**) et un puits (**output**) de stockage.

5.2. Co-spécification système d'un modèle d'interaction physique de modulation de débit d'eau

Cette co-spécification système se focalise dans un premier temps sur l'objet d'intérêt « eau » constitutif (« ***flowing object*** ») de la finalité fonctionnelle (modulation de puissance) de l'interaction physique ciblée.

5.2.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction physique de modulation de débit d'eau

Notre scénario débute au cours d'une première itération de **spécification** entre un **espace-problème** d'ingénierie opérationnelle et un **espace-solution** d'ingénierie système. Au cours de cette itération, « *il s'agit d'explorer le problème à résoudre dans tous ses aspects en analysant les besoins et contraintes des parties prenantes intéressées à l'utilisation et à l'exploitation, tout en anticipant sur les contraintes des autres parties prenantes susceptibles d'intervenir dans le cycle de vie du système et tout particulièrement durant sa conception* » (Fiorèse & Meinadier, 2012). Dans notre cas d'étude, les parties prenantes intéressées sont représentées par le domaine d'ingénierie opérationnelle, dont les besoins ont été formalisés sous la forme d'exigences parties prenantes (R_{IO_SysML}) (Cf. Annexe 5-1).

A cette étape du processus de co-spécification, l'objectif de l'**espace-solution** d'ingénierie système est d'établir un ensemble cohérent et complet d'exigences systèmes non ambiguës et vérifiables, satisfaisant aux exigences des parties prenantes. En ce sens, l'ingénieur système se base sur différentes **connaissances** provenant de domaines spécialistes pour progresser dans le processus de spécification système et raffiner les exigences (Figure 74).

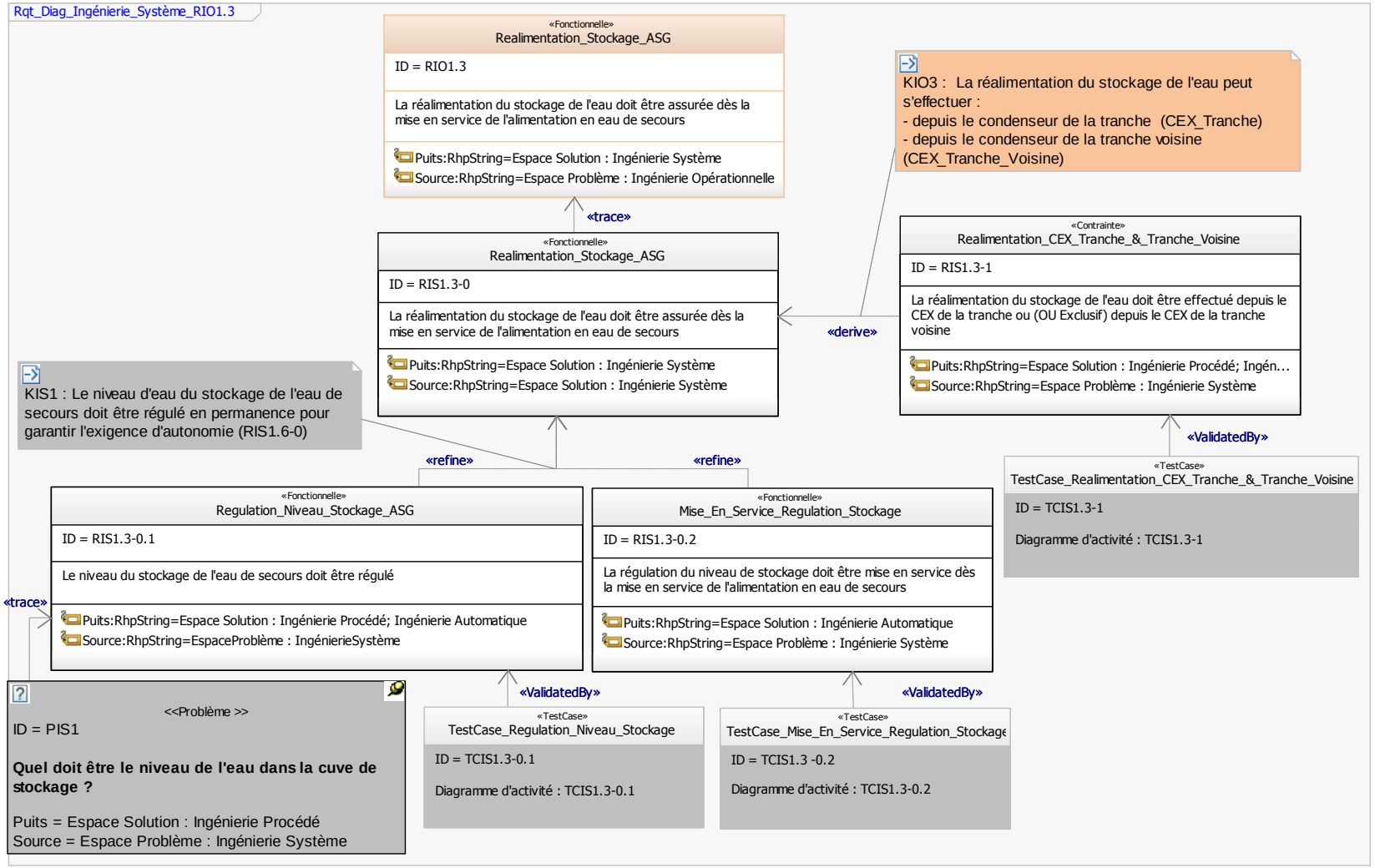


Figure 74 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence opérationnelle ($R_{IO\ 1.3}$) par l'espace-solution d'ingénierie système

Ce processus d'analyse des exigences est illustré (Figure 74) avec l'exigence ($R_{IO1.3}$) du domaine d'ingénierie opérationnelle représentée dans un diagramme d'exigences SysML. Ainsi, le lien « *trace* » représente la relation de **description** depuis l'**espace-problème** d'ingénierie opérationnelle vers l'**espace-solution** d'ingénierie système. Le texte de l'exigence n'est pas modifié, mais son identifiant ($R_{IO1.3}$) change pour montrer que l'ingénieur système s'est approprié cette exigence dans son domaine ($R_{IS1.3\text{-}0}$). En se basant sur un ensemble de **connaissances** provenant soit de son **espace-solution** (K_{IS1}), soit d'autres **espaces-solutions** spécialistes (K_{IO3}), l'ingénieur système analyse l'exigence ($R_{IS1.3\text{-}0}$) pour la raffiner ($R_{IS1.3\text{-}0.1}$) et en dériver une nouvelle ($R_{IS1.3\text{-}1}$). Ces **connaissances** sont représentées par des notes graphiques SysML de type « *Rationale* » apportant des justifications aux transformations d'exigences (raffinement, dérivation, composition,…).

Dans le cas où l'ingénieur système ne possède pas la **connaissance** pour progresser dans le raffinement des exigences il se pose un nouveau problème (P_{IS1}) qui est ensuite diffusé à travers le bus de co-spécification, vers les ingénieries spécialistes dans l'attente d'une solution. Les problèmes sont représentés par des notes graphiques SysML de type « *Problem* » exprimant des problèmes à résoudre. Les « *tags* » : « *Puits* » et « *Source* » représentent respectivement l'espace d'origine de l'exigence (ou du problème) et l'espace auquel elle (il) est destinée. L'**espace-solution** d'ingénierie système effectue cette analyse pour l'ensemble des exigences décrites par l'**espace-problème** d'ingénierie opérationnelle (Cf. Annexe 5-2).

Ainsi au cours de cette première itération, l'ingénieur système définit un ensemble d'exigences et de problèmes à résoudre, destinés aux différentes ingénieries spécialistes impliquées dans le projet. Dans ce chapitre, nous nous focalisons sur l'interaction avec l'**espace-solution** d'ingénierie du procédé devant spécifier un modèle exécutable du procédé satisfaisant à l'ensemble des exigences ($R_{IS_Pro_SysML}$) décrit par l'**espace-problème** d'ingénierie système (Cf. Annexe 5-3).

5.2.2. Spécification prescriptive d'un modèle comportemental constitutif d'une interaction physique de modulation de débit d'eau

5.2.2.1. Eléments de modélisation d'une interaction physique

Les pratiques courantes en ingénierie du procédé consistent à représenter le procédé graphiquement en respectant des standards tels que la norme (ISO 10628, 2014) afin de faciliter la communication entre les ingénieries, fournisseurs d'équipements et utilisateurs. En ce sens, la norme ISO 10628 structure le procédé selon deux niveaux :

- Le niveau général permet de représenter fonctionnellement l'installation sous la forme d'un schéma d'opérations unitaires : des rectangles contenant des libellés textuels reliés par des flèches qui définissent le sens des échanges de matières et d'énergies entre les opérations unitaires (Figure 75). Ce modèle représente d'un point de vue statique le procédé sous forme de transformations de flux entrants en flux sortants selon la triade « ***Transformations, Inputs, Output*** ».

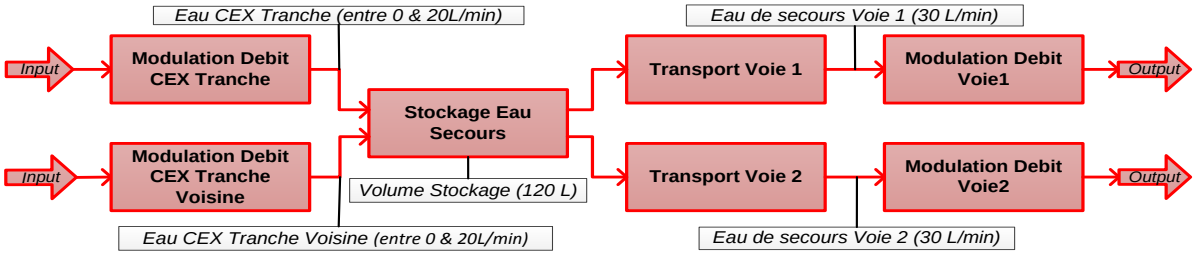


Figure 75 : Modèle orienté flux du procédé CISPI (S_{IPro}) sous forme de schéma graphique standardisé à un niveau général

- Le niveau de détail, permet de représenter organiquement chaque opération unitaire par :
  - son « Plan Circulation des Fluides» (PCF) représentant les principaux équipements ainsi que les flux de matière/énergie (Figure 76a).
  - son «Plan de Tuyauterie et d'Instrumentation » (PTI), dérivé du PCF et représentant la réalisation technique du procédé à travers les équipements, les tuyauteries mais également les fonctions de mesure et de contrôle des procédés (Figure 76b).

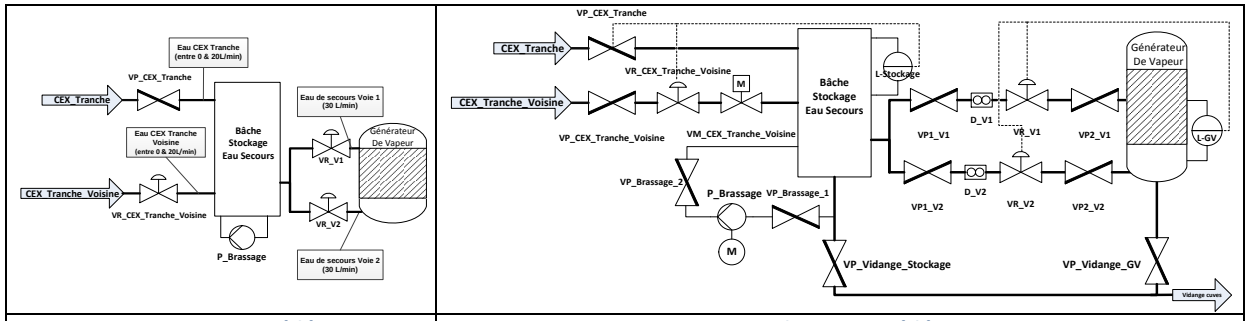


Figure 76a : PCF procédé CISPI	Figure 76b : PTI Procédé CISPI

Figure 76 : Modèle de procédé orienté flux (S_{IPro}) sous forme de schéma graphique standardisé à un niveau de détail selon la norme (ISO 10628, 2014)

Une partie de cette **connaissance** essentielle à la spécification système du procédé doit pouvoir être partagée avec l'**espace-problème** d'ingénierie système, voire avec l'ensemble des ingénieries spécialistes impliquées dans le projet de co-spécification. Ainsi, la **spécification** descriptive du procédé (***S_{IPro}***) doit être transformée sous forme de diagrammes SysML en tant que langage commun entre toutes les ingénieries spécialistes et l'ingénierie système.

5.2.2.2. Définition d'un modèle comportemental de l'interaction physique ciblée

A cette étape de notre scénario de co-spécification du SYSTÈME, l'**espace-solution** d'ingénierie du procédé doit spécifier le procédé sans se soucier qui des agents techniques ou humains effectueront les opérations. En ce sens, nous nous intéressons dans ce chapitre uniquement à la **spécification** du procédé (***S_{IPro}***) à un niveau général (Figure 75), décrit comme une suite d'opérations physiques à réaliser pour atteindre une finalité.

Afin de traduire cette spécification du procédé (***S_{IPro}***) en langage SysML, le diagramme d'activité (*act*) apparait comme étant le plus approprié. En effet, comme souligné par (Friedenthal, et al., 2008) « *The activity diagram is the primary representation for modeling flow-based behavior* ». Ce diagramme à la fois fonctionnel et comportemental permet de représenter la transformation de flux d'entrée en flux de sortie sous la forme d'une suite d'opérations (« *Action* » SysML), que les flux soient de nature discrète ou continue (matière, énergie, …). De plus le diagramme d'activité offre l'avantage de représenter en un seul modèle le comportement global du procédé.

Ainsi, le procédé CISPI modélisé dans un diagramme d'activité est représenté par une activité (que l'on peut associer à la mission du procédé (***R_{IOO}***), elle-même décomposé en actions interconnectés entre elles par des flots d'objets (« *Object Flow* ») (Figure 77). Les flots d'objets connectés aux broches d'entrée ou de sortie des actions *(« Action Pin »*) représentent ce qui est produit, consommé ou transformé au cours de l'exécution de ces actions.

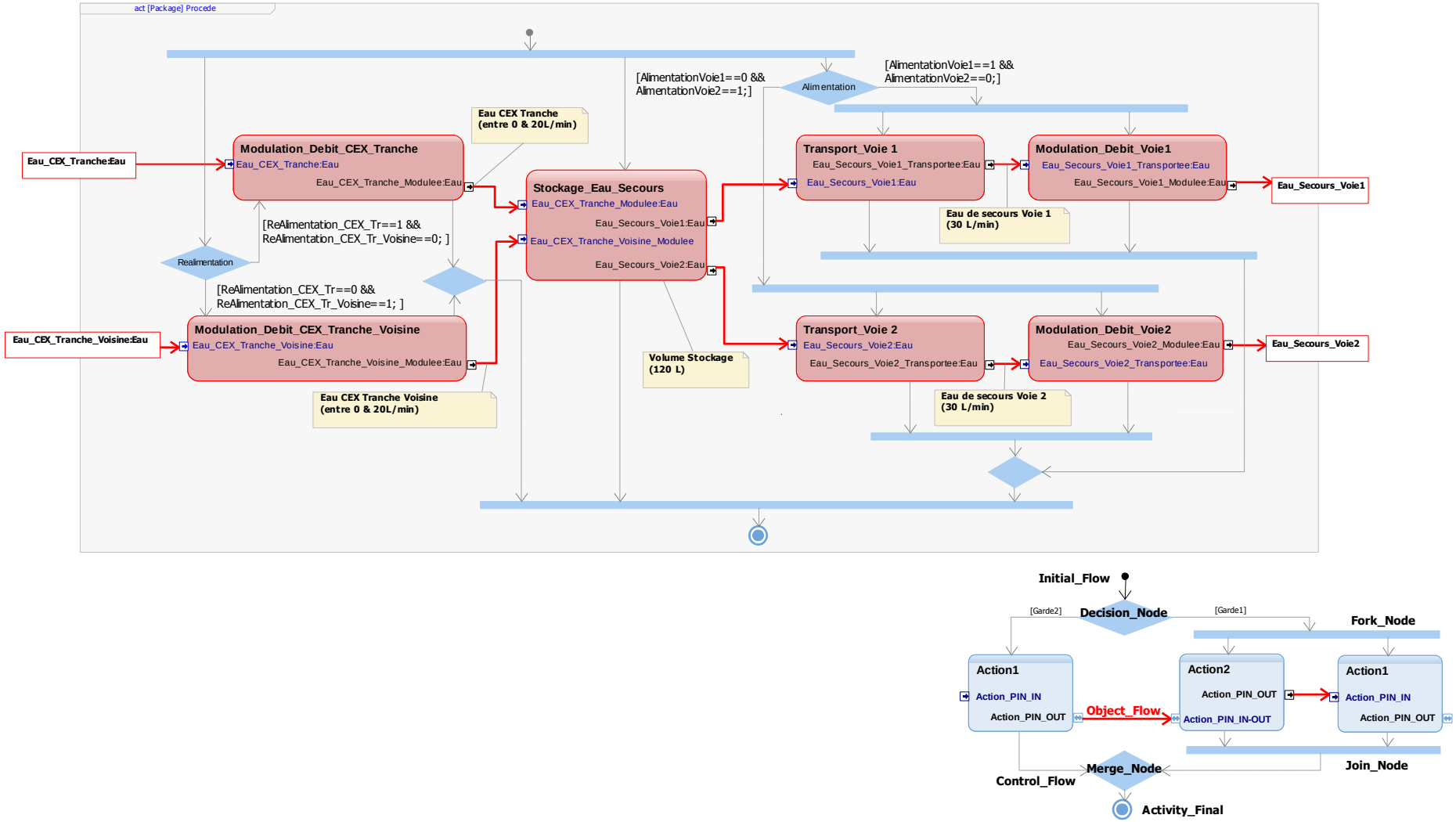


Figure 77 : Modèle orienté flux du procédé CISPI sous forme de diagramme d'activité (act) en langage SysML

Notons que contrairement aux représentations du standard ISO 10628, le diagramme d'activité SysML permet également de représenter la vision comportementale du procédé à travers des flots de contrôle *(« Control Flow »).* Par le biais des jetons empruntés au formalise des réseaux de Pétri, « *le concept de flot de contrôle permet ainsi d'exprimer des contraintes sur la séquence d'exécution des actions. Un jeton qui arrive sur un flot de contrôle permet le démarrage de l'action concernée. La terminaison d'une action met un jeton à disposition sur le flot de contrôle sortant. Lorsqu'un flot de contrôle connecte deux actions, l'action cible à l'extrémité du flot de contrôle ne peut pas démarrer tant l'action source n'est pas terminée* » (Roques, 2011).

Ainsi, des comportements complexes peuvent être définis en utilisant divers éléments du diagramme d'activité SysML :

- Les nœuds de décision (« *Decision Node* ») sont des nœuds de contrôle structuré représentant un choix dynamique en plusieurs conditions (« *Garde* ») devant être mutuellement exclusives.
- Les nœuds d'embranchements (« *Fork Node* ») sont des nœuds de contrôle structuré représentant un embranchement parallèle. L'embranchement duplique le jeton entrant sur chaque flot sortant. Les jetons sur les arcs sortants sont indépendants et concurrents.
- Les joins (« *Join Node* ») sont des nœuds de contrôle structuré représentant une synchronisation entre actions. Le join ne produit son jeton de sortie que lorsqu'un jeton est disponible sur chaque flot entrant.

En utilisant cette notation, l'ingénieur du procédé raffine le modèle du procédé orienté flux d'objet pour définir le comportement du procédé et propose ainsi un modèle de **spécification** du procédé en langage SysML (S_{IPro_SysML}) (Figure 77). Notons que cette **prescription** du comportement du procédé, se traduira d'un point de vue conduite sous la forme de procédures ou d'automatismes.

Lors de la **prescription** de cette **spécification** (S_{IPro_SysML}), l'**espace-solution** doit joindre un diagramme d'exigences (req) SysML représentant les liens de traçabilité[49] entre les éléments de son modèle et les exigences décrites par l'ingénierie système. Comme le souligne (Sannier & Baudry, 2011) « les raisons pour lesquels les exigences sont spécifiées ont un caractère souvent implicite limité à une ou quelques personnes possédant une expertise particulière, qui peuvent justifier un choix technologique, politique, économique à un moment donné ». Ainsi à l'échelle d'une durée de vie d'une centrale[50], il est vital d'expliciter les liens entre les solutions prescrites et les différents niveaux et les différentes natures d'exigences pour être à même de connaitre l'impact d'une modification d'une exigence au cours de la conception, du développement et de l'exploitation de la centrale. Nous utilisons le lien de dépendance « *satisfy* » pour exprimer le fait que la solution satisfait les exigences décrites. Cette dépendance représente le lien de **prescription** entre l'**espace-solution** d'ingénierie du procédé et l'**espace-problème** d'ingénierie système.

[49] (Gotel & Finkelstein, 1994) définissent la traçabilité comme : « *the ability to describe and follow the life of a requirement, in both a forward and backward direction (i.e from its origin, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases)* ».

[50] Durée de vie d'une centrale : plusieurs années de conception et des décennies d'exploitation (40 à 60 ans selon les centrales)

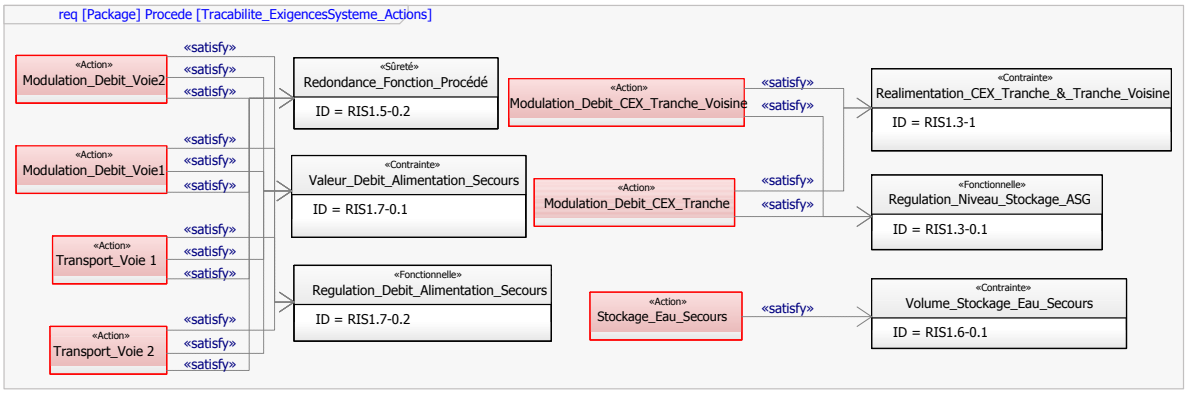


Figure 78 : Traçabilité entre la *spécification* descriptive d'ingénierie système (Exigences - R_{IS_Pro}) et la *spécification* prescriptive d'ingénierie du procédé (Actions - S_{IPro_SysML})

5.2.3. Co-intégration système d'un modèle comportemental constitutif d'une interaction physique sûre de modulation de débit d'eau

Une fois la **spécification** du procédé (S_{IPro}) vérifiée et traduite en langage SysML (S_{IPro_SysML}), celle-ci est prescrite à l'**espace-problème** d'ingénierie système pour être validée en fonction des exigences système. Pour des systèmes critiques tels que les centrales nucléaires, les premières exigences sur lesquelles se focalise l'ingénieur système sont les exigences de sûreté, où l'utilisation de méthodes formelles pour la validation est particulièrement recommandé (Pyster, et al., 2012).

Itération avec l'espace-solution d'ingénierie de sûreté (validation d'un point de vue sûreté)

Dans notre scénario de co-spécification, la spécification du procédé (S_{IPro_SysML}), doit satisfaire entre autres à l'exigence de sûreté ($R_{IS1.5\text{-}0}$) relative au critère de défaillance unique (Figure 79). L'**espace-problème** d'ingénierie système, n'ayant pas les connaissances ni le savoir-faire pour valider formellement la satisfaction de cette exigences de sûreté, fait appel au domaine d'ingénierie de sûreté pour résoudre ce problème.

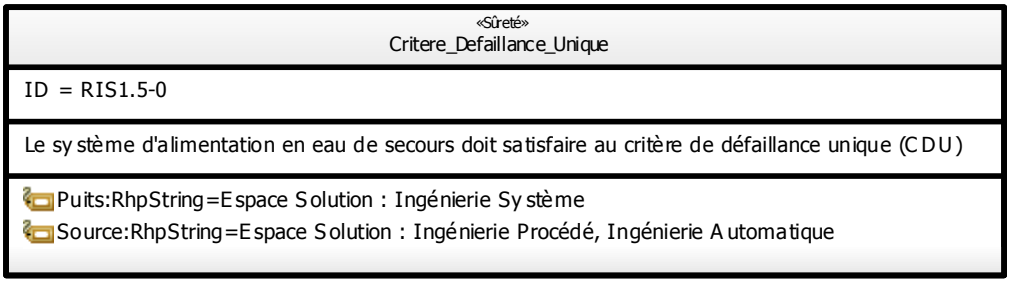


Figure 79 : Exigence $R_{IS1.5\text{-}0}$: Critère de Défaillance Unique

L'ingénieur sûreté avec lequel nous travaillons, expert en model-checking, utilise sa **connaissance** ($K_{ISûr}$) en recherche d'atteignabilité[51] et en sûreté pour valider formellement que la spécification du procédé (S_{IPro_SysML})

[51] La recherche d'atteignabilité est une méthode de vérification formelle qui détermine dans un espace d'état s'il existe un chemin qui vérifie une propriété donnée.

satisfait l'exigence ($R_{IS1.5-0}$). En ce sens, l'ingénieur sûreté reformule l'exigence ($R_{IS1.5-0}$) en tant qu'une propriété à satisfaire (Tableau 6) :

Propriété$_{Sûreté1}$	« *Si une défaillance survient sur un processus du procédé (en dehors du processus de stockage) existe-t-il une alternative pour atteindre la finalité du procédé (maintenir une capacité d'extraction de chaleur) ?* ».

Tableau 6 : Reformulation de l'exigence $R_{IS1.5-0}$ sous la forme d'une propriété de sûreté à satisfaire

Ainsi quel que soit l'état dans lequel se trouve le procédé[52], si cette propriété est satisfaite le résultat du model-checking permet de conclure que la **spécification** du procédé satisfait l'exigence ($R_{IS1.5-0}$). Notons que pour effectuer l'analyse d'un point de vue formel, les outils utilisés par le domaine d'ingénierie sûreté requièrent de traduire la spécification du procédé décrite en SysML (S_{IPro_SysML}) vers un langage formel. Ce processus de validation formelle par recherche d'atteignabilité est détaillé dans l'article (Bouffaron, Marange, & Morel, 2014). Le résultat de cette analyse, permet alors à l'**espace-solution** d'ingénierie de sûreté de prescrire à l'**espace-problème** d'ingénierie système une **spécification** ($S_{ISûr_SysML}$) statuant que la **spécification** du procédé est validée.

Cette itération entre l'**espace-problème** d'ingénierie système et l'**espace-solution** d'ingénierie de sûreté illustre l'intérêt de l'utilisation d'un langage commun tel que SysML, pour faciliter l'échange et la compréhension de modèles entre les ingénieries spécialistes impliquées (*ingénierie du procédé, ingénierie sûreté*) dans le projet.

Itération avec l'espace-solution d'ingénierie opérationnelle (validation d'un point de vue système)

Une fois la spécification du procédé validée d'un point de vue sûreté, **l'espace-problème** d'ingénierie système prend en compte le reste des exigences pour valider le procédé d'un point de vue système. La **validation** consiste alors à observer que le procédé en interaction avec son environnement (***W***), manifeste les phénomènes requis (P_{DO}) par le domaine opérationnel. En ce sens, le processus de validation requiert d'avoir un modèle de l'interaction$_{Procédé\text{-}Environnement}$, de l'environnement mais également des scénarios de test définissant les procédures à suivre durant un cas test.

Dans notre cas d'application, le modèle de l'interaction représente les fonctions de transport de l'eau entre la plate-forme CISPI et le générateur de vapeur refroidissant le circuit primaire. Au niveau fonctionnel, nous considérons ces fonctions de transport comme des fonctions identité n'affectant pas le flux d'eau transportée (Cf. Annexe 5-4). Le modèle de l'environnement, représente l'évolution de la puissance résiduelle du circuit primaire *(en situation accidentelle ou incidentelle*) en fonction du débit d'alimentation en eau de secours. Dans notre scénario de co-spécification ce modèle de l'environnement est prescrit par l'**espace-solution** d'ingénierie opérationnelle à l'**espace-problème** d'ingénierie système. Enfin, l'ensemble des scénarios de test sur lequel s'appuie l'ingénieur système sont définis durant l'analyse des exigences (Figure 74) par un ensemble de cas test « *testCase* » représentant des méthodes de validation de la satisfaction d'une ou plusieurs exigences. Ces cas test sont représentés en SysML par des sous-types d'exigences et peuvent être décris de manière textuelle ou par des diagrammes comportementaux SysML tels que des diagrammes d'états (Figure 80).

[52] Dans ce cas l'état du procédé est caractérisé, par les défaillances (ou non) des processus de transport ou de modulation

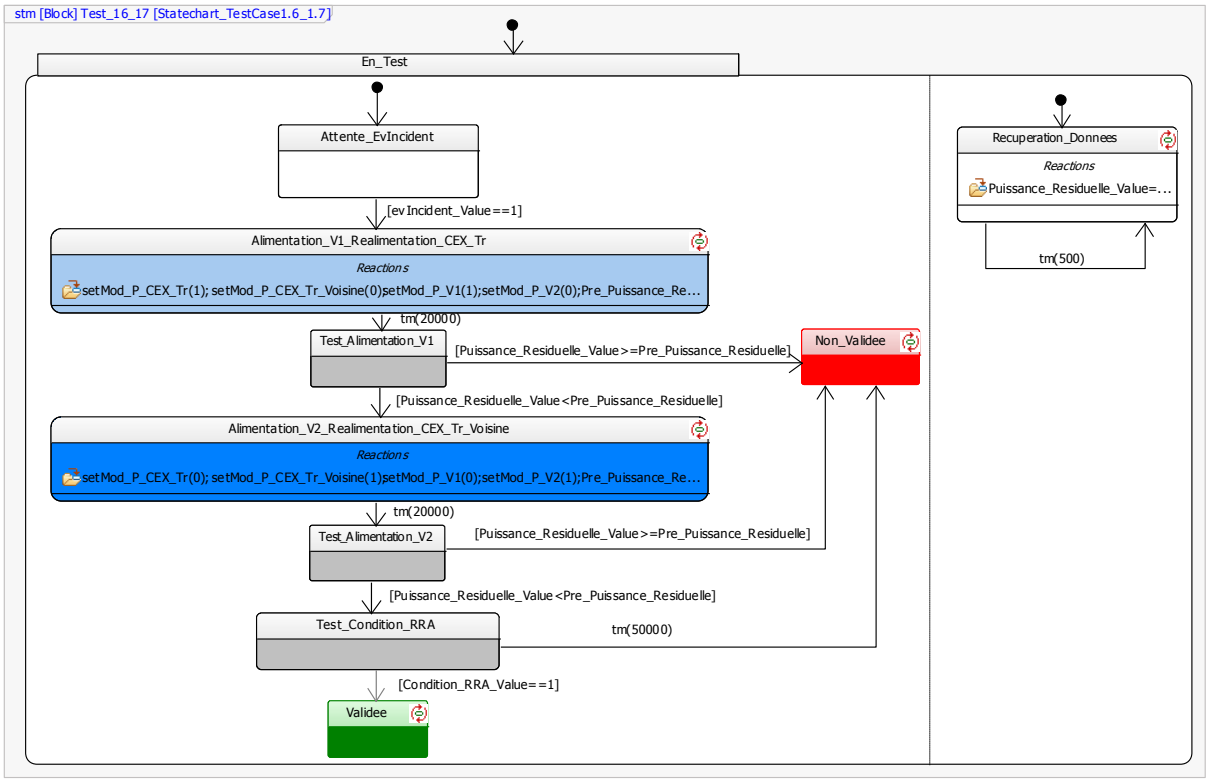


Figure 80 : Diagramme d'état (stm) SysML décrivant le « testCase » pour la validation des exigences ($R_{IS1.6}$ et $R_{IS1.7}$)

Alors que la validation du procédé aurait pu être effectuée par co-simulation (Figure 81) avec les modèles de l'interaction, de l'environnement opérationnel et des scénarios de test à exécuter, l'outil que nous avons retenu nous a limités dans l'exécution du diagramme d'activité. En effet, la sémantique des diagrammes d'activité est suffisamment précise pour envisager une exécution des modèles dans un environnement outillé (Friedenthal, et al., 2008). Cependant, l'utilisation de « Fork Node » et de « Join Node » dans le diagramme d'activité du procédé (Figure 77) ne permet pas la compilation53 avec l'outil IBM® Rational® Rhapsody®. Cette validation du comportement a donc été effectuée manuellement en suivant la sémantique d'exécution des diagrammes d'activités selon les règles d'activation et de terminaison d'une action (Roques, 2011).

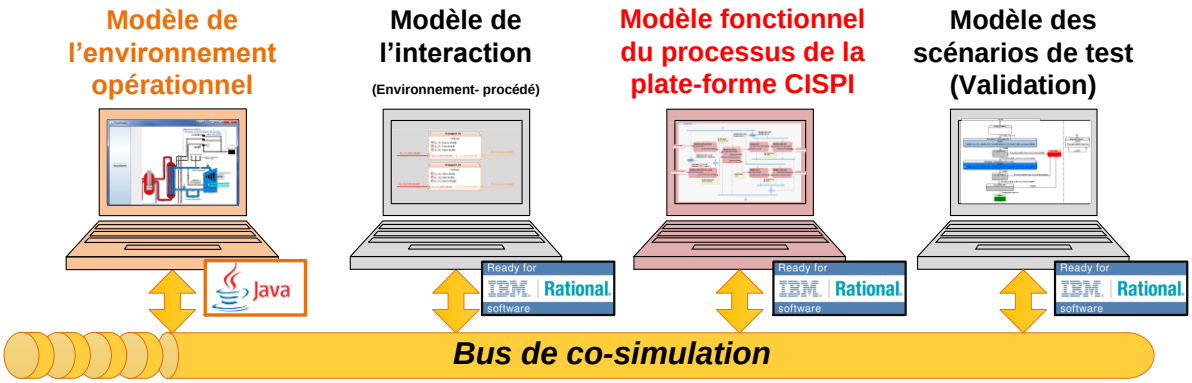


Figure 81 : Environnement de co-simulation pour la validation de la spécification du procédé (S_{IPro_SysML})

[53] Les modèles SysML sont convertis en langage C/C++ puis compilés pour obtenir un fichier exécutable utilisé durant la simulation

Problématiques liées à la validation des exigences

A cette étape du scénario de co -spécification, l'ingénieur système n'est pas en mesure de valider entièrement la **spécification** du procédé (S_{IPro_SysML}). En effet, le niveau d'abstraction définis par la norme (ISO 10628, 2014) permet de valider le comportement du procédé uniquement d'un point de vue général sans prendre en compte sa dynamique. En ce sens, le modèle prescrit n'est pas suffisamment précis pour s'assurer du respect de l'ensemble des exigences liées aux performances du procédé. Par exemple, il n'est pas possible pour l'ingénieur système de valider la satisfaction de l'exigence ($R_{IS1.7\text{-}0.1}$) (Figure 82), sans une exécution du modèle du procédé d'un point de vue physique.

«Contrainte» Valeur_Debit_Alimentation_Secours
ID = RIS1.7-0.1
Le débit d'alimentation en eau de secours doit être de 30L/min
Puits:RhpString=Espace Solution : Ingénierie Procédé; Ingénierie Auto... Source:RhpString=Espace Problème : Ingénierie Système

Figure 82 : Exigence $R_{IS1.7\text{-}0.1}$: Débit d'alimentation en eau de secours

De plus, l'analyse du comportement du procédé ne permet pas d'identifier les moyens d'actions (*variables de contrôles*) permettant d'obtenir le comportement souhaité. Enfin, nous notons que la norme (ISO 10628, 2014) est muette sur la façon de décomposer le procédé en opérations unitaires (*processus*) alors que cette étape est nécessaire afin d'obtenir des éléments observables, modifiables et d'une complexité maitrisable.

Ce constat fait émerger le besoin de proposer une méthodologie pour spécifier un modèle exécutable du procédé de façon processive (Dulmet & Lhote, 2001) à un niveau fonctionnel, selon une organisation structurée et objective de processus opérant des transformations de matière et d'énergie, sans se soucier de savoir, qui des agents techniques ou humains effectuera ces transformations. Cette réflexion se base sur une interprétation du paradigme HMS (Holonic Manufacturing System) (Valckenaers, et al., 1997) décomposant le procédé selon une arborescence logique, où les processus représentent les feuilles de l'arbre. Ces feuilles décrivent des entités tangibles de la hiérarchie d'un point de vue physique. En ce sens, cet agencement de processus doit respecter les lois de la physique. Par conséquent, une nouvelle itération avec l'**espace-solution** d'ingénierie du procédé est nécessaire pour raffiner la **spécification** du procédé (S_{IPro_SysML}).

5.3. Co-spécification système d'un modèle exécutable d'interaction multi-physiques de modulation de débit d'eau

Notons que l'approche système et SysML discrétisent couramment un ensemble de modélisations-simulations mono-physiques qui ne tiennent pas compte du couplage des phénomènes entre eux (Retho, 2015). Cette co-spécification système raffine ainsi, dans un deuxième temps, la connaissance acquise sur l'interaction ciblée afin de mieux prendre en compte les lois d'agencement multi-physiques des processus de modulation (transformation) des flux matière-énergie mis en jeu. Cette seconde itération avec l'**espace-solution** d'ingénierie du procédé est essentielle pour définir la **connaissance** (K_{IPro}) requise afin de spécifier un

modèle exécutable du procédé CISPI. Cela nous a conduit à nous intéresser à un ensemble de travaux en modélisation processive (Dulmet & Lhote, 2001) formalisant un ensemble de règles d'assemblage de processus multi-physiques, principalement sur la base des travaux de (Paynter, 1961).

5.3.1. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction multi-physiques de modulation de débit d'eau

5.3.1.1. Eléments de connaissances en modélisation processive d'une interaction multi-physiques

Processus énergétiques

La méthodologie proposée par (Feliot, 1997) distingue trois classes de processus énergétiques. Il démontre en se basant sur le tétraèdre d'état de la théorie des Bond-Graph (Figure 83) et de considérations énergétiques que ces processus constituent un base (au sens mathématique du terme) permettant de représenter tout procédé par une combinaison de ces trois éléments.

Le tétraèdre d'état (Paynter, 1961) définit un ensemble de variables et de relations généralisées, commun aux différents domaines de la physique et constituant des éléments de base pour la définition et la modélisation des processus physiques. Le concept du tétraèdre d'état est le résultat des travaux sur l'abstraction des différentes lois physiques appliquées dans des domaines différents tels que : la mécanique, l'hydraulique, l'électrique, la thermique,…

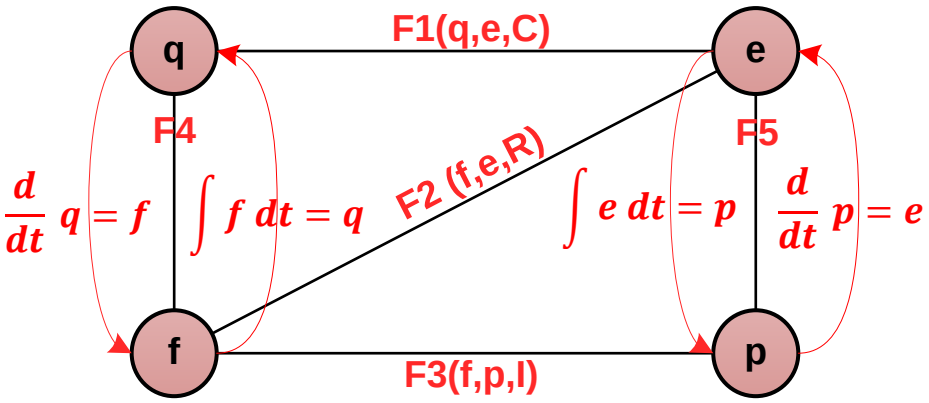


Figure 83 : Tétraèdre d'état de la théorie Bond-Graph (Paynter, 1961)

Les variables généralisées sont des variables communes à différents domaines de la physique caractérisant de nombreux phénomènes physiques et décrit usuellement par une quantité de substance : « leurs flux » ainsi que les causes ou les conséquences de ces flux : « leurs forces ». On distingue ainsi quatre types de variables généralisées :

- Les flux généralisés (**f**) : flux de masse, de volume, de charge électrique, d'entropie
- Les efforts généralisés (**e**) : forces responsables ou causes de ces flux (gradient de température, potentiel électrique, pression…)

- Les déplacements généralisés (**q**) : intégrale du flux par rapport aux temps (masse, volume, charge électrique)
- Les impulsions généralisées (**p**) : intégrale de l'effort par rapport aux temps (accumulation de flux magnétique dans une bobine, accumulation de moment angulaire dans un ressort, …)

Le tableau suivant (Tableau 7) résume les variables généralisées de flux, d'effort, de déplacement et d'impulsions pour différents domaines de la physique.

Domaine de la physique	Flux (f)	Effort (e)	Déplacement généralisé (q)	Impulsion généralisée (p)
Electricité	I : courant électrique [A]	U : potentiel électrique [V]	Q : charge électrique [C]	Φ : flux magnétique [V.s]
Mécanique de translation	ϑ : vitesse de translation [$m.s^{-1}$]	F : force [N]	X : position [m]	P : impulsion [N.s]
Mécanique de rotation	ω : vitesse de rotation [$rad.s^{-1}$]	γ : couple [N.m]	α : angle [rad]	Γ : moment angulaire [N .m.s]
Hydraulique	Qv : débit volumique [$m^3.s^{-1}$]	Pr : pression [$N.m^{-2}$]	V : volume [m^3]	Pp : moment de pression [$N.m^{-2}.s$]
Thermique	$\dot{S}$: flux d'entropie [$J.K^{-1}.s^{-1}$]	θ : Température [K]	S : entropie [$J.K^{-1}$]	

Tableau 7 : Classification des variables : flux, effort, déplacement et impulsion pour quelques domaines de la physique (Paynter, 1961)

Les relations généralisées reliant les variables généralisées_sont de deux types : les relations structurelles et les relations constitutives (Tableau 8) :

- **Les relations structurelles** : sont des relations indépendantes des notions métriques ne contenant pas de paramètres, mais uniquement des coefficients sans dimension. Typiquement, ce sont des relations d'équilibre, des lois de conservation (conservation de masse, conservation de l'énergie), loi des nœuds généralisée au flux, loi des mailles généralisée aux efforts… Les lois de conservation correspondent aux deux opérations mathématiques fondamentales : l'intégration et la dérivation par rapport au temps. Elles sont représentées par les relations F4 et F5 dans le tétraèdre de Paynter (Figure 83).
- **Les relations constitutives :** sont des relations contenant des variables relevant d'un même domaine physique (loi d'Ohm en électricité, équation entre flux thermique et variation de température, équation entre pression et volume dans une cuve en hydraulique). Ces relations sont représentées par les relations F1, F2 et F3 dans le tétraèdre de Paynter (Figure 83) :
    - **F1(q,e,C) :** relation entre déplacement, effort et un paramètre C (Capacité généralisée)
    - **F2(f,e,R) :** relation entre flux, effort et un paramètre R (Résistance généralisée)
    - **F3(f,p,I) :** relation entre flux, impulsion et un paramètre I (Inductance généralisée)

	Relations généralisées, cas linéaire				
	Relations structurelles		Relations constitutives		
Système physique	**F4** : $f = \frac{d}{dt}q$	**F5** : $e = \frac{d}{dt}p$	**F1**(q, e, C)	**F2**(f, e, R)	**F3**(f, p, I)
Electricité	$I = \frac{d}{dt}Q$	$U = \frac{d}{dt}\Phi$	$Q = c * U$	$U = R * I$	$\Phi = \mathrm{L} * \mathrm{I}$
Mécanique de translation	$\vartheta = \frac{d}{dt}X$	$\mathcal{F} = \frac{d}{dt}P$	$\mathcal{F} = K * X$	$\mathcal{F} = \alpha * \vartheta$	$P = m * \vartheta$
Hydraulique	$Q_v = \frac{d}{dt}V$	$P_r = \frac{d}{dt}P_p$	$V = \frac{A}{\rho * g} * P_r$	$Q_v * \frac{8\pi\mu l}{A^2} = P_r$	$Q_v = \frac{1}{L} * P_p$ $L = \frac{\rho}{lA} P_r$
C capacité, **R** résistance, **L** inductance, **K** raideur du ressort, $\boldsymbol{\alpha}$ coefficient de frottement, **m** masse, **A** section du tube, $\boldsymbol{\rho}$ masse volumique, **g** accélérations, $\boldsymbol{\mu}$ viscosité					

Tableau 8 : Modalités de relations généralisées du tétraèdre d'état dans différentes théories physiques (Feliot, 1997)

Comme le montre les relations de couplages (F1, F2, F3) du tétraèdre d'état (Figure 83), à une variable généralisée (e, f, p, q) correspond toujours une variable conjuguée. Ces variables se retrouvent toujours dans la nature associées par paires : (effort, flux), (effort, déplacement) ou (impulsion, flux) :

- D'un point de vue physique, un couple de variables (effort, flux) est équivalent à une puissance. En effet, le produit « effort*flux » dans un même domaine de la physique a la dimension d'une puissance.
- Les couples (déplacement, effort) et (impulsion, flux) sont respectivement équivalents à une énergie potentielle et une énergie cinétique. En effet les produits « déplacement*effort » et « impulsion * flux » dans un même domaine de la physique ont la dimension d'une énergie.

Ces couples définissent deux classes d'objets processés, dont la nature peut prendre deux formes : celle de puissance, et celle d'énergie (potentielle et/ou cinétique) constituant une typologie des différentes parties de l'environnement « E^e » (ensemble des objets énergétiques processés) selon :

$$E^e = E^{pu} \cup E^{en} \tag{18}$$

où « E^{pu}» représente les objets processés de nature puissance et «E^{en} » représente les objets de nature énergie (« E^{ec} » énergie cinétique et « E^{ep} » énergie potentielle) avec :

$$E^{en} = E^{ec} \cup E^{ep} \tag{19}$$

On induit ainsi du tétraèdre de Paynter un système de transformations (intégrale ($\int$), dérivée ($\frac{d}{dt}$) et proportionnelle (α)) sur les parties de l'environnement « E^{pu} » et « E^{en} » correspondant à une transformation d'un flux de puissance ou d'énergie en un autre type de puissance ou énergie (Figure 84).

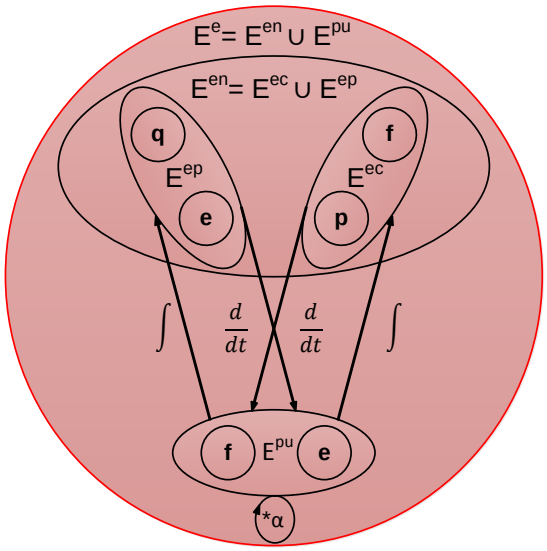


Figure 84 : Système de transformation induit par le tétraèdre d'état sur les parties de l'environnement constituées des flux d'énergies et des flux de puissances (Feliot, 1997)

Le tétraèdre induit ainsi trois classes de processus élémentaires :

- La classe des processus de transformations de puissance « E^{pu} » en énergie « E^{ec} » ou « E^{ep} » :

$$\boldsymbol{E^{pu} \xrightarrow{\int dt} E^{en}\ ;\ (e,f) \rightarrow \{(q,e),(p,f)\}} \quad (20)$$

- La classe des processus de transformations d'énergie « E^{ec} » ou « E^{ep} » en puissance « E^{pu} » :

$$\boldsymbol{E^{en} \xrightarrow{d/dt} E^{pu}\ ;\ \{(q,e),(p,f)\} \rightarrow\ (e,f)} \quad (21)$$

- La classe des processus de transformations de puissance « E^{pu} » en puissance « E^{pu} » :

$$\boldsymbol{E^{pu} \xrightarrow{\alpha:1/\alpha} E^{pu}\ ;\ (e,f) \rightarrow\ (e,f)} \quad (22)$$

Cet ensemble de classes de processus élémentaire constitue une base (au sens mathématique) de processus énergétiques de transformations de l'environnement (énergie et puissance). En effet ces processus élémentaires suffisent à former par combinaison tous les processus envisageables et cet ensemble de processus est minimal.

Patrons des processus énergétiques interprétés en processus Accumulations, Transport et Conversion

(Feliot, 1997) défend dans ses travaux qu'il existe une correspondance bijective entre les classes de processus énergétique induit du tétraèdre de Paynter et de la décomposition triadique de (Le Moigne, 1994). Le Moigne définit les processus comme un ensemble ordonné de changements affectant la position dans le temps, dans l'espace et dans leur forme, d'une famille d'objets identifiés (Figure 85).

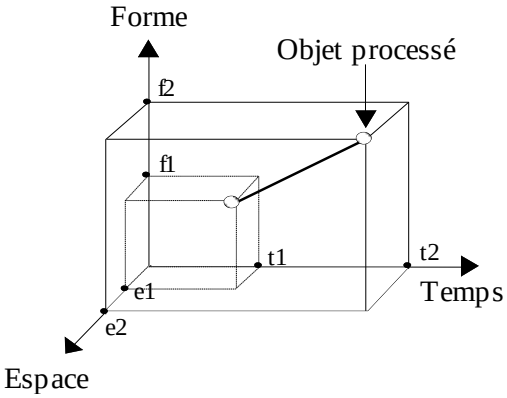


Figure 85 : Le référentiel T, E, F (Temps, Espace, Forme) permettant de repérer la position des objets soumis à des processus (Décomposition triadique des processus) (Le Moigne, 1994).

Tout processus s'intègre ainsi dans une typologie simple symbolisée par un repère à trois axes : le temps, l'espace la forme, selon une décomposition triadique :

- Un processus de type Temps (T) : consomme des puissances en entrées et produits des énergies en sorties selon la classe de processus énergétique défini par (20) (ex : Stockage de matière, d'énergie, de mémorisation d'informations)
- Un processus de type Espace (E) : consomme des énergies en entrée et produit des puissances en sortie selon la classe de processus énergétique défini par (21) (ex : Transport de matière, conduction d'énergie, transmission d'information)
- Un processus de types Forme (F) : consomme des puissances en entrée et produit des puissances en sorties selon la classe de processus énergétique défini par (22) (ex : Transformation de matière, conservation d'énergie, transformation d'information ou calcul)

En ce sens, (Feliot, 1997) propose une représentation des processus Temps, Espace et Forme d'un point de vue énergétique en s'inspirant de la sémantique des couples de variables généralisées en termes de puissance et d'énergie (Tableau 9). Ainsi les couples de puissance sont représentés par le vecteur (e; f), les couples d'énergie cinétique par le vecteur (p; f) et les couples d'énergie potentielle par le vecteur (e; q).

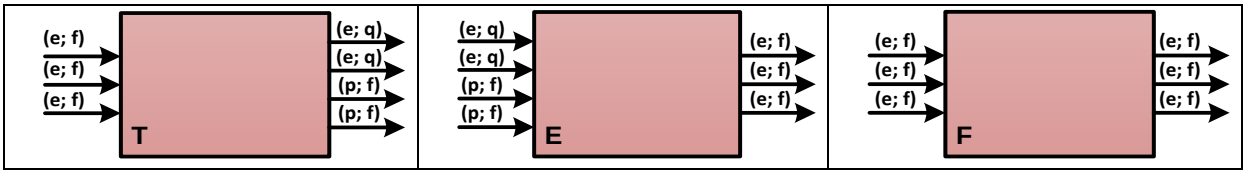


Tableau 9 : Représentation des processus énergétiques de type T, E, F (Feliot, 1997)

Nous attirons l'attention sur le fait que cette correspondance bijective, se fait sur la base d'un certains nombres d'hypothèses :

- Concernant le processus E, affectant la position spatiale de la quantité d'énergie dans le temps, le temps de propagation est négligé (ce qui n'est pas le cas des transports réels).
- Concernant le processus F, transformant une puissance en une puissance dans le temps, le temps de la transformation est également négligé (on considère que la transformation est instantanée).
- Concernant le processus T, accumulant une énergie par intégration de puissance par rapport au temps dans un espace délimité, seul le temps est pris en compte. En effet, selon (Feliot, 1997) le type des processus est définit de manière dynamique et la position spatiale du stockage n'est pas affectée au cours du temps

Si ces travaux ont pour intérêt de faire un pont, essentiel dans notre cas, entre l'approche système et la modélisation multi-physiques, les hypothèses de cette vision fonctionnelle ne prennent pas suffisamment en compte le couplage dans l'espace-temps des phénomènes multi-physiques. En négligeant le paramètre de temps, ces hypothèses ne permettent pas de refléter au plus juste la physique du procédé. D'un point de vue mathématique un processus de stockage correspond à l'intégration dans le temps d'une puissance en énergie potentielle (remplissage du stock), un processus de transport correspond à l'intégration dans le temps en énergie dissipée par frottements dans les tuyauteries. Ainsi la correspondance bijective faite par Féliot entre les processus énergétiques et les processus T, E et F est contestable. Il apparait que le processus énergétique de stockage définis selon (20) est une combinaison de processus élémentaire T et E introduit par Le Moigne, il en est de même pour le processus de transport définis selon (21). En ce sens, nous retenons la notation introduit

par (Medjaher, 2005) qui associe les processus énergétiques introduits par Feliot à des processus d'accumulation (A), de transport (T) et de conversion(C) conformément à la théorie et à la représentation de Paynter (Tableau 10).

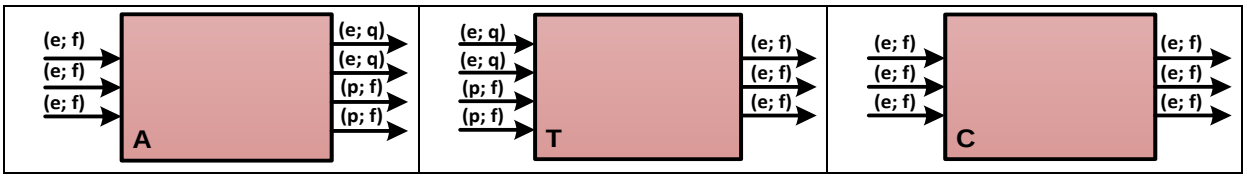


Tableau 10 : Interprétation des travaux de (Feliot, 1997) selon ceux de (Medjaher, 2005) pour la définition des processus énergétiques de type A, T, C

Règles d'agencement des processus énergétiques

L'agencement des processus A, T et C est guidé selon des règles de connexion entre processus qui respectent les lois de la physique. On distingue deux types de règles : de composition, et d'équivalence que nous listons ci-dessous.

Règles de composition :

- Deux processus ne peuvent être mis l'un à la suite de l'autre que si les sorties du premier sont du même type (Puissance ou Energie) que les entrées du second. De ce fait il apparait que :
  - à un processus A produisant des flux d'énergies en sortie, ne peuvent succéder que des processus de type T, recevant en entrée des flux d'énergie.
  - à un processus de type C, ne peuvent succéder que des processus de type C ou de type A
  - à un processus de type T, ne peuvent succéder que des processus de type C ou de type A

- Un processus de type T doit toujours être suivi d'un processus de type C. En effet, les processus de type T représentent des processus de transport d'énergie considéré comme idéal, sans dissipation, perte, conversion ou transformation d'énergie. Ce cas ne reflète pas la réalité des processus physiques, il ne faut donc pas négliger les transformations (processus de type C) qui accompagnent les processus de type T
- Une séquence de processus peut commencer par un processus d'un des trois types A, T ou C

Règles d'équivalence :

- Une séquence, de longueur quelconque, consommant en entrée des flux de puissance et produisant des flux de puissance en sortie est équivalent à un processus global de type C (Figure 86).

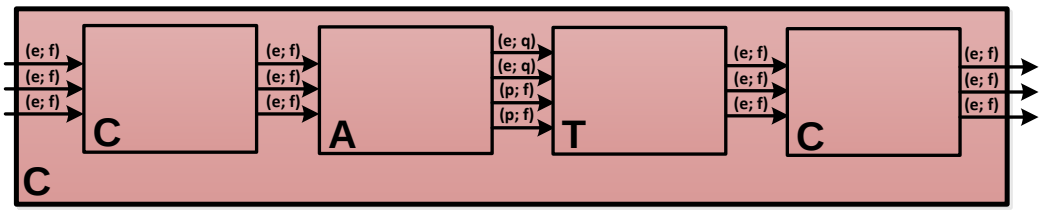


Figure 86 : Exemple d'une séquence équivalente à un processus de type F. Adaptée de (Feliot, 1997)

- Une séquence, de longueur quelconque, consommant en entrée des flux de puissance et produisant des flux d'énergie en sortie est équivalent à un processus global de type A (Figure 87).

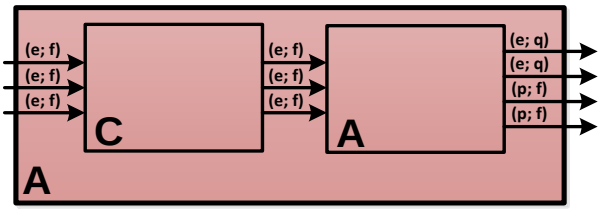


Figure 87 : Exemple d'une séquence équivalente à un processus de type T. Adaptée de (Feliot, 1997)

- Une séquence, de longueur quelconque, consommant en entrée des flux d'énergie et produisant des flux de puissance en sortie est équivalent à un processus global de type T (Figure 88).

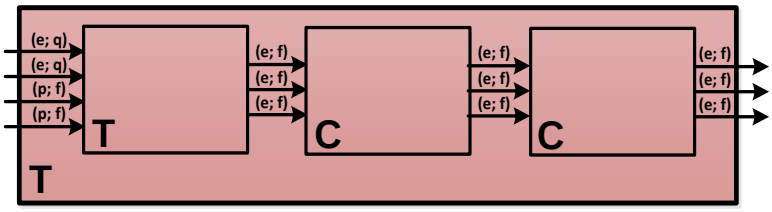


Figure 88 : Exemple d'une séquence équivalente à un processus de type E. Adaptée de (Feliot, 1997)

Les règles d'équivalence donnent la forme de décomposition des différents types A, T, C de niveau N en réseaux de processus de type A, T, C à un niveau hiérarchique N+1.

Dynamique des processus énergétiques

D'un point de vue dynamique, nous cherchons à formaliser les différentes transformations d'énergies opérant à l'intérieur de chacun des processus d'accumulation, de transport et de conversion. Compte tenu de sa mission « *L'eau de secours doit alimenter le générateur de vapeur afin d'évacuer la puissance résiduelle du circuit primaire en situation accidentelle ou incidentelle* », le processus CISPI est assimilable à un processus hydraulique opérant sur des flux d'énergies et de puissances. En ce sens nous limitons notre étude aux processus énergétiques dans le domaine de l'hydraulique.

Dans le domaine de l'hydraulique, le processus A est vu comme un processus d'accumulation, correspondant à la classe de processus de transformations de puissance en énergie selon (20) avec e = Pression (Pr), q= Volume (V), f = Débit volumique (Qv) et p = Moment de pression (P_P). Ainsi la transformation opérée par le processus A prend la forme suivante : $(\boldsymbol{Pr}, \boldsymbol{Qv}) \rightarrow \{(\boldsymbol{V}, \boldsymbol{Pr}), (\boldsymbol{Pp}, \boldsymbol{Qv})\}$ où le couple $(\boldsymbol{Pp}, \boldsymbol{Qv})$ représente une énergie cinétique. L'énergie cinétique à l'intérieur de la fonction d'accumulation étant nulle, cela nous amène à reconsidérer le processus A selon la transformation : $(\boldsymbol{Pr}, \boldsymbol{Qv}) \rightarrow (\boldsymbol{V}, \boldsymbol{Pr})$ avec $\boldsymbol{V} = \int_0^t \boldsymbol{Qv}(\mathrm{t})\ \boldsymbol{dt}$. En ce sens, le processus de stockage de type A produit une énergie hydraulique accumulée dans le stockage représentée par le vecteur (***Pr_2 ; V_2***) où ***(Pr_2)*** est la pression statique au fond de la cuve et ***(V_2)*** le volume de fluide dans la cuve. Cette énergie résulte du bilan de l'accumulation des puissances hydrauliques en entrée et en sortie. La puissance hydraulique en entrée correspond au fluide fourni en entrée représenté par le vecteur ***(Pr_0; Qv_0)*** où

Partie 3 : Co-spécification exécutable de constituants de conduite du procédé du système CISPI

(Pr_0) représente la pression du fluide à son admission dans la cuve et Qv_0 le débit en entrée. La puissance en sortie quant à elle correspond au fluide cédé par la cuve, elle est représentée par le vecteur ***(Pr_1 ; Qv_1)***, où ***(Pr_1)*** est la pression du fluide à sa sortie (c'est également celle qui règne au fond de la cuve) et ***(Qv_1)*** est le débit en sortie (Tableau 11).

Représentation du processus dans le domaine de l'hydraulique	Relation d'entrées-sorties
	On a : $Pr_1 = Pr_2$ (Pa) et $V_{Stockage}(t) = V_{Stockage}(0) + \left(\int_0^t Qv_{Entrée}(t)\ dt - \int_0^t Qv_{Sortie}(t)\ dt\right)$ Avec $V_{Stockage}(t) = V_2(t),\ Qv_{Entrée}(t) = Qv_0(t)$ et $Qv_{Sortie}(t) = Qv_1(t)$ $V_2(t) = V_2(0) + \left(\int_0^t Qv_0(t)\ dt - \int_0^t Qv_1(t)\ dt\right)$ (m^3)

Tableau 11 : Spécification de la dynamique du processus énergétique d'accumulation

Dans le domaine de l'hydraulique le processus C est vu comme un processus de conversion, correspondant à la classe de processus de transformations de puissance en puissance selon (22). Ainsi la transformation opérée par le processus C prend la forme suivante $(\boldsymbol{Pr}, \boldsymbol{Qv}) \xrightarrow{\alpha;\, 1/\alpha} (\boldsymbol{Pr}, \boldsymbol{Qv})$. En ce sens, le processus de dissipation de type C, dû à la restriction d'une vanne par exemple, produit une puissance en sortie ***(Pr_1 ; Qv_1)*** et des pertes de charges (***Pr2 ; Qv2***) que l'on reboucle à l'entrée du processus pour traduire d'un point de vue énergétique une contre réaction. Notons que les flux ***(Qv_0)*** et ***(Qv_1)*** respectivement en entrée et en sortie du processus correspondent au même flux matériel ***($Qv_0 = Qv_1$)*** (Tableau 12)

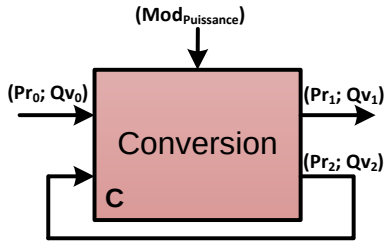
Représentation du processus dans le domaine de l'hydraulique	Relation d'entrées-sorties
	On a : $Qv_0(t) = Qv_1(t) = Qv_2(t)$ Considérons les pertes de charges singulières exprimées par : $\Delta P = k * \frac{\rho * v(t)^2}{2}$ Avec : • k = coefficient constant. • ρ = Masse volumique de l'eau = 995.58 kg.m^{-3} • $\Delta P = Pr_1(t) - Pr_0(t)$ (Pa) • $v(t) =$ Vitesse d'écoulement $(m.s^{-1}) = \frac{Qv(t)}{Section_{Forme}}$ $Pr_0(t) = \frac{k * \rho}{2} * \left(\frac{Qv_0(t)}{Section_{Forme}}\right)^2 * Mod_{Puissance} + Pr_1(t)$ $Pr_3(t) = Pr_0(t) - Pr_1(t)$

Tableau 12 : Spécification de la dynamique du processus énergétique de conversion

Dans le domaine de l'hydraulique le processus T est vu comme un processus de transport, correspondant à la classe de processus de transformations d'énergie en puissance selon (21). Ainsi la transformation opérée par le processus T prend la forme suivante $\{(\boldsymbol{V},\boldsymbol{Pr}),(\boldsymbol{Pp},\boldsymbol{Qv})\} \rightarrow (\boldsymbol{Pr},\boldsymbol{Qv})$ où le couple $(\boldsymbol{Pp},\boldsymbol{Qv})$ représente une énergie cinétique. Selon les règles définis dans le paragraphe 5.3.1.3, un processus de type T ne peut succéder qu'à un processus de type A, or l'énergie cinétique en sortie d'un processus de type A dans le domaine de l'hydraulique est nulle. Ceci nous amène à reconsidérer le processus T selon la transformation $(\boldsymbol{V},\boldsymbol{Pr}) \rightarrow (\boldsymbol{Pr},\boldsymbol{Qv})$ avec $\boldsymbol{Q_v} = \frac{\boldsymbol{dV}}{\boldsymbol{dt}}$ (Tableau 13).

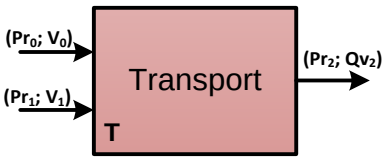
Représentation du processus dans le domaine de l'hydraulique	Relation d'entrées-sorties
	On a : $Pr_0 = Pr_2$ (Pa) De plus, d'après le théorème de Bernoulli : $Pr_a(t) + \frac{1}{2}\rho * v_a(t)^2 + \rho * g * z_a(t) = Pr_b(t) + \frac{1}{2}\rho * v_b(t)^2 + \rho * g * z_b(t)$ Avec : • $Pr_a(t) = Pr_1(t)$ pression à la surface du stockage (Pa), • $v_a(t)$ = vitesse de l'eau à la surface du stockage = 0 ($m.s^{-1}$), • $z_a(t)$ = hauteur de l'eau dans la cuve de stockage = $\frac{V_0(t)}{Section_{Stockage}}$ (m) • $Pr_b(t) = Pr_0(t)$ pression de l'eau à la sortie du stockage (Pa), • $v_b(t)$ = vitesse de l'eau à la sortie du stockage ($m.s^{-1}$), • $z_b(t)$ = hauteur de l'eau au fond de la cuve = 0 m • ρ = Masse volumique de l'eau = 995.58 $kg.m^{-3}$ • g = Constante gravitationnelle = 9.80665 $m.s^{-2}$ ➔ $v_b(t) = \sqrt{\left\lvert \frac{2*(Pr_1(t) - Pr_0(t))}{\rho} + 2 * g * \frac{V_0(t)}{Section_{Stockage}} \right\rvert}$ On a : $Qv_b(t) = v_b(t) * Section_{Sortie}$ ➔ $Qv_2(t) = Section_{Sortie} * \sqrt{\left\lvert \frac{2*(Pr_1(t) - Pr_0(t))}{\rho} + 2 * g * \frac{V_0(t)}{Section_{Stockage}} \right\rvert}$

Tableau 13 : Spécification de la dynamique du processus énergétique de transport

La spécification de ces processus énergétiques, de leurs dynamiques et des règles d'agencements permet d'envisager une spécification exécutable du procédé de la plate-forme CISPI (Annexe 5-5).

5.3.1.2. Construction d'un modèle multi-physique exécutable de l'interaction ciblée

Afin de spécifier le procédé de la plate-forme CISPI en un modèle exécutable permettant à la fois de représenter la structure et la dynamique du procédé, l'**espace-solution** d'ingénierie du procédé a retenu le langage Modelica (Modelica, 2012). Modelica est un langage textuel orienté-objet pour la modélisation structurelle et comportementale de systèmes complexes et multidomaines (Fritzson, et al., 2002) développé depuis 1996 par une large communauté et standardisé par l'association à but non lucratif : « Modelica

Association » [54]. Il propose également une approche graphique de modélisation à l'aide de diagrammes de blocs affranchissant ainsi l'ingénieur de l'écriture fastidieuse des éléments de programmations sous-jacents.

D'un point de vue structurel, il permet la modélisation des systèmes par assemblage de composants. Ces composants sont définis par des signaux (logique ou analogique sans nécessairement avoir une signification physique) et par des ports d'énergie caractérisé par des couples de variables généralisées (effort, flux) selon le tétraèdre de Paynter. La connexion avec les autres composants à travers les ports d'énergie et de matière est réalisée par le biais de connecteurs. Les connexions sont régies par les lois de Kirchhoff généralisées (conservation de l'énergie) : la somme des valeurs des variables de flux qui aboutissent en un même point est nulle, et les variables qui ne sont pas des flux sont égales. D'un point de vue comportemental, chacun des composants est décrit mathématiquement par un ensemble d'équations différentielles, algébriques et/ou discrètes définissant leur comportement dynamique. Les modèles continus, discrets et hybrides sont supportés par ce langage. L'assemblage des composants selon les lois de la physique permet de garantir le comportement global du système. Le langage Modelica permet de réaliser à la fois une description causale et acausale des modèles. Un modèle causal est caractérisé par des connexions de type signaux unidirectionnels alors que le modèle acausal possède des ports énergétiques. La modélisation acausale présente des avantages par rapport à la modélisation causale (Gaaloul, 2012) car elle offre une description :

- « Plus naturelle : cette description est proche de la réalité car les connexions y représentent un transit des variables énergétiques (efforts, flux) entre les modèles ».
- Plus flexible : l'introduction d'un nouveau modèle ne nécessite pas le ré-ordonnancement manuel des équations du système pour établir la nouvelle causalité
- Plus automatique : la causalité de chaque modèle n'est plus traitée au préalable par transformation sous forme d'une représentation d'état. Il suffit de décrire le modèle à travers des équations différentielles algébriques et la causalité du système sera traitée dans sa globalité par le compilateur ».

Le fait que le langage repose sur le paradigme de programmation orientée objet, présente un intérêt majeur de réutilisation plus aisée des composants du système grâce aux concepts d'héritage et de polymorphisme. De nouveaux modèles peuvent être ainsi créés par extension des modèles existants. Cela est intéressant pour la création de nombreuses bibliothèques de modèles réutilisables de domaines variés (La bibliothèque standard Modelica couvre plusieurs domaines de la physique : mécanique, électrique, thermique, hydraulique … (Modelica, 2008), en Physiologie (Mateják, et al., 2014), en système à évènements discrets (Otter, Årzén, & Dressler, 2005), système d'air conditionné , de ventilation et de chauffage (Wetter, 2010), en automobile (Andreasson, 2003),….). A notre connaissance, la majorité des bibliothèques Modelica permettent la modélisation des systèmes physique uniquement d'un point de vue logique. Bien que les bibliothèques « Bond Graph » (Cellier & Nebot, 2005) permettent de modéliser les procédés physiques à un niveau fonctionnel, cette représentation ne nous apparait pas comme la plus adéquate pour la structuration du procédé en processus. De plus, la notation « Bond graph » ne facilite pas la traduction en langage SysML ainsi que le partage de **connaissances** entre l'ingénieur du procédé et l'ingénieur système. Ces raisons nous ont motivées à créer une nouvelle bibliothèque Modelica permettant la représentation des procédés physiques à un niveau fonctionnel selon les processus énergétiques Accumulation, Transport et Conversion tels que définis précédemment (Cf. Annexe 5-6).

[54] https://www.modelica.org/

Ainsi en raffinant le modèle du procédé de la plate-forme CISPI (Figure 77), l'**espace-solution** d'ingénierie du procédé spécifie un modèle exécutable de ce dernier, selon un agencement de processus accumulation, transport et conversion décrits dans la bibliothèque Modelica (Figure 89). En ce sens, les actions de modulation de débit sont vues comme des processus de conversion, l'action de stockage comme un processus d'accumulation et les actions de transport comme des processus de transport. Cette **spécification** *(S_{IPro})* modélisée avec l'outil Dymola permet à l'ingénieur du procédé de compiler le code Modelica pour générer un modèle exécutable du procédé. Le simulateur intégré à l'outil a pour tâche de résoudre le système d'équations à chaque pas temporel, simulant ainsi le comportement dynamique du procédé étudié.

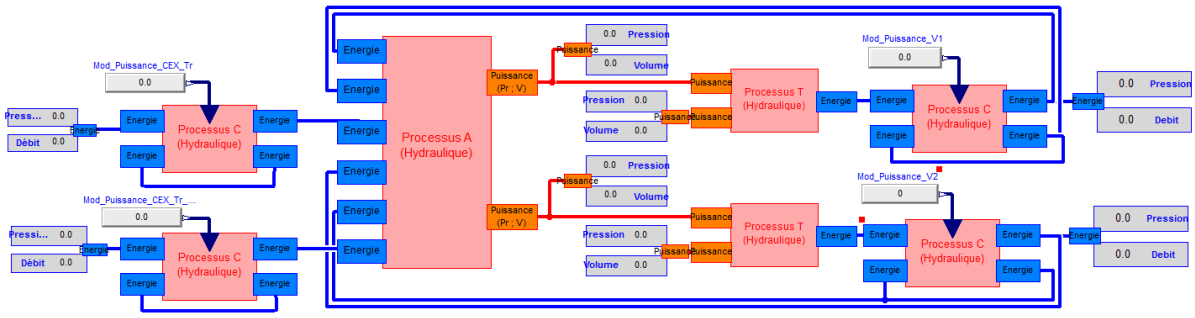


Figure 89 : Modèle orienté flux énergétique du procédé CISPI sous forme de diagramme de bloc en langage Modelica

La simulation du procédé, offre à l'ingénieur du procédé deux niveaux de vérification du modèle proposé. En effet, la compilation permet de vérifier la construction du modèle d'un point de vue syntaxique (*erreurs dans le code*) ainsi que sa complétude (*oublie de la déclaration d'une variable, d'une connexion…).* Une fois le modèle compilé, l'exécution de modèle offre un second niveau de vérification. Ainsi, en définissant différent scénarios de test l'**espace-solution** d'ingénierie du procédé est capable de vérifier que le procédé spécifié satisfait les exigences décrites par l'**espace-problème** d'ingénierie système. Par exemple, l'ingénieur du procédé est en mesure de vérifier que le dimensionnement de la **spécification** du procédé (***S_{IPro}***) de la plate-forme CISPI est adapté pour atteindre les performances décrit par l'exigence (***$R_{IS1.7-0.1}$***).

«Contrainte» Valeur_Debit_Alimentation_Secours
ID = RIS1.7-0.1
Le débit d'alimentation en eau de secours doit être de 30L/min
Puits:RhpString=Espace Solution : Ingénierie Procédé; Ingénierie Auto… Source:RhpString=Espace Problème : Ingénierie Système

Figure 90 : Exigence $R_{IS1.7-0.1}$: Performance requise du procédé relative au débit d'alimentation

Le scénario de test décrit par l'ingénieur du procédé consiste à vérifier par simulation que pour une situation initiale prescrite, le procédé spécifié est capable de fournir le débit d'alimentation attendu (30 L/min soit 0.0005 $m^3.s^{-1}$). Notons que pour ce scénario le débit d'alimentation est fonction de la hauteur du stockage de

l'eau ainsi que de la modulation en puissance du processus de conversion aussi bien pour la voie d'alimentation 1 que pour la voie d'alimentation 2.

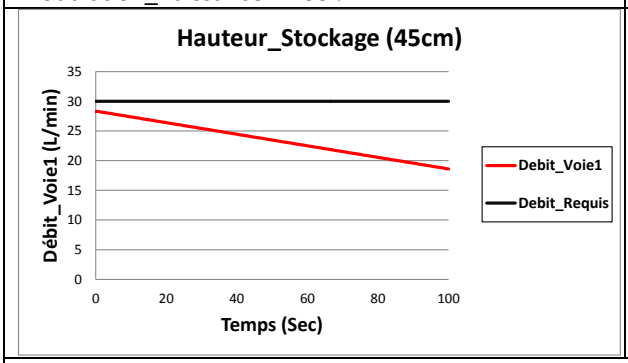
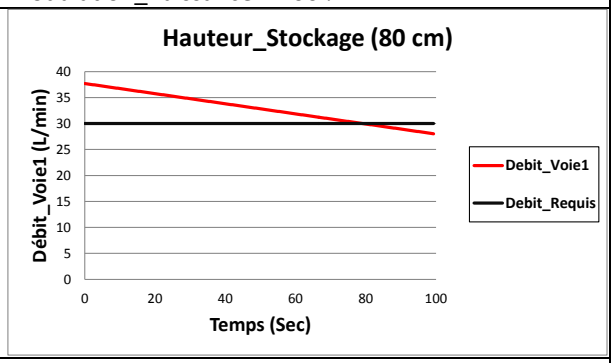
Situation initiale : Hauteur au niveau du stockage : 45 cm Modulation_Puissance = 100 %	Situation initiale : Hauteur au niveau du stockage : 80 cm Modulation_Puissance = 100 %
	

Tableau 14 : Traces d'exécution pour vérifier la satisfaction de l'exigence $R_{IS1.7\text{-}0.1}$

L'**espace-solution** d'ingénierie du procédé vérifie ainsi par exécution de modèles, que pour une hauteur de stockage inférieure à 50 cm le procédé de la plate-forme CISPI tel qu'il est modélisé ne pourra pas fournir un débit d'alimentation de 30 L.min^{-1} (Tableau 14). En ce sens, le procédé dérive et prescrit une nouvelle exigence ($S_{IPro1.0}$) définissant le niveau minimum dans la cuve de stockage à 50 centimètres (Figure 91).

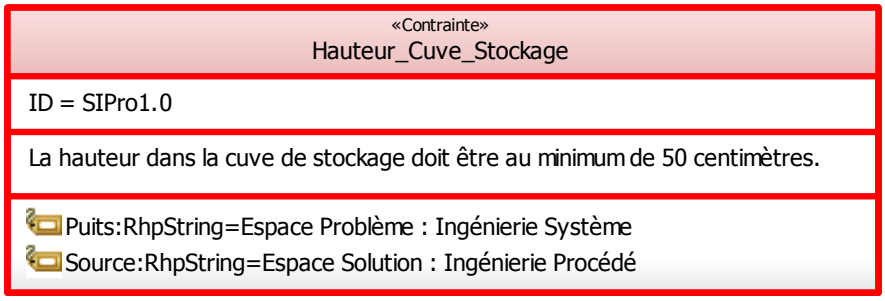


Figure 91 : Exigence prescrite par l'*espace-solution* ingénierie du procédé ($S_{IPro1.0}$) : Hauteur cuve stockage

Une fois que l'**espace-solution** d'ingénierie du procédé a vérifié sa **spécification** du procédé (S_{IPro}), celui-ci la prescrit à l'**espace-problème** d'ingénierie système pour validation par co-exécution de modèles afin d'en faire un constituant du système CISPI.

5.3.1.3. Traduction d'un modèle exécutable constitutif de l'interaction multi-physiques ciblée

En ce sens la **spécification** du procédé en langage Modelica doit être traduite en langage SysML. Ces règles de traduction ont fait, et continuent de faire l'objet de nombreux travaux scientifiques (Renier & Chenouard, 2011), (Johnson, Kerzhner, Paredis, & Burkhart, 2012), (Garro & Tundis, 2014)… ayant mené à la création d'un standard (OMG, 2012b). Ce standard spécifie un profil SysML : « SysML4Modelica » permettant

de représenter les artefacts de modélisation Modelica ainsi que les règles de transformation entre SysML et le langage Modelica. Sur la base de ces travaux, nous avons défini une table de correspondance entre les artefacts de modélisation Modelica et SysML utilisés pour la **spécification** du procédé (Cf. Annexe 5-7). Cette table de correspondance enrichit ainsi la **connaissance** de l'**espace-solution** d'ingénierie du procédé (K_{IPro}) pour prescrire une **spécification** du procédé en langage SysML (S_{IPro_SysML}).

5.3.1.3.1. Modèle structurel multi-physiques

Dans ce paragraphe, **l'espace-solution** d'ingénierie du procédé se questionne sur la traduction de la structure du procédé en langage Modelica vers le langage SysML. La modélisation structurelle en SysML, reposant essentiellement sur deux diagrammes utilisés conjointement, permet de représenter le procédé selon la triade « **Relationship, Whole, Parts** » du conceptagon :

- le diagramme de définition blocs (bdd) sert à définir, classer et hiérarchiser les blocs (« block »), qui constituent la brique de base pour la modélisation de la structure d'un système (**Whole**)
- le diagramme interne de bloc (ibd) modélise la structure interne d'un bloc. Il représente les composants de ce bloc modélisés par des instances de blocs (« Part »), ainsi que les flux de matière, d'énergie et/ou d'information pouvant circuler entre ces composants

Par conséquent, la traduction du modèle Modelica s'effectue en deux étapes : La première étape consiste à traduire les « models » Modelica en « blocks » SysML, dans un diagramme de définition de blocs (bdd), pour représenter les dépendances (**Relationship**) entre les processus Accumulation, Transport, Conversion (**Parts**) et le processus CISPI (**Whole**) (Figure 92). Ces processus énergétiques sont alors vus comme des composants fonctionnels du processus global définis selon la **spécification** du procédé CISPI (S_{IPro}).

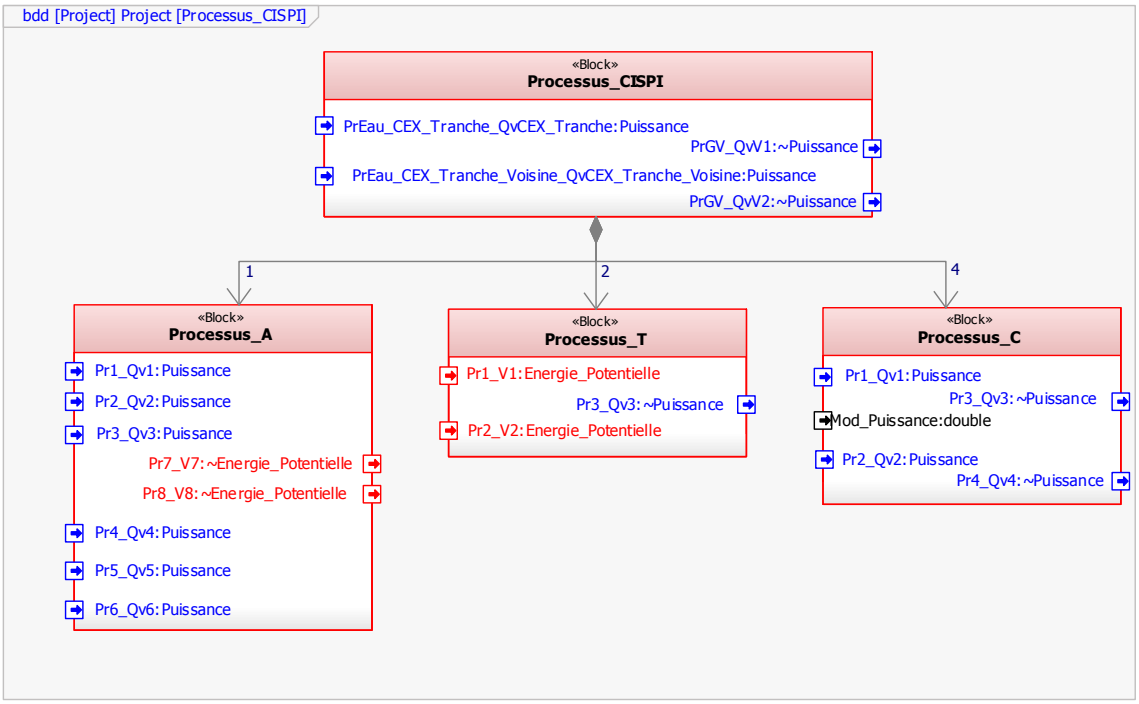


Figure 92 : Diagramme de définition de bloc (bdd) SysML - Composition du processus CISPI

Les interfaces permettant aux processus d'échanger des flux, sont représentées par des « *FlowPorts* ». En ce sens, les « *FlowPorts* » sont la traduction SysML des connecteurs Modelica représentant les interfaces des processus, caractérisées selon un type de flux et une direction. Les différents flux (*flux d'énergie cinétique, flux d'énergie potentielle, flux de puissance*) traversant le processus CISPI sont des flux non atomiques, composés chacun d'entre eux par deux autres flux. Prenons l'exemple du flux de puissance composé d'un flux représentant le débit volumique et d'un second représentant la pression, le « *FlowPort* » de puissance associé est alors typé par une spécification de flux (« *flowSpecification*») incluant plusieurs propriétés de flux (« *flowProperties* ») caractérisées chacune par un nom, un type et une direction (Figure 93). Chacun des flux traversant le système sont ainsi spécifiés par l'**espace-solution** d'ingénierie du procédé (Cf. Annexe 5-8).

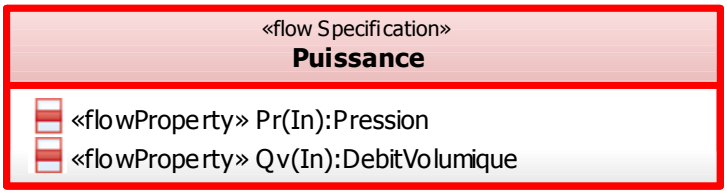


Figure 93 : FlowSpecification de type Puissance

La direction des ports SysML est spécifiée selon la triade « ***Transformation, Input, Output*** » définissant les entrées et les sorties des processus selon l'écoulement des flux de matière et d'énergie à travers le système. Notons que SysML offre la possibilité de représenter des « *FlowPorts* » conjugués, c'est-à-dire des ports typés par des « *flowSpecification*» mais dont toutes les directions des propriétés ont été inversées. Cela permet de réutiliser la même spécification pour représenter des ports de directions opposées (entrée-sortie). Dans le logiciel IBM Rational Rhapsody, les ports conjugués sont distingués des autres ports par un : « ~ » devant le type du « *FlowPort* ».

La seconde étape consiste à représenter les flux de matières et d'énergies circulant entre les processus Accumulation, Conversion, Transport composant le système. Ces flux sont représentés en langage Modelica par équations de type « *Connect* » spécifiant une interaction entre des connecteurs Modelica de même type. En lange SysML ces flux sont représentés dans les diagrammes internes de blocs (ibd) par des connecteurs reliant deux instances de blocs (« Parts ») entre elles, leur fournissant l'opportunité d'agir à travers les « *FlowPorts* » (Figure 94).

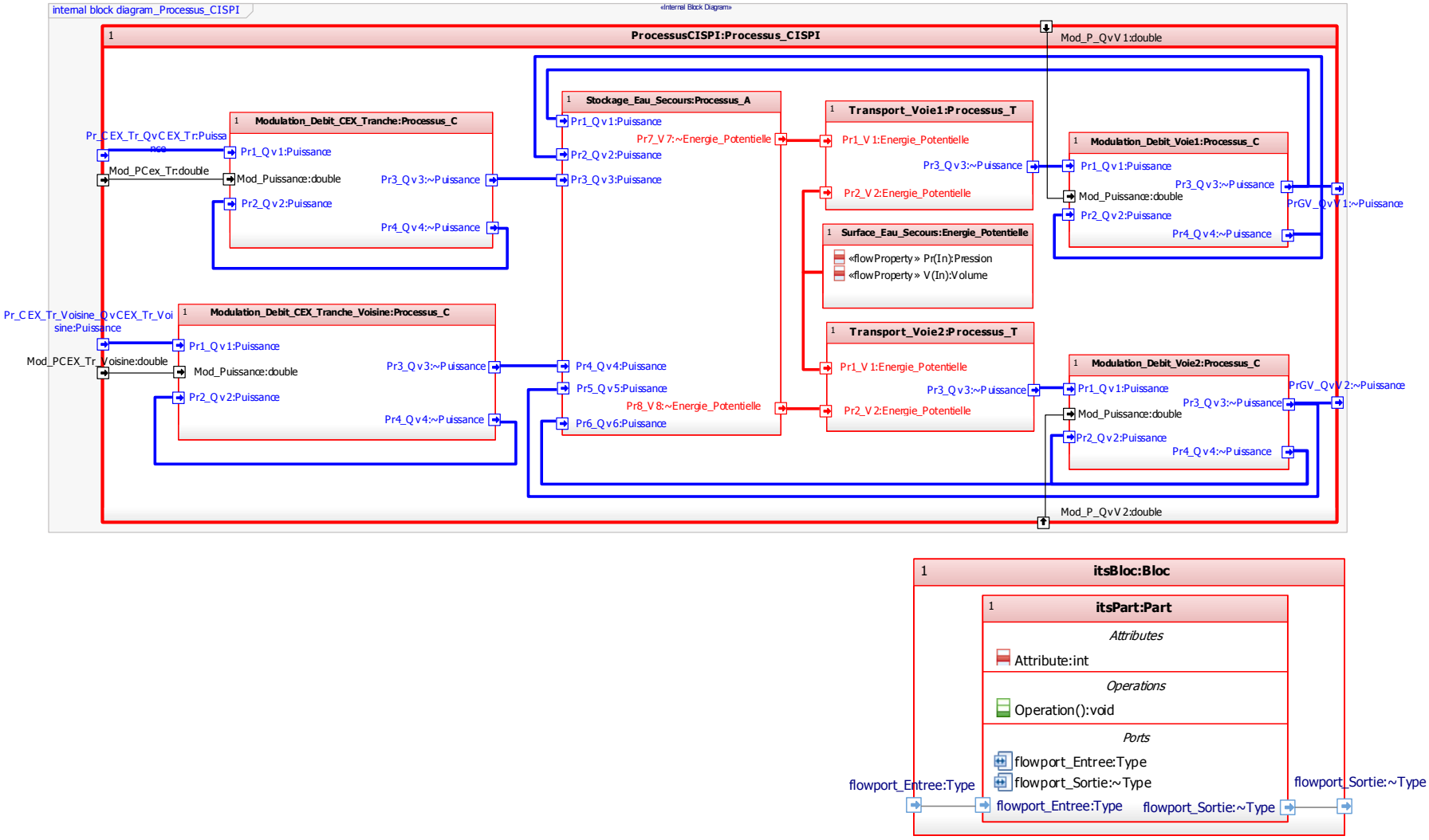


Figure 94 : Diagramme interne de bloc SysML – Structure du processus CISPI

5.3.1.3.2. Modèle comportemental multi-physiques

Dans ce paragraphe, l'**espace-solution** d'ingénierie du procédé se pose la question de traduire la dynamique du procédé en langage Modelica vers le langage SysML. Bien que le langage SysML propose un ensemble de diagrammes permettant de représenter le comportement d'un bloc (*diagramme de séquence, diagramme d'activités, diagramme d'états*), ces diagrammes n'apparaissent pas appropriés pour représenter la dynamique continue du procédé. En effet, les diagrammes comportementaux SysML, dont l'origine provient des diagrammes UML (Unified Modeling Language) dans le domaine de l'informatique, sont plus adaptés pour représenter des comportements discrets que continus. En ce sens, notre intérêt se porte sur les diagrammes paramétriques permettant de représenter des contraintes sous la forme d'équations et de relations mathématiques entre des grandeurs physiques ou logiques. Ces contraintes représentés dans les diagrammes paramétriques sont des relations acausales (il n y a pas de notion d'entrée-sortie au niveau des paramètres), par conséquent, ce diagramme est classé dans les diagrammes structurels SysML. Cependant en fixant la relation d'entrée-sortie entre les paramètres manipulés par ces contraintes, l'exécution des équations devient alors causale permettant ainsi d'envisager la modélisation de la dynamique du procédé. Nous rappelons que l'**espace-solution** d'ingénierie du procédé définit les entrées-sorties du système lors de la spécification du procédé à travers la triade « ***Transformation, Input, Output*** », définis par la direction des ports au niveau des blocs SysML. La causalité d'exécution des blocs est définit selon l'écoulement des flux de matières et d'énergies à travers le système représenté dans le diagramme interne de bloc.

A titre d'exemple, nous illustrons la traduction du modèle dynamique du processus T en langage Modelica vers un diagramme paramétrique SysML (Figure 95). Comme nous venons de le voir la dynamique d'un processus est spécifié à travers les ports d'entrée-sortie ainsi que les relations de transformations entre ces entrées-sorties. Les connecteurs Modelica (représentant des ports) sont traduits dans un diagramme paramétrique par des paramètres (« *Parameter* »). Ces paramètres ne permettant pas de définir une direction, nous représentons les ports d'entrée à gauche (***Pr1_V1, Pr2_V2***) et les ports de sortie à droite (***Pr3_Qv3***). Les autres variables des équations Modelica sont traduites par des attributs (« *Attribute* »). Les équations Modelica sont quant à elle traduites par des contraintes SysML définies entre des paramètres et des variables. Les contraintes sont représentés par un bloc avec un stéréotype « *constraint* ». Cette représentation s'effectue dans un diagramme de définition de blocs (Cf. Annexe 5-9). Ces contraintes sont ensuite instanciées dans le diagramme paramétrique sous forme de « *constraintProperties* » et reliées entre elles à travers leurs paramètres par des liens « *BindingConnector* » pour représenter le système d'équations définissant les transformations à l'intérieur du processus T.

Ainsi en appliquant l'ensemble des règles de traduction, l'**espace-solution** d'ingénierie du procédé représente la spécification de la dynamique des processus Accumulation, Transport et Conversion en langage SysML (Cf. Annexe 5-9). La structure, c'est-à-dire les relations entre ces processus est définis dans le diagramme interne de blocs, conduisant ainsi à la spécification de la dynamique globale du procédé de la plate-forme CISPI.

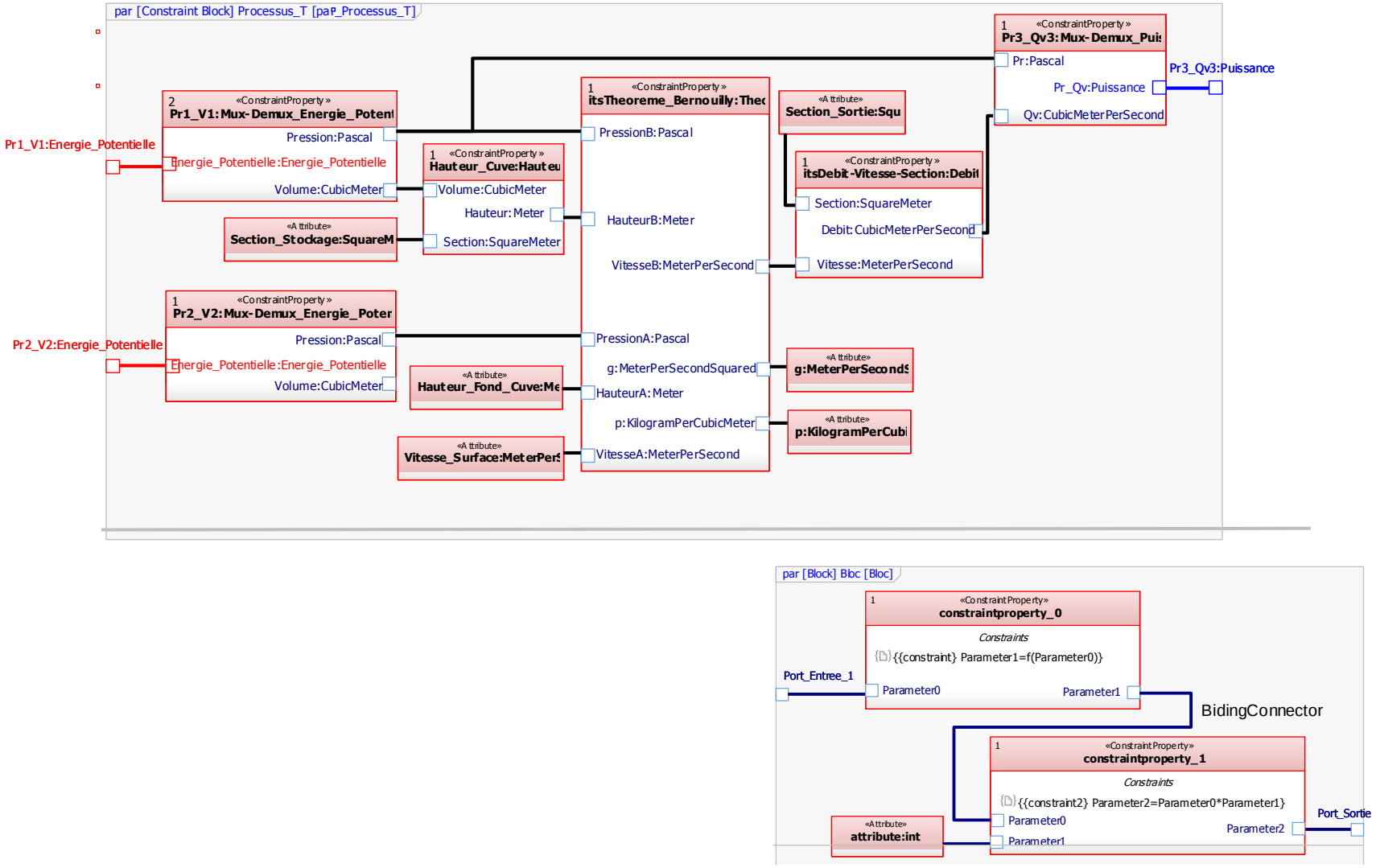


Figure 95 : Diagramme paramétrique SysML – Dynamique du processus T

5.3.2. Co-intégration système d'un constituant d'interaction multi-physiques de modulation de débit d'eau

De même que pour la première itération entre l'**espace-problème** d'ingénierie système et **l'espace-solution** d'ingénierie du procédé, la spécification du procédé prescrite et traduite en SysML doit être validée. La plate-forme CISPI étant un système critique, l'ingénieur système se focalise dans un premier temps sur la validation de la spécification au regard des exigences de sûreté. Cette étape de validation nécessite une nouvelle itération avec l'**espace-solution** d'ingénierie de sûreté afin d'assurer que l'architecture fonctionnelle prescrite satisfait à l'exigence de sûreté ($\boldsymbol{R_{IS1.5\text{-}0.2}}$) (*relative à la redondance des voix d'alimentation en eau de secours*). Une fois la spécification validée d'un point de vue sûreté, l'ingénieur système peut continuer la validation de la spécification vis-à-vis des autres exigences (fonctionnelles, contraintes, ….).

Une fois prescrite dans l'**espace-solution** d'ingénierie système, la spécification du procédé en langage SysML offre la possibilité à l'ingénieur système d'analyser les performances du système d'intérêt en évaluant les diagrammes paramétriques. L'outil IBM Rational Rhapsody, retenu par l'**espace-solution** d'ingénierie système, permet d'évaluer de façon automatisée les diagrammes paramétriques, en ajoutant le profil PCE (*Parametric Constraint Evaluator*) au projet SysML (Rahman, Nor, & Mizukawa, 2012) .En ce sens, les contraintes dans les diagrammes paramétriques doivent être décrites selon le langage PCEL (*Parametric Constraint Expression Language*) dont les opérateurs et les fonctions ont été empruntés au langage d'action Modelica. Notons que l'**espace-solution** du procédé devant traduire sa spécification du procédé du langage Modelica vers SysML n'a pas à traduire les équations dans un autre langage, évitant ainsi une source d'erreurs potentielles. En basculant dans la vue d'évaluation des contraintes, l'ingénieur système est ainsi capable d'exécuter les diagrammes paramétriques afin d'évaluer que la spécification du procédé ($\boldsymbol{S_{IPro_SysML}}$) satisfait les exigences du système. L'outil se charge alors de traduire les contraintes dans un langage spécifique et interprétable par des solveurs d'équations algébriques résolvant le système d'équations. Cependant à l'heure actuelle, les équations non linéaires ne sont pas prises en compte par l'outil d'évaluation des contraintes (PCE) alors que la plus part des systèmes physiques sont non linéaires. Cela fait apparaitre les limites des outils en ingénierie système qui n'ont pas vocation à intégrer l'ensemble des outils et techniques des ingénieries spécialistes au sein d'un même outil. De plus, il peut arriver dans certains cas que les ingénieries spécialistes ne veulent pas partager le contenu de leurs modèles pour des raisons de confidentialités, limitant également le travail de l'ingénieur durant la validation des spécifications.

Les techniques de co-simulation actuelles permettent de contourner ces problématiques, en exécutant les modèles spécialistes dans leurs domaines, avec leurs propres outils et en orchestrant l'échange des données calculées ou requises entre ces modèles mais également avec les modèles SysML dans le domaine de l'ingénierie système. En ce sens, l'ingénieur système est en mesure de valider par co-exécution de modèles la spécification du procédé en évaluant ses performances selon un ensemble de cas tests. Nous focalisons notre étude sur l'exigence ($\boldsymbol{R_{IS1.7\text{-}0.1}}$), que l'**espace-problème** d'ingénierie système n'a pas pu valider au cours de la première itération et qui a motivé le raffinement de la spécification du procédé en un modèle exécutable au cours de cette seconde itération. D'un point de vue technique, le bus de co-simulation agit comme un contrôleur de communication entre chaque couple modèle/simulateur connectés au bus de co-simulation via des ports d'entrée-sortie intégrés aux modèles. En ce sens, chacune des interfaces du modèle du procédé, des cas tests, de l'interaction$_{\text{Procédé-Environnement}}$ et de l'environnement sont connectés à des ports de type « Cosimate » (Figure 96) permettant l'échange des données du système à travers le bus de co-simulation (Figure 81).

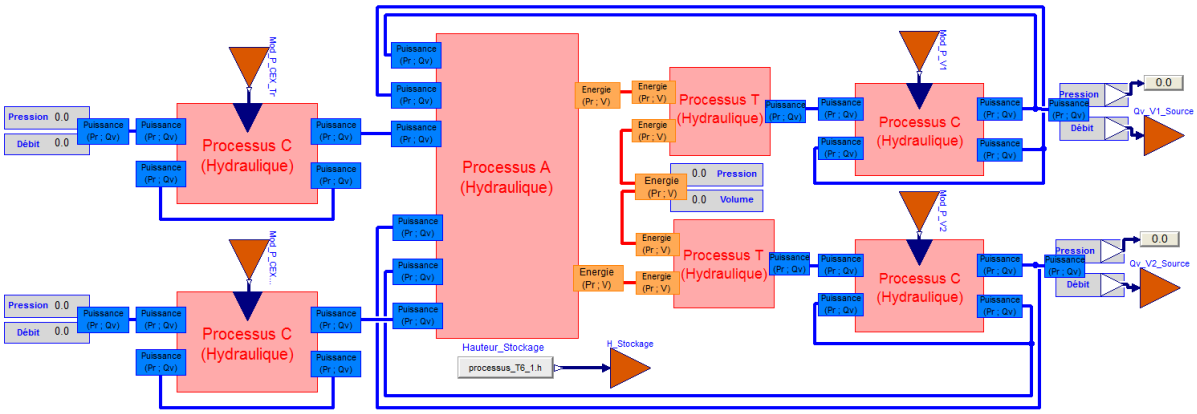


Figure 96 : Interfaces du modèle Modelica du procédé connectées à des ports "Cosimate" (en orange)

Afin de configurer convenablement l'environnement de co-simulation, le domaine d'ingénierie système est en charge d'architecturer l'ensemble des modèles spécialistes. Pour cela, l'ingénieur système se réfère aux modèles prescrits en langage SysML, pour identifier leurs interfaces d'entrées/sorties et définir les liens, et les données échangées entre ces différents modèles. Cette architecture est spécifiée en langage SysML dans un diagramme interne de bloc (ibd) (Figure 97), et sert de référence pour la configuration de l'outil CosiMate (Cf. Annexe 5-10).

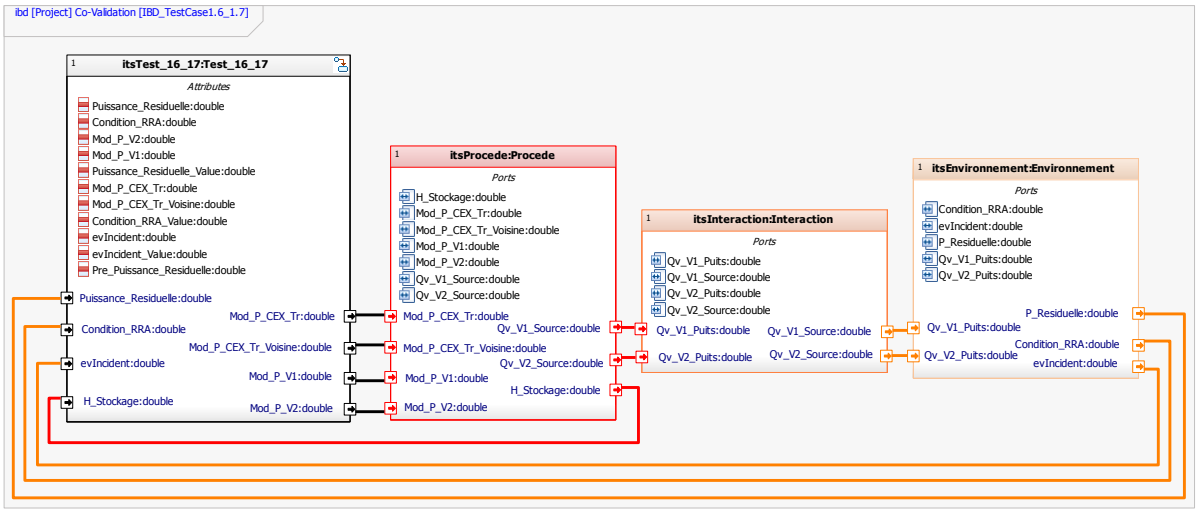


Figure 97 : Première itération - Définition de l'architecture de co-simulation système (*Sous-système physique*) par l'espace-problème d'ingénierie système

Une fois l'ensemble des modèles interconnectés les uns avec les autres à travers le bus de co-simulation, l'ingénieur système est en mesure de dérouler les scénarios de tests et de valider par co-exécution de modèles la spécification du procédé de la plate-forme CISPI au regard des exigences parties prenantes. Afin de faciliter le dialogue avec le domaine opérationnel, mais également d'interagir avec l'ensemble des modèles spécialistes en co-simulation, l'ingénieur système spécifie une interface de supervision (Figure 98) permettant d'afficher les éléments essentiels à la validation, tout en masquant l'exécution des modèles spécialistes en tâche de fond.

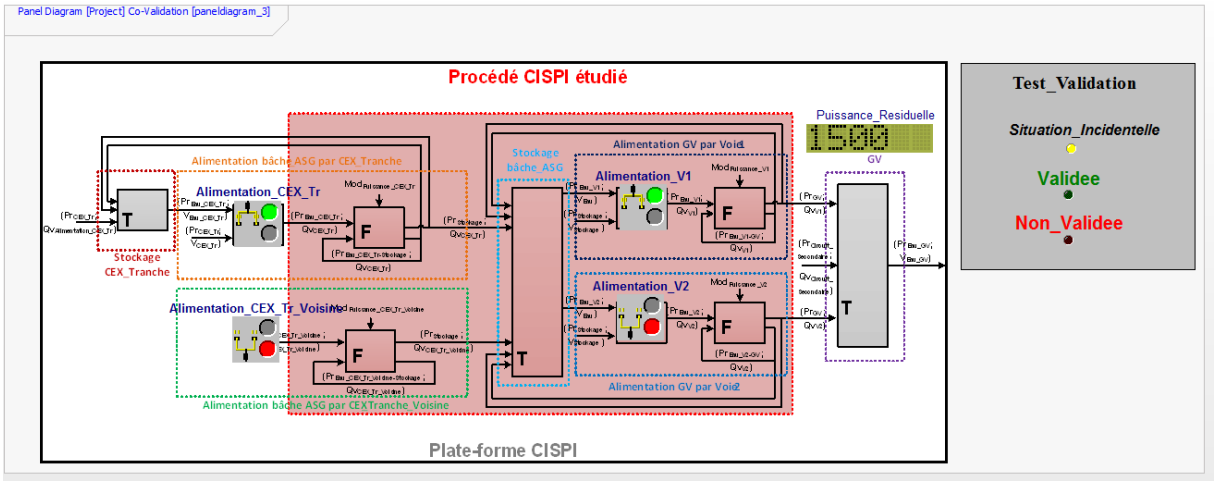


Figure 98 : Interface de supervision du déroulement du "TestCase_1.6_1.7" défini dans un "Panel Diagram" Rhapsody

Ainsi, notre environnement de co-simulation (Figure 99) permet l'exécution concourante de l'ensemble des modèles spécialistes, dont l'exécution automatique du scénario de test (Figure 80) pour valider la satisfaction de la spécification du procédé CISPI (S_{IPro}) au regard des exigences ($R_{IS1.6}$ et $R_{IS1.7}$).

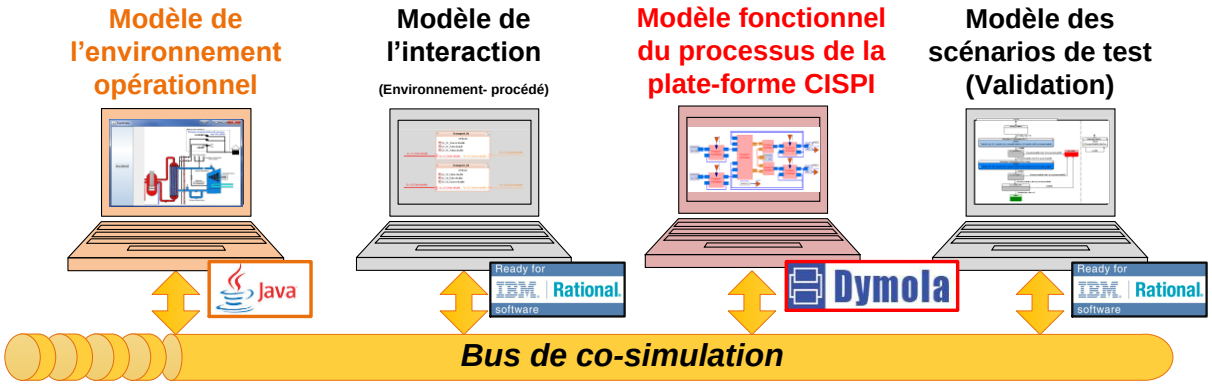


Figure 99 : Environnement de co-simulation système pour la validation de la spécification du processus de la plate-forme CISPI (S_{IPro_SysML})

Les traces d'exécution de ce scénario de test peut être récupérées sous la forme d'un diagramme de séquence SysML (Figure 100) permettant de conserver l'historique des comportements exécutées, mais aussi permettant la comparaison avec des scénarios de références définis en phase d'ingénierie (Figure 100). En ce sens, et après analyse de cette trace d'exécution, l'ingénieur système est en mesure de valider la spécification du procédé.

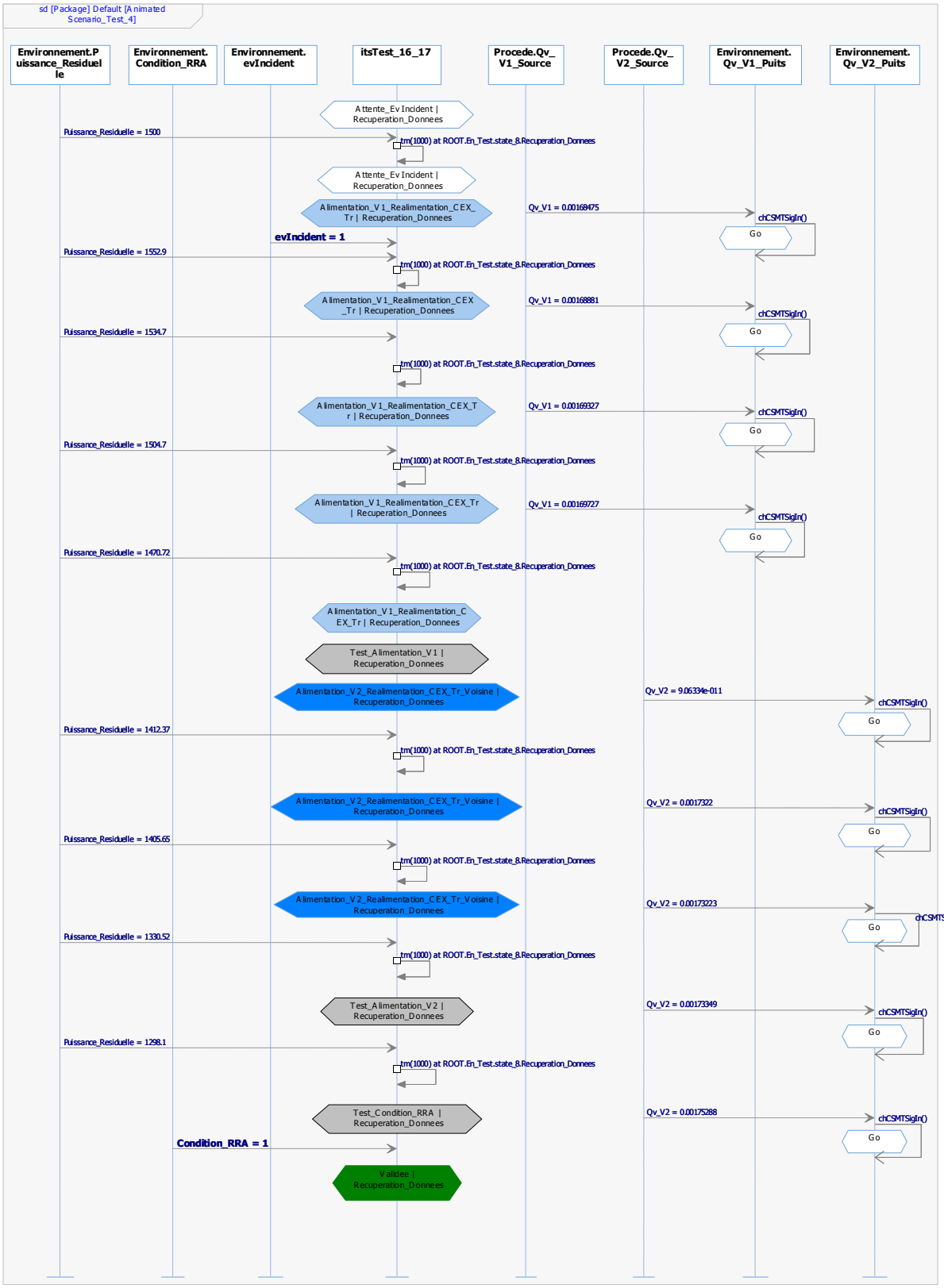


Figure 100 : Trace d'exécution du « *TestCase_1.6_1.7* » en co-simulation pour s'assurer de la satisfaction des exigences ($R_{IS1.6}$ et $R_{IS1.7}$)

5.4. Conclusion : Co-Spécification exécutable d'un modèle partiel constitutif de la partie physique ciblée

L'objectif de ce chapitre était de définir une interaction physique constitutive de la plate-forme CISPI dans notre environnement de co-spécification système en étant guidé dans un premier temps par la triade « ***Function, Structure, Process*** ». La co-spécification SysML résultante s'est avérée insuffisante pour rendre compte des couplages des phénomènes physiques en jeu pour l'interaction de modulation de débit d'eau visée.

Le raffinement de cette co-spécification en termes de triade « ***Transformation, Input, Output***» avec l'ingénieur du procédé nous a amenés à considérer l'aspect multi-physique de la modélisation de cette interaction avant sa traduction SysML pour son intégration système.

Cette connaissance (K_{h_Pro}) multi-physiques du procédé à conduire nous parait essentielle pour construire un premier modèle total (M_{H_SysML}) du système étudié, notamment pour mieux satisfaire certaines exigences de sûreté ($R_{IS_Pro_SysML}$) sensibles aux couplages des phénomènes physiques siège d'interactions critiques.

D'un point de vue conduite, le raffinement de cette co-spécification système se poursuit selon la triade « ***Communication, Control, Command*** » afin de maitriser, d'un point de vue cybernétique, le comportement de l'interaction ciblée.

Partie 3 : Co-spécification exécutable de constituants de conduite du procédé du système CISPI

6. Chapitre 6 : Co-spécification exécutable d'un constituant logique du système de conduite CISPI

6.1. Introduction : Interaction logique ciblée du système CISPI

Notre intérêt dans ce chapitre se porte sur la partie logique du modèle global de conduite de la plate-forme CISPI, correspondant au contrôle-commande du procédé, dont nous co-spécifions un modèle M_{hL} selon la triade « **Communication, Control, Command** » du « conceptagon » et le patron architectural CMMS-IAMS retenu (Chapitre 4). Cela nous amène à architecturer deux modèles constituant le modèle logique $\boldsymbol{M_{hL}}$ requérant des connaissances relatives à différents sous-domaines spécialistes d'ingénierie d'automatisation.

Dans un premier temps, notre étude se focalise sur la co-spécification d'un modèle de contrôle (***Control***) entre l'**espace-problème** d'ingénierie système et un **espace-solution** d'ingénierie d'automatisation spécialisée dans le contrôle de processus continus. Les principales fonctions du sous-système de contrôle concernent des traitements déterministes cycliques (ou automatismes) (Després, 1991) constitués d'opérations simples exécutées cycliquement et dont les performances temporelles et de sécurité sont adaptés à l'inertie et aux conditions de fonctionnement du procédé à contrôler. Plus précisément nous nous intéressons aux automatismes décisionnels de régulation dont les grandeurs réglantes ont été identifiées durant la définition du processus CISPI en chapitre 5. En ce sens, le contrôle de la plate-forme CISPI observe et agit sur le processus CISPI au travers des grandeurs réglantes de modulation en puissance (*Mod_Puissance*) définies au niveau des processus de dissipation de type C (Tableau 12), afin de maintenir une puissance d'alimentation en eau de secours suffisante pour assurer la capacité d'extraction de chaleur requis par la commande.

Une première itération de co-spécification permet alors à l'**espace-solution** d'ingénierie d'automatisation d'évaluer deux solutions : PID (Proportionnel, Intégrateur, Dérivé) et CSM (Commande Sans Modèle) (Fliess & Join, 2013) en regard des exigences systèmes requises pour la spécification de la fonction de contrôle (Tableau 15). Cette évaluation se base sur la connaissance théorique de l'ingénieur d'automatisation mais également sur des simulations ainsi que des retours d'expériences (Join, Fliess, 2014).

De façon plus détaillée, la fonction de régulation intelligente doit pouvoir s'intégrer à l'architecture du CMMS-IAMS en fournissant des informations relatives à la dérive du comportement du système d'un point de vue global, reliant ainsi le sous-système de maintenance, au sous-système de contrôle-commande à travers le sous-système de gestion technique ($\boldsymbol{R_{IS_Aut\text{-}Cont1.3}}$). Cette fonction de régulation doit être suffisamment générique pour que l'ingénieur-système puisse, en phase de conception de l'architecture organique, se baser sur des critères objectifs et mesurables de performances et de sûreté afin d'allouer cette fonction aussi bien à un agent techniques qu'à un agent humain ($\boldsymbol{R_{IS_Aut\text{-}Cont1.1}}$). Nous détaillons cette étape d'ingénierie en partie 4 de notre mémoire. Enfin, en phase d'exploration du problème, la mise en œuvre de cet algorithme de régulation intelligent doit être aisée, afin de pouvoir prototyper rapidement les solutions envisagées, les vérifier et les valider par co-exécution de modèles, et ceci quel que soit les situations de conduites envisagées ($\boldsymbol{R_{IS_Aut\text{-}Cont1.2}}$).

Cette première itération entre les domaines d'ingénierie système et d'ingénierie d'automatisation conduit l'**espace-problème** d'ingénierie système à retenir la solution de la CSM. En effet, la CSM offre des perspectives systémiques et d'intégration CMMS que les correcteurs PID classiques n'apportent pas.

Exigences	Solution 1 : PID	Solution 2 : CSM	*Commentaires*
$R_{IS_Aut\text{-}Cont1.3}$ *(Défaut global)*			A notre connaissance, les correcteurs PID n'offrent pas de moyens permettant d'évaluer la situation globale du procédé sous contrôle. En revanche de par sa formulation, la CSM permet de réguler le procédé à conduire en adaptant son comportement vis-à-vis des variations de son environnement, tout en fournissant une information relative à la situation globale de conduite (procédé + environnement) pouvant servir à détecter un défaut système. (*Ce point est détaillé dans la suite du mémoire)*
$R_{IS_Aut\text{-}Cont1.2}$ *(Temps de configuration)*			Les correcteurs PID, tout comme la CSM, ne requièrent pas une connaissance précise du modèle du procédé à conduire. Il existe des méthodes de réglage euristique pour les PID qui se fondent sur une identification réalisée à partir de relevés graphiques ou plus avancée, numérique. Cependant, le paramétrage précis d'un PID demande une connaissance approfondie aussi bien du fonctionnement du contrôleur que du procédé contrôlé (Boes, 2014). Cette étape de modélisation précise du procédé, qu'elle soit basée sur des lois physiques ou des procédures d'indentification, s'avère être fastidieuse et chronophage (évaluée par les industriels à plus de 80% du temps nécessaires à la conception). Au contraire, le réglage des coefficients de la CSM est facilité puisqu'elle réduit la commande du système à celle d'un simple intégrateur dont les propriétés sont bien connues. En ce sens, la CSM permet de s'affranchir de l'étape de modélisation du système et par conséquent de gagner du temps dans la mise en œuvre de l'algorithme de régulation. De plus, la CSM s'adapte aux modifications du procédé au cours de son exploitation (notamment liés aux phénomènes de vieillissement), éliminant ainsi les étapes de recalibrage pouvant être requises pour un correcteur PID.
$R_{IS_Aut\text{-}Cont1.1}$ *(Généricité)*			La spécification à un niveau d'abstraction fonctionnel des correcteurs PID aussi bien que la CSM ne présagent pas d'une répartition sur des agents techniques ou humains. De plus ces algorithmes de régulation génériques peuvent être appliqués à de nombreux domaines : • PID (le plus utilisé en industrie) : thermique (Cott & Macchietto, 1989), pneumatique (Hamdan & Gao, 2000), électrique (Nasri, Nezamabadi-Pour, & Maghfoori, 2007), chimie (Carot, 1994)… • CSM : électrotechnique (Michel, Join, Fliess, Sicard, & Chériti, 2010), barrage hydroélectrique (Join, Robert, & Fliess, 2010), transport (Fliess, Abouaissa, Iordanova, & Join, 2012), magnétisme (De Miras, Riachy, Fliess, Join, & Bonnet, 2012), mécanique (Vanne EGR) (Join, Masse, & Fliess, 2008)

Tableau 15 : Matrice d'analyse qualitative entre exigences ($R_{IS_Aut\text{-}Cont}$) et solutions (PID et CSM) prescrite par l'espace-solution d'ingénierie d'automatisation

Dans un second temps, notre étude se focalise sur la co-spécification d'un modèle de commande (***Command***) entre l'**espace-problème** d'ingénierie système et un **espace-solution** d'ingénierie d'automatisation spécialisé dans les systèmes à évènements discrets. Les principales fonctions de commande concernent le

séquencement[55] d'actions, ainsi que l'élaboration de rapports (automatismes informationnels) à destination d'opérateurs ou d'autres fonctions de conduite. A l'échelle de notre cas d'utilisation, le modèle de commande de la plate-forme CISPI observe le générateur de vapeur pour définir la stratégie de conduite (*Alimentation_Voie1 ou Alimentation_Voie2*) et les consignes de contrôle permettant de maintenir une capacité d'extraction de chaleur suffisante afin d'évacuer la puissance résiduelle du circuit primaire en situation incidentelle ou accidentelle.

L'**espace-problème** d'ingénierie-système est alors responsable de l'architecture des modèles de spécifications de contrôle et de commande selon le patron architectural CMMS, afin de prescrire à l'**espace-solution** d'ingénierie système une solution de la partie logique de la plate-forme CISPI.

6.2. Co-Spécification système d'un modèle exécutable constitutif d'une interaction de contrôle de modulation de débit d'eau

Il nous a semblé opportun d'intégrer, à ce niveau d'abstraction fonctionnelle, les travaux de (Fliess & Join, 2013) relatifs à la commande sans modèle (CSM) pour co-spécifier un comportement adaptatif d'une fonction de contrôle aux variations mesurables du processus à conduire. De par sa formulation, la CSM ne requiert pas d'identification préalable détaillée de la structure organique du sous-système opérant à contrôler. De plus elle fournit une information relative à la détection d'une dérive du comportement satisfaisant notre exigence architecturale CMMS/IAMS ($\boldsymbol{R_{ISAut1\text{-}1}}$).

6.2.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction de contrôle de modulation de débit d'eau

Afin de définir un algorithme de régulation en puissance d'alimentation en eau de secours, **l'espace-problème** d'ingénierie système décrit un ensemble d'exigences ($\boldsymbol{R_{IS_Aut\text{-}Cont}}$) (Figure 101) à l'**espace-solution** d'ingénierie d'automatisation. Nous distinguons deux types d'exigences :

- fonctionnelles relatives à des contraintes de conception du modèle de contrôle ayant conduit à la sélection de la CSM comme solution
- opérationnelles relatives aux fonctions et performances requises par le modèle de contrôle, se concrétisant par un paramétrage de la CSM

[55] Les fonctions de séquencement (par exemple : fonction de lignage) consistent à exécuter, selon une séquence déterminée des actions élémentaires. Chacune d'elles peut être un automatisme informationnel, une commande de positionnement d'un actionneur, ou d'un groupe d'actionneurs, ou une autre séquence.

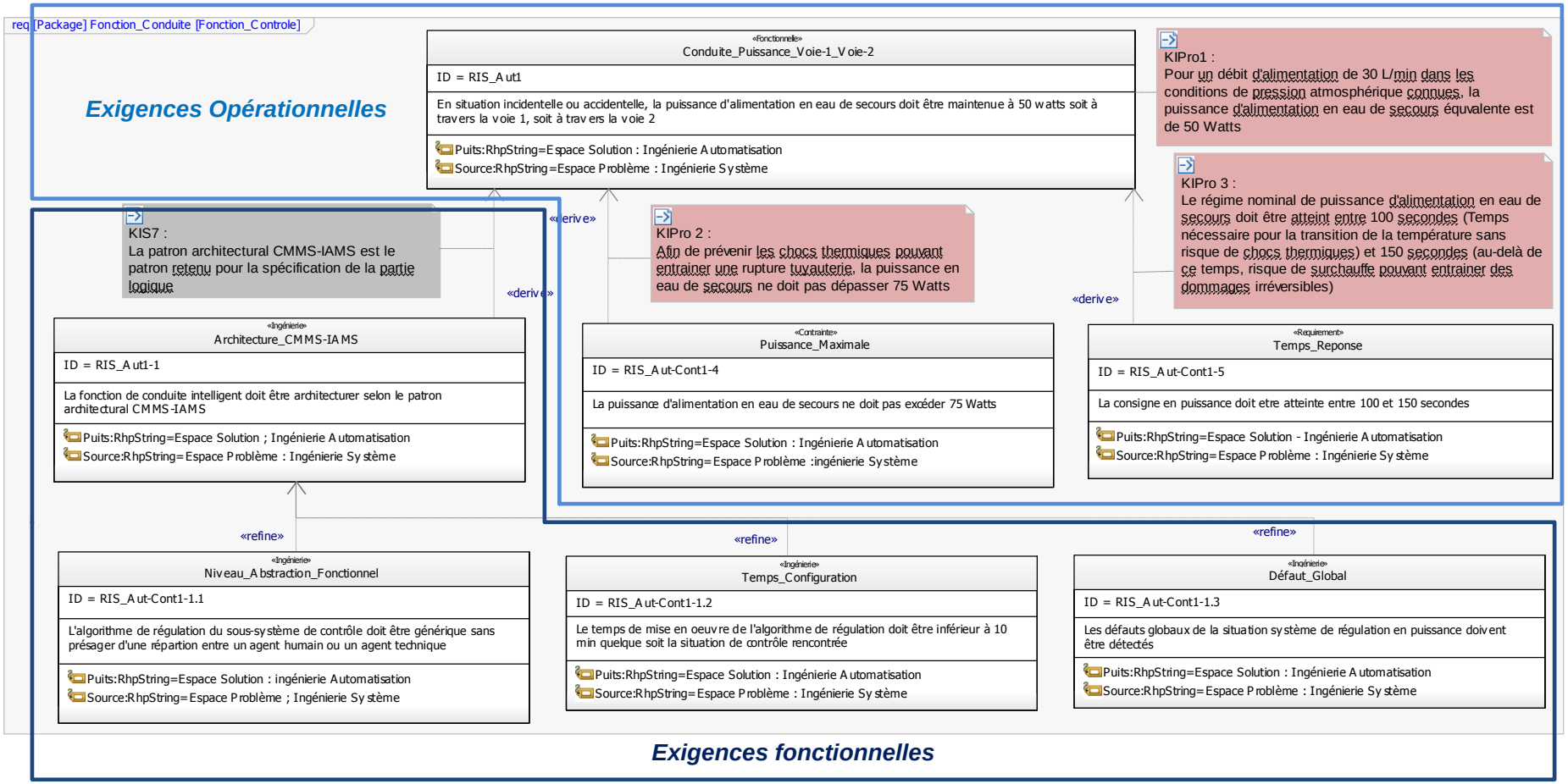


Figure 101 : Diagramme d'exigences ($R_{IS_Aut\text{-}Cont}$) relatives à la fonction de contrôle de régulation en puissance d'alimentation en eau de secours

6.2.2. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction de contrôle de modulation de débit d'eau

6.2.2.1. Eléments de connaissances d'une interaction de contrôle

La spécification d'un modèle du contrôle de la plate-forme CISPI s'appuie sur les fondements théoriques de la CSM (Commande Sans Modèles) constituant une partie de la **connaissance** (***K_{IAut}***) de l'**espace-solution** d'ingénierie d'automatisation.

Modèle ultra-local

La CSM repose sur une modélisation locale du procédé à contrôler, sans cesse réactualisée, à partir de la seule connaissance du comportement entrée/sortie. Ainsi quel que soit le procédé étudié, la CSM substitue le modèle du comportement d'entrée/sortie[56] du procédé par un modèle « phénoménologique » le plus souvent du premier ordre, et dit ultra-local car valable sur un court laps de temps (Join, 2012) (Figure 102):

$$y^{(\nu)} = F + \alpha u \tag{23}$$

Où :

- L'ordre de dérivation ($\boldsymbol{\nu}$), en général égale à 1, fixé par l'opérateur, est étranger à l'ordre de dérivation maximum de y, inconnu, dans le procédé à contrôler. Pour des cas très particuliers (*sans frottements par exemple*) l'ordre de dérivation peut être fixé à 2.
- Le paramètre constant ($\boldsymbol{\alpha}$), fixé également par l'opérateur afin que les valeurs numériques de αu et $y^{(\nu)}$ aient le même ordre de grandeur.
- Le paramètre (***F***), qui contient toutes les informations « structurelles », dépend de toutes les autres variables du procédé, y compris des perturbations et de leurs dérivées. L'estimation en temps réel de la valeur numérique de F permet la réactualisation du modèle ultra-local du procédé (23) à chaque instant.

Correcteur Proportionnel, Intégral, Dérivé intelligent (i-PID)

Ce modèle ultra-local du procédé est ensuite couplé à un correcteur PID « intelligent » afin d'obtenir le comportement désiré. Par exemple, si l'on considère l'ordre de dérivation ($\boldsymbol{\nu = 2}$), nous avons :

$$u = -\frac{F - \ddot{y}^* + K_P e + K_I \int e + K_D \dot{e}}{\alpha} \tag{24}$$

Où :

- ($\boldsymbol{y}^*$) est la trajectoire de référence de la sortie
- ($\boldsymbol{e = y - y}^*$) est l'erreur de poursuite
- ($\boldsymbol{K_P, K_I, K_D}$) sont les gains de réglage.

[56] Le modèle d'un procédé est représenté approximativement par une équation différentielle de la forme :

$$E(y, \dot{y}, \dots, y^{(a)}, u, \dot{u}, \dots, u^{(b)}) = 0$$

Notons qu'en combinant (23) et (24) et en supposant une estimation parfaite de F (*ce qui n'est jamais le cas dans la réalité*), le système étudié se ramène contrairement aux PID classiques, à un intégrateur pur du second ordre selon (25), où le paramètre (***F***) n'apparait pas, éliminant ainsi les parties inconnus et les perturbations du procédé. Les erreurs d'estimation supposées alors petites et bornées sont rejetées à l'aide du correcteur fermant ainsi la boucle de contrôle (Figure 102).

$$\ddot{e} + K_P e + K_I \int e + K_D \dot{e} = 0 \tag{25}$$

Cela s'avère être un bénéfice non négligeable pour le réglage des gains du correcteur. En effet, le système en boucle fermée est alors naturellement précis ce qui nous permet de fixer le terme intégral (***K$_I$***) à 0. Ainsi, pour les systèmes particuliers dont l'ordre de dérivation ($\boldsymbol{\nu} = \mathbf{2}$), un correcteur PD permet de fixer la dynamique du système en boucle fermé, tandis que pour un système dont l'ordre de dérivation ($\boldsymbol{\nu} = \mathbf{1}$), un simple correcteur P suffit.

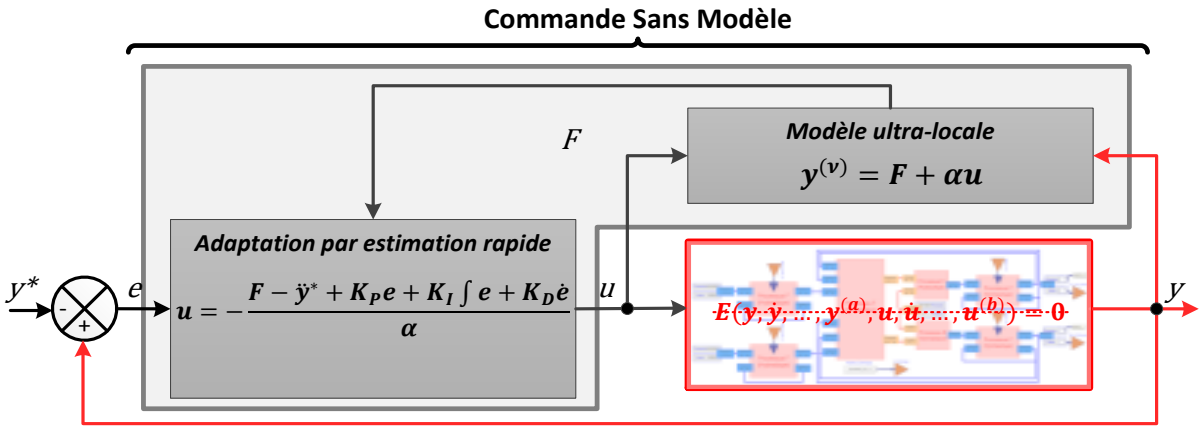


Figure 102 : Calcul de la commande en se fondant sur des outils d'estimation algébriques rapides (modèle ultra-locale) et non asymptotiques, sans démarche d'identification classique et décorrélée du procédé à contrôler.

Interprétation de la CSM selon le paradigme architectural du CMMS

Nos travaux de thèse nous ont conduits à interpréter la CSM selon le paradigme architectural du CMMS, en reliant les sous-systèmes de contrôle-commande, de maintenance et de gestion technique à travers le paramètre (***F***) de la CSM.

En effet, le paramètre (***F***) de la CSM, réajusté en permanence, représente localement la différence entre le procédé sous contrôle et son modèle ultra-local. Il intègre ainsi l'ensemble des variations aussi bien du procédé sous contrôle que des perturbations extérieures.

> « *Dans le terme estimé en ligne de la CSM se trouve l'ensemble des erreurs de modèles, les perturbations, les défauts …* » (Esteban, Pascal, & Esteve, 2009)

En étudiant la dynamique du paramètre (***F***) de la CSM, il est alors possible de détecter l'apparition d'un défaut au niveau du procédé, ou bien d'une perturbation extérieure. Une fois le système en régime permanent et sans changement de consigne, l'apparition d'une discontinuité dans le signal dérivée du paramètre (***F***), traduit une

défaillance du procédé sous contrôle, ou bien une perturbation extérieure soudaine. Notons cependant, que cette détection d'un défaut se limite à un changement violent au niveau du système étudié. En effet lors d'une dérive lente du procédé (usure) ou de l'environnement, les variations dans la dynamique du paramètre (***F***) ne sont pas suffisamment significatives pour être détectées et sont alors intégrés à chaque réajustement du paramètre.

Ainsi d'un point de vue conduite, le paramètre (***F***) est utilisé par l'algorithme de régulation, tandis que d'un point de vue maintenance il fournit une information pour la détection d'un défaut. Contextualisé dans le cadre architectural du CMMS, les valeurs prises par le paramètre (***F***) au niveau du sous-système de contrôle sont enregistrées dans le sous-système de gestion technique, pour que le sous-système de maintenance puisse calculer sa dérivée et mettre en œuvre un algorithme de détection de discontinuité du signal, afin de détecter un défaut dans le comportement du procédé ou d'une perturbation de l'environnement (Figure 103).

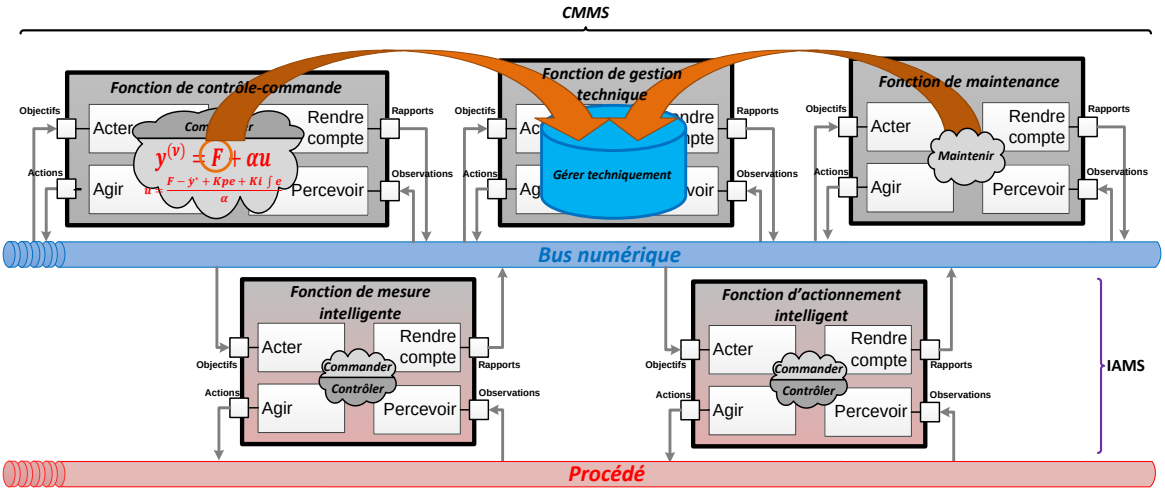


Figure 103 : Mise en œuvre systémique de la CSM selon le paradigme architectural CMMS

6.2.2.2. Construction d'un modèle exécutable de contrôle de l'interaction ciblée

Nous illustrons la mise en œuvre de la CSM pour la co-spécification d'une fonction de contrôle relative à la régulation en puissance de l'alimentation en eau de secours, afin de satisfaire les exigences opérationnelles $\{R_{IS_Aut1}, R_{IS_Aut\text{-}Cont1\text{-}4}, R_{IS_Aut\text{-}Cont1\text{-}5}\}$. Nous illustrons également l'étude de la dynamique du paramètre F pour la détection de défauts au niveau de la situation système de conduite.

En situation accidentelle ou incidentelle, la finalité de la plate-forme CISPI est de maintenir une capacité d'extraction de chaleur afin d'évacuer la puissance résiduelle du circuit primaire, jusqu'à ce que les conditions de mise en service du système RRA soient atteintes. Cette capacité d'extraction de chaleur, se traduit d'un point de vue physique par une puissance de refroidissement (d'alimentation en eau de secours) qu'il est nécessaire de maintenir au cours du temps, afin de satisfaire la finalité du système.

D'un point de vue contrôle, le maintien de la puissance d'alimentation en eau de secours est assuré par une fonction de contrôle vue au travers de la boucle de rétroaction en puissance entre la partie logique et la partie physique de la plate-forme CISPI. Cette boucle vise à « ***Agir*** » et « ***Percevoir*** » pour maintenir le procédé dans un état donné en fonction de phénomènes émergeants, et pour satisfaire les rétroactions requises par la

commande (Figure 104). L'«intelligence» embarquée par la fonction de contrôle se concrétise d'une part par l'algorithme de régulation intelligent CSM implanté au cœur de la fonction, et d'autre part par des règles de filtrage des rétroactions dans le cas de consignes ou de mesures incohérentes par rapport à la situation de conduite. Ces règles de filtrage peuvent être très simple (*plage de fonctionnement autorisée*) ou plus complexe avec la prise en compte de la dynamique du procédé.

Notons qu'à un niveau fonctionnel, nous considérons l'interaction physico-logique comme un opérateur de nature transformant la puissance « physique » observée en une donnée logique interprétable par la CSM. De la même manière, la consigne logique d'actionnement est transformée en une action physique modulant la puissance en sortie de la voie 1 pour ce cas précis.

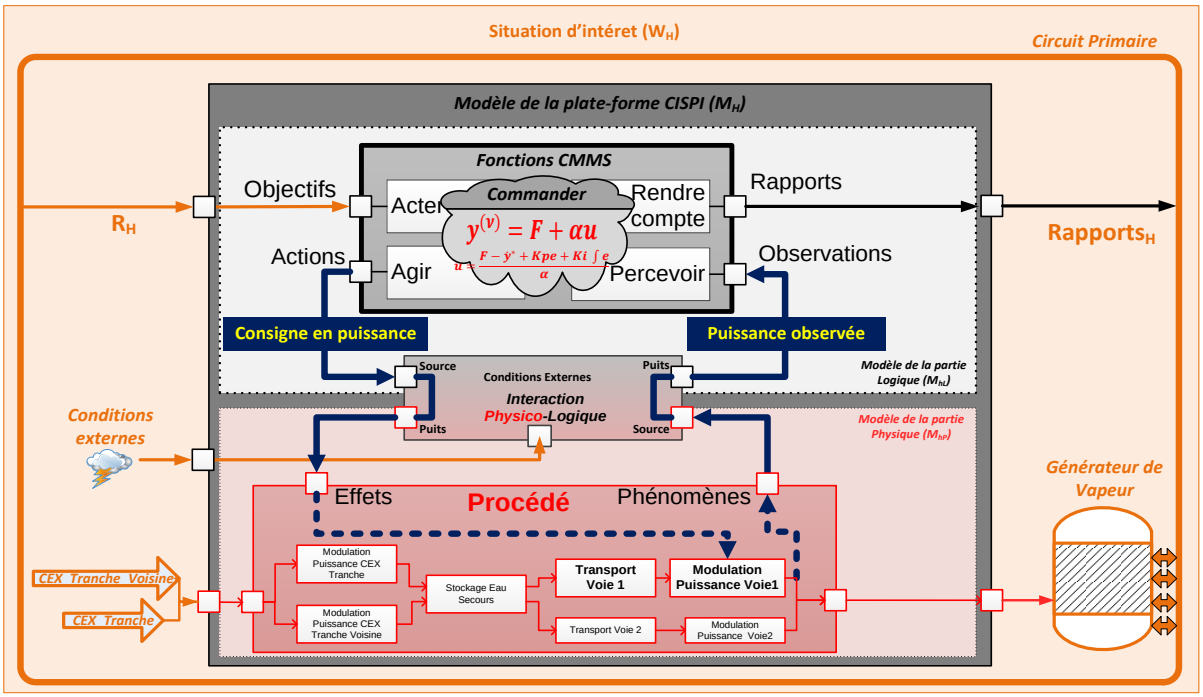


Figure 104 : Boucle de rétroaction en puissance contrôlée par une fonction de conduite intelligente intégrant la CSM

Ainsi, en se basant sur l'ensemble de ses connaissances (K_{IAut}), sur les exigences systèmes (R_{IS_Aut}) et sur le patron architectural d'une fonction de conduite, l'**espace-solution** d'ingénierie d'automatisation spécifie dans un premier temps la fonction de régulation avec l'outil spécialiste Simulink®[57] sous la forme d'un diagramme fonctionnel hiérarchique (Figure 105). L'environnement Simulink® propose un éditeur graphique ainsi qu'un ensemble personnalisable de bibliothèques de blocs et de solveurs pour la modélisation multidomaines et la simulation de systèmes dynamiques.

[57] http://fr.mathworks.com/products/simulink/

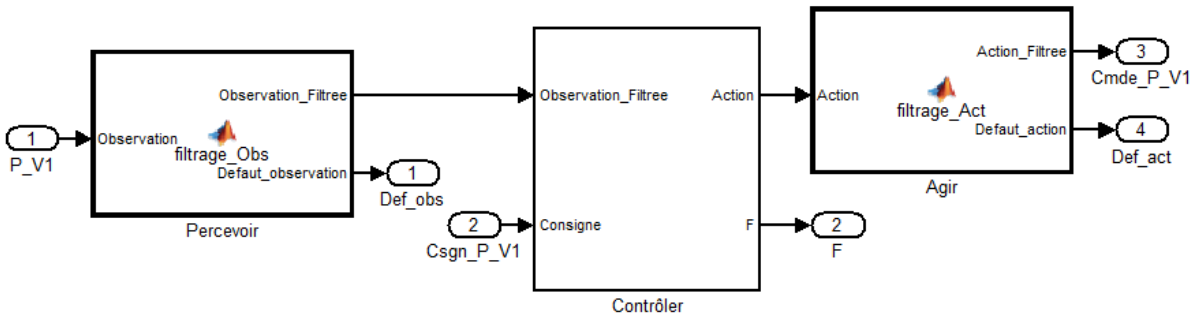


Figure 105 : Diagramme fonctionnel hiérarchique Simulink ® d'une fonction de contrôle
Application à la régulation de la puissance d'alimentation en eau de secours à travers la voie 1 de la plate-forme CISPI

De façon plus détaillée, ce diagramme fonctionnel hiérarchique se compose de deux blocs fonctionnels « ***Percevoir*** » et « ***Agir*** » contenant des algorithmes Matlab® décrivant les opérations de filtrage des observations (Figure 106a), et des actions (Figure 106c), et d'un bloc fonctionnel « ***Contrôler*** » implémentant l'algorithme de régulation intelligente (CSM) (Figure 106b). Selon les accords de confidentialité signée avec la société AL.I.E.N[58] (représentant le domaine d'ingénierie d'automatisation) nous ne divulguons pas dans ce mémoire le contenu du bloc CSM.

```
function
[Observation_Filtree,
Defaut_observation] =
filtrage_Obs(Observat
ion)
%#codegen
Puissance_Min = 25;
Puissance_Max = 75;
if (Observation >
Puissance_Min) &&
(Observation <
Puissance_Max)

Observation_Filtree =
Observation;

Defaut_observation =
false;
else

Observation_Filtree =
0;

Defaut_observation =
true;
end
```

Figure 106a : Bloc « Agir »

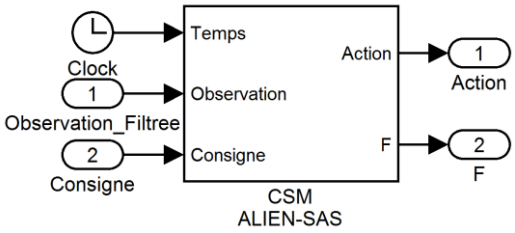


Figure 106b : Bloc « Contrôler »

```
function
[Action_Filtree,
Defaut_action] =
filtrage_Act(Action)
%#codegen
Commande_Max = 1;
Commande_Min = 0;
if (Action >=
Commande_Min) &&
(Action <=
Commande_Max)

    Action_Filtree =
Action;
    Defaut_action =
false;
else
    Action_Filtree =
0;
    Defaut_action =
true;
end
```

Figure 106c : Bloc « Percevoir »

Figure 106 : Description en vision boite blanche des blocs fonctionnels composant le modèle de contrôle de la partie logique de la plate-forme CISPI

[58] AL.I.E.N : Algèbre pour Identification et Estimation Numériques : http://alien-sas.com/csm-et-mfc/

Un moyen pour l'**espace-solution** d'ingénierie d'automatisation de vérifier que la spécification satisfasse l'ensemble des exigences systèmes consiste à coupler le modèle de la fonction de contrôle avec le modèle du procédé. Ainsi, par co-exécution de modèles l'ingénieur d'automatisation pourra s'assurer que les performances requises par le système sont respectées. Afin de définir les interfaces entre ces deux modèles, il est nécessaire de traduire la spécification de la fonction de contrôle en langage SysML pour que l'ensemble des ingénieries spécialistes puissent communiquer autour d'un langage commun. De plus, une fois ce modèle prescrit à l'**espace-problème** d'ingénierie système, ce dernier sera en mesure de valider la spécification du modèle de contrôle au regard des exigences systèmes et de l'architecture CMMS-IAMS visée.

6.2.2.3. Traduction d'un modèle exécutable de contrôle de l'interaction ciblée

A cette étape du scénario de co-spécification, l'**espace-solution** d'ingénierie d'automatisation doit traduire la spécification du modèle de contrôle en langage SysML ***($S_{IAut\text{-}Cont_SysML}$)*** afin de la prescrire à **l'espace-problème** d'ingénierie système. Notons que l'ingénieur d'automatisation ne traduit pas l'ensemble du modèle SysML, mais se focalise uniquement sur les éléments du modèle nécessaire à la validation au regard des exigences systèmes.

De nombreux travaux (Qamar, During, & Wikander, 2009), (Sindico, Di Natale, & Panci, 2011),… se sont intéressés à la transformation des modèles en langage SysML vers le langage Simulink. Parmi ces travaux, nous retenons ceux de (Cressent, Idasiak, & Kratz, 2011) qui définissent une table de correspondance entre les artefacts de modélisation Simulink et SysML (Tableau 16), et sur laquelle l'**espace-solution** d'ingénierie d'automatisation se base pour effectuer la traduction de la spécification du modèle de contrôle de la plate-forme CISPI.

Concept	Simulink	SysML
Composants	Block	Block/part
Relation	Line	Block association
Sous-composants	Subsystems	Part
Connecteurs / Flux	Inport, Outport / Line	FlowPort / FlowSpecification
Etats	Stateflow Diagram / States	State Diagram / States
Contraintes	Block	Parametric Diagram / Constraint Bloc
Relation entre contraintes	Line	Parametric Diagram / Connections
Exigences	Block	Requirement Diagram
Relations entre Exigences	Line	Requirement Diagram / Connections

Tableau 16 : Table de correspondance entre les artefacts de modélisation Simulink et SysML de (Cressent, Idasiak, & Kratz, 2011)

En ce sens, l'architecture de la fonction de contrôle est représentée dans un diagramme de définition de blocs SysML (Figure 107).

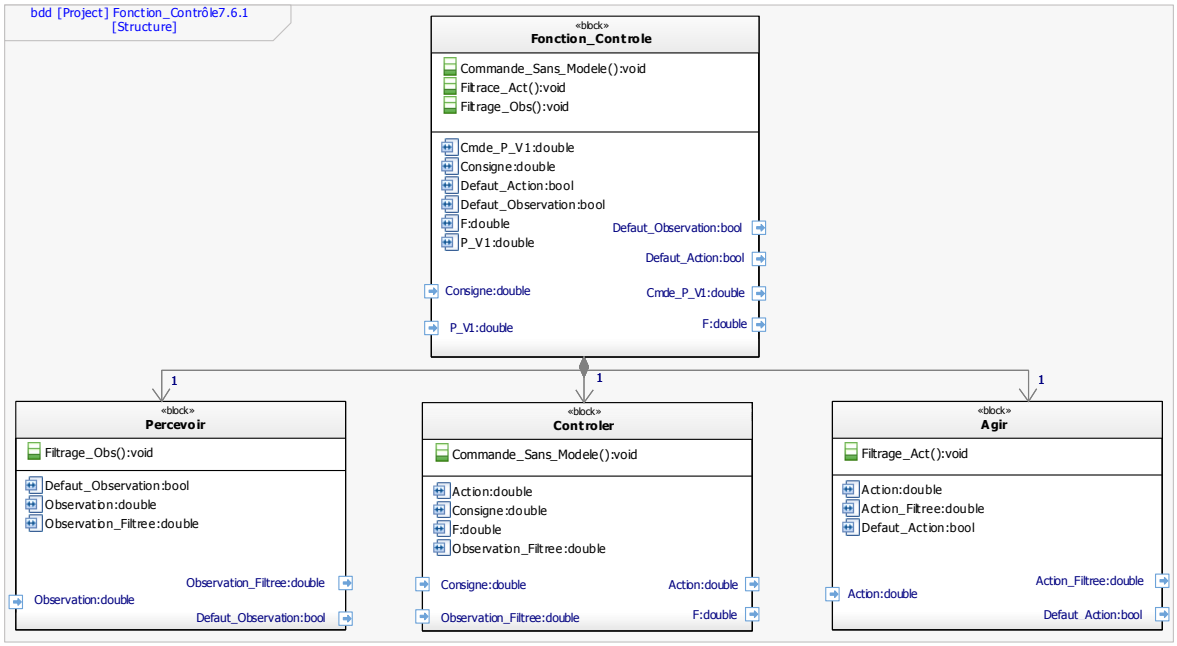


Figure 107 : Diagramme de définition de blocs (bdd) - Architecture de la fonction de contrôle

La structure hiérarchique entre le « block » Simulink de la fonction de contrôle et les sous-systèmes « sub-system » Simulink : « ***Percevoir, Controler*** et ***Agir*** », est représentée par une relation de composition SysML entre le « block » SysML de la fonction de contrôle «***Fonction_Controle***» et les « blocks » SysML : « ***Percevoir, Controler*** et ***Agir*** ».

Concernant la structure interne de la fonction de contrôle, cette dernière est représentée dans un diagramme interne de blocs SysML. Ainsi, les liens « Line » entre les ports d'entrée/sortie « Inport/Outport » des différents « blocks » Simulink sont représentés en SysML par des « Flows» entre les ports d'entrée/sortie « FlowPorts » des instances de blocs « Parts » composant la fonction de contrôle (Figure 108).

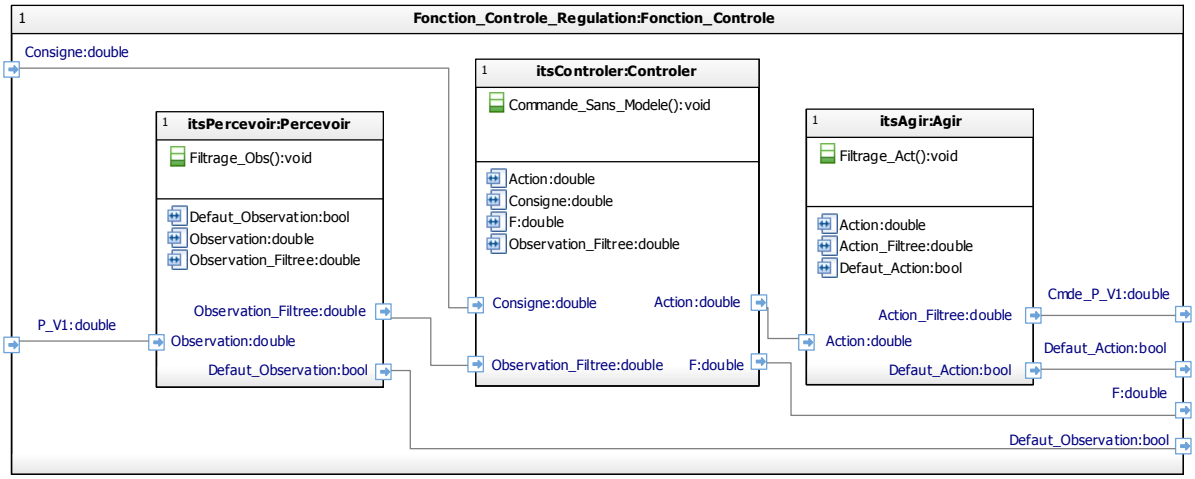


Figure 108 : Diagramme interne de bloc (ibd) - Structure interne de la fonction de contrôle

Pour des raisons de confidentialité évoquées précédemment, l'**espace-solution** d'ingénierie d'automatisation se limite à la prescription de la spécification du modèle de contrôle en SysML d'un point de vue structurel. En ce sens, l'**espace-problème** d'ingénierie système est en mesure de s'assurer de la satisfaction des exigences d'ingénierie relatives au patron architectural CMMS-IAMS retenu {$R_{IS_Aut-1-1}$, $R_{IS_Aut-Cont1-1.1}$, $R_{IS_Aut-Cont1-1.2}$, $R_{IS_Aut-Cont1-1.3}$}. D'un point de vue comportemental, la spécification de la fonction de contrôle est validée par co-exécution de modèles avec le modèle du processus de la plate-forme CISPI, permettant ainsi à l'**espace-solution** d'ingénierie d'automatisation de conserver la connaissance de son modèle dans son domaine d'ingénierie.

6.2.3. Co-intégration d'un modèle exécutable constitutif d'une interaction de contrôle de modulation de débit d'eau

La validation de la spécification ($S_{IAut_Validée}$) de la fonction de contrôle, sous la forme d'un diagramme fonctionnel hiérarchique dans l'outil Simulink est effectuée par co-exécution de modèles (Figure 109) avec la spécification du processus de la plate-forme CISPI, sous la forme d'un modèle Modelica dans l'outil Dymola (Cf. Annexe 6-1). Cette co-simulation permet entre autres de s'assurer de la satisfaction des exigences systèmes opérationnelles {R_{IS_Aut1}, $R_{IS_Aut-Cont1-4}$, $R_{IS_Aut-Cont1-5}$}.

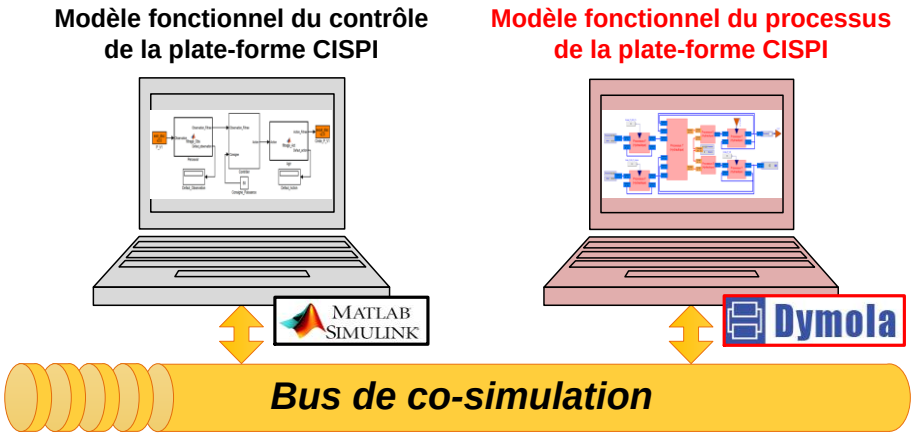


Figure 109 : Environnement de co-simulation système pour la validation de la spécification ($S_{IAut-Cont}$) d'une fonction de contrôle du processus de la plate-forme CISPI

Afin de mettre en œuvre cette co-spécification à un niveau système, l'ingénieur système architecture les modèles du modèle de contrôle et du processus de la plate-forme CISPI en définissant les liens et les données échangées entre ces modèles. Ce travail est d'autant plus facilité que les prescriptions des spécifications spécialistes sont traduites en langage SysML. En ce sens, l'ingénieur système définit dans un diagramme interne de blocs SysML les interactions entre les différents modèles spécialistes des sous-systèmes.

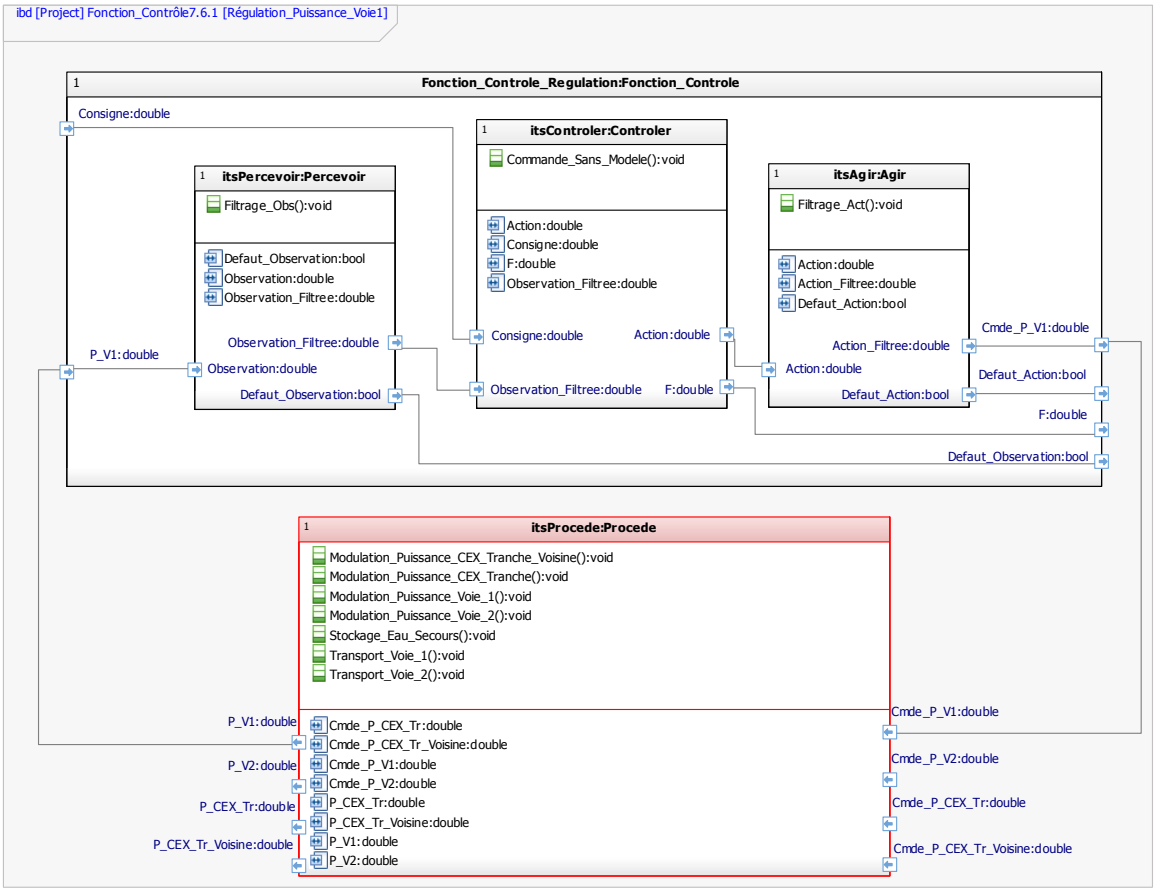


Figure 110 : Diagramme interne de blocs SysML (ibd) - Définition des interactions entre les modèles de contrôle et physique de la plate-forme CISPI pour la validation par co-exécution de modèles de la fonction de régulation en puissance à travers la voie 1

Une fois la configuration de l'environnement de co-simulation effectuée (Cf. Annexe 6-1) l'ingénieur système est en mesure de valider par co-simulation l'ensemble des exigences systèmes en analysant les traces d'exécution des modèles (Figure 111) (Figure 112) selon différents scénarios de test.

Le premier scénario de test consiste à observer le comportement de la fonction de contrôle, dans un premier temps en régime constant autour du point de consigne requis (50 Watts) puis dans un second temps autour d'un point de consigne en dehors des limites recommandées (80 Watts). L'analyse des traces d'exécution de ce scénario permet à l'ingénieur système de conclure que :

- L'exigence (R_{IS_Aut1}) est satisfaite, la fonction de contrôle arrive à maintenir la puissance d'alimentation en eau de secours à travers la voie 1 autour de la consigne de 50 Watts
- L'exigence ($R_{IS_Aut\text{-}Cont1\text{-}4}$) est satisfaite, au-delà d'une puissance observée supérieure à 75 Watts (t=380 sec), la fonction de contrôle se met en défaut, la valeur de l'observation est forcée à zéro au niveau de la fonction par conséquent la commande en sortie est au maximum, ce qui explique l'augmentation rapide de la puissance observée. Notons que sur l'apparition du défaut d'observation, la fonction de commande forcera la valeur de la consigne en puissance à 0 pour éviter toute détérioration du système.

- L'exigence (***$R_{IS_Aut\text{-}Cont1\text{-}5}$***) est satisfaite, le temps de réponse de la fonction de contrôle est d'environ de 140 secondes (compris entre 100 sec et 150 sec). Ainsi, les performances requises sont atteintes

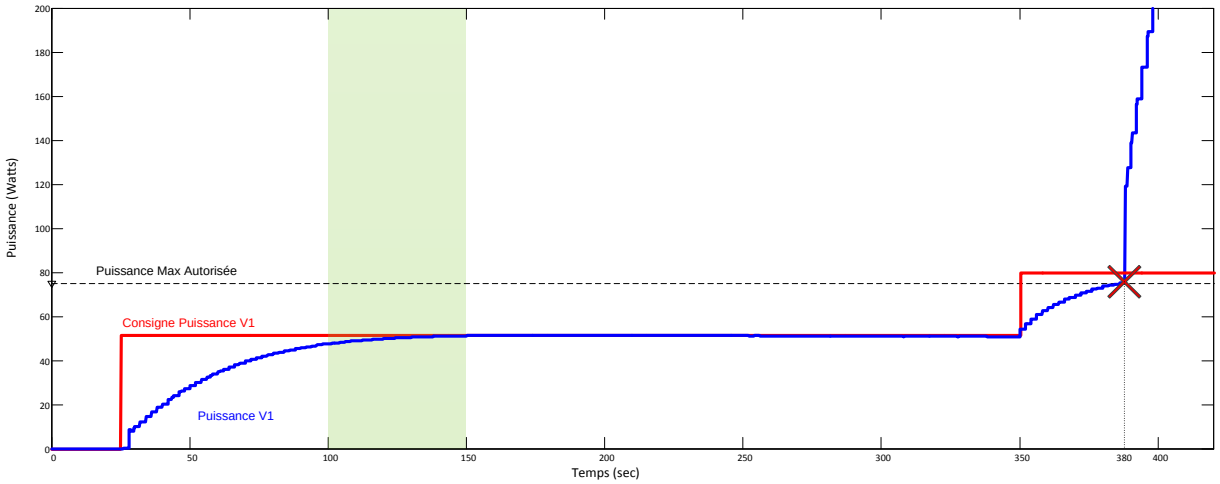


Figure 111 : Trace d'exécution modèle Simulink – Scénario test 1 (Dépassement Puissance Max)

Le second scénario de test, consiste à observer le comportement de la fonction de contrôle, et du signal dérivée du paramètre F, lors de l'apparition d'un défaut au niveau du processus sous contrôle. Ce défaut se matérialise par une fuite au niveau de la voie 1 d'alimentation en eau de secours à t = 350 secondes. D'un point de vue physique nous avons modélisé cette fuite par une « puissance de fuite » que nous soustrayons à la puissance observée. En analysant la trace d'exécution du scénario 2 **(**Figure 112**)**, l'ingénieur système peut conclure que :

- L'exigence (***$R_{IS_Aut\text{-}Cont1\text{-}1.3}$***) est satisfaite. En effet, l'analyse du signal dérivée du paramètre F permet de détecter l'apparition d'un défaut au niveau du processus de la plate-forme CISPI. Cela se traduit par une discontinuité au niveau du signal dérivé du paramètre F lors de l'apparition de ce défaut à t= 350 secondes et sans changement de consigne.

Lors de l'apparition du défaut au niveau du processus de la plate-forme CISPI, nous notons le caractère adaptatif de la commande sans modèle, qui s'adapte à la « puissance de fuite » en rectifiant immédiatement la commande en puissance au niveau de la voie 1 afin de maintenir la puissance observée autour du point de consigne requis (50 Watts).

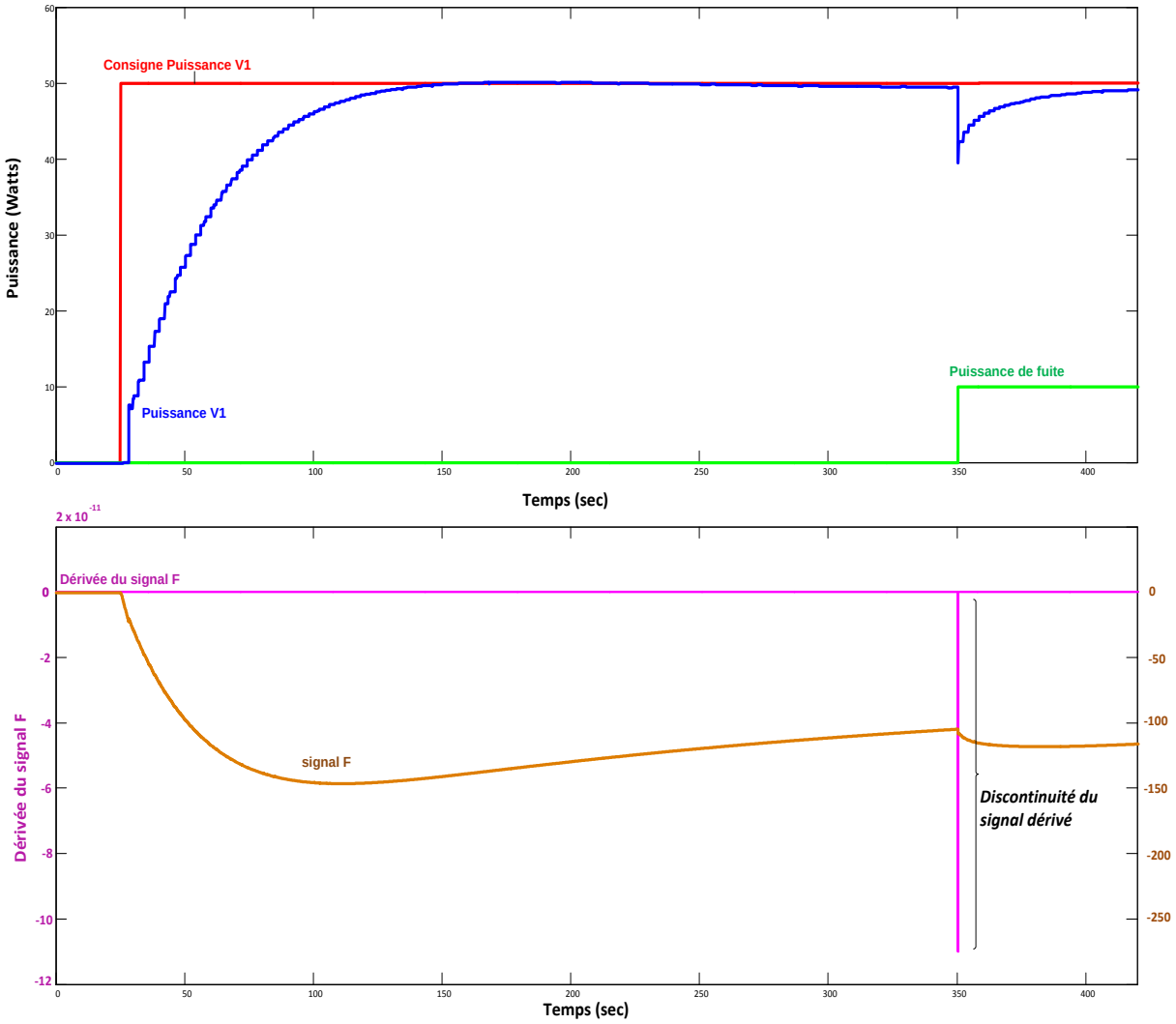


Figure 112 : Traces d'exécution modèle Simulink – Scénario test 2 (Détection Défaut Global)

6.3. Co-spécification exécutable d'un modèle constitutif d'une interaction de commande de modulation de débit d'eau

Dans ce paragraphe, notre centre d'intérêt se focalise sur la co-spécification d'un modèle de commande entre l'**espace-problème** d'ingénierie système et un **espace-solution** d'ingénierie d'automatisation. Ce modèle, spécifié selon le patron architectural CMMS-IAMS, établit les consignes de contrôle permettant de maintenir ou de faire évoluer la situation système courante de conduite, en fonction de la situation système requise par l'environnement. Il établit également des comptes rendus à destination de l'environnement, afin de s'assurer du bon déroulement des activités de conduite.

6.3.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction de commande de modulation de débit d'eau

De la même manière que pour le modèle de contrôle, le domaine d'ingénierie système requiert un ensemble de **connaissances** pour spécifier le modèle de commande de la plate-forme CISPI. Ne possédant pas ces **connaissances**, l'**espace-problème** d'ingénierie système formalise un ensemble d'exigences systèmes ($R_{IS_Aut_Comm}$) et les décrit à travers le bus de co-spécification système. Dans notre scénario de co-spécification, largement influencé par les domaines de compétences de notre laboratoire, ces exigences sont adressées à un **espace-solution** d'ingénierie d'automatisation spécialisé dans les systèmes à évènements discrets. Cet espace est à même de répondre à ces problématiques en formalisant les procédures de conduite selon une Approche Par Etat (APE). Plus précisément, nous nous intéressons à la fonction de commande d'alimentation en eau de secours à travers la voie 1 ou la voie 2 de la plate-forme CISPI (Figure 113).

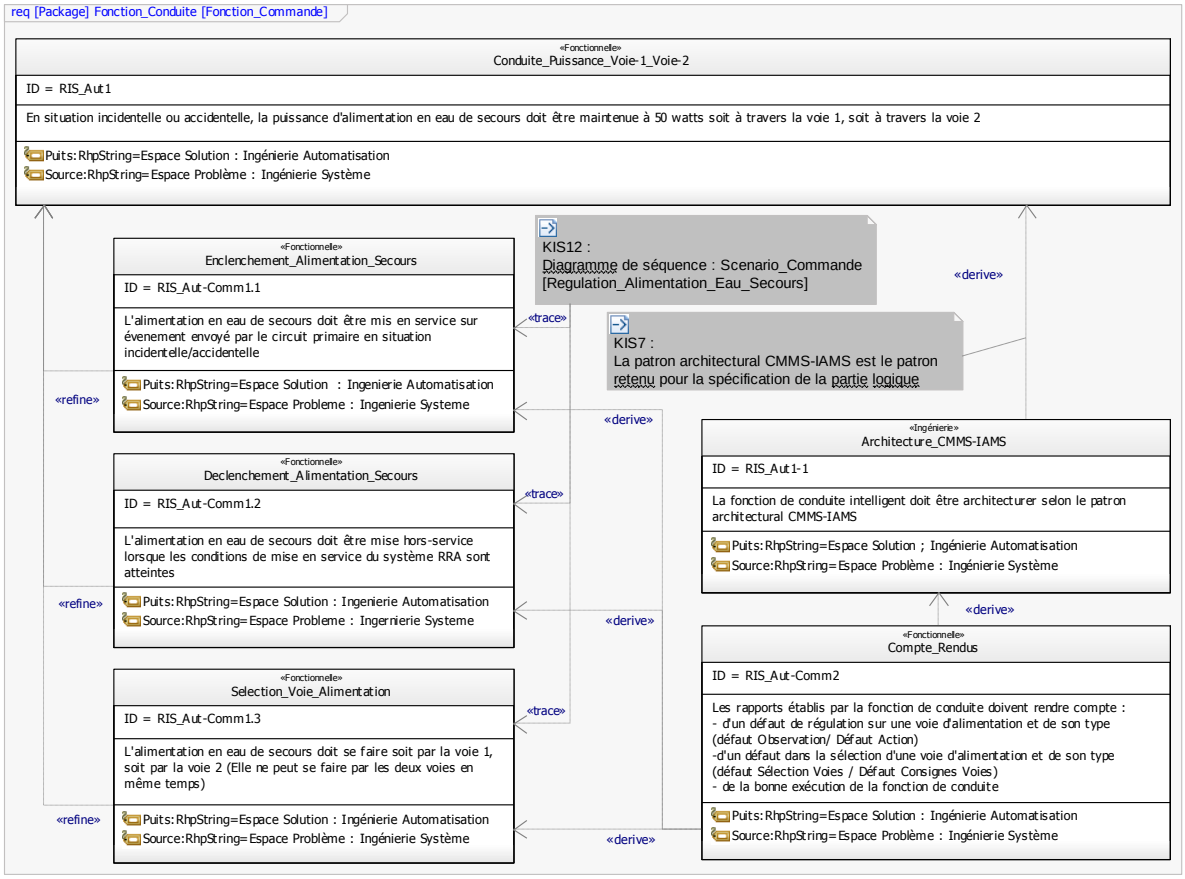


Figure 113 : Exigences ($R_{IS_Aut\text{-}Comm}$) relatives à la fonction de commande d'alimentation en eau de secours de la plate-forme CISPI

Pour une meilleure compréhension des exigences systèmes de commande, l'**espace-problème** d'ingénierie système décrit en même temps que ces exigences, un diagramme de séquence SysML formalisant le comportement attendu du système vis-à-vis de son environnement (Figure 114). A l'heure actuelle, pour de nombreux domaines industriels (aéronautique, ferroviaire, production d'énergie électrique, ….) les exigences textuelles définissent la référence contractuelle entre deux espaces problème et solution. Bien que ces

exigences puissent être illustrées par des points de vue ou des éléments du modèle du système, ce dernier ne constitue pas aujourd'hui une référence contractuelle. Le modèle est alors utilisé pour assurer la cohérence entre les différents points du vue et parties prenantes du projet.

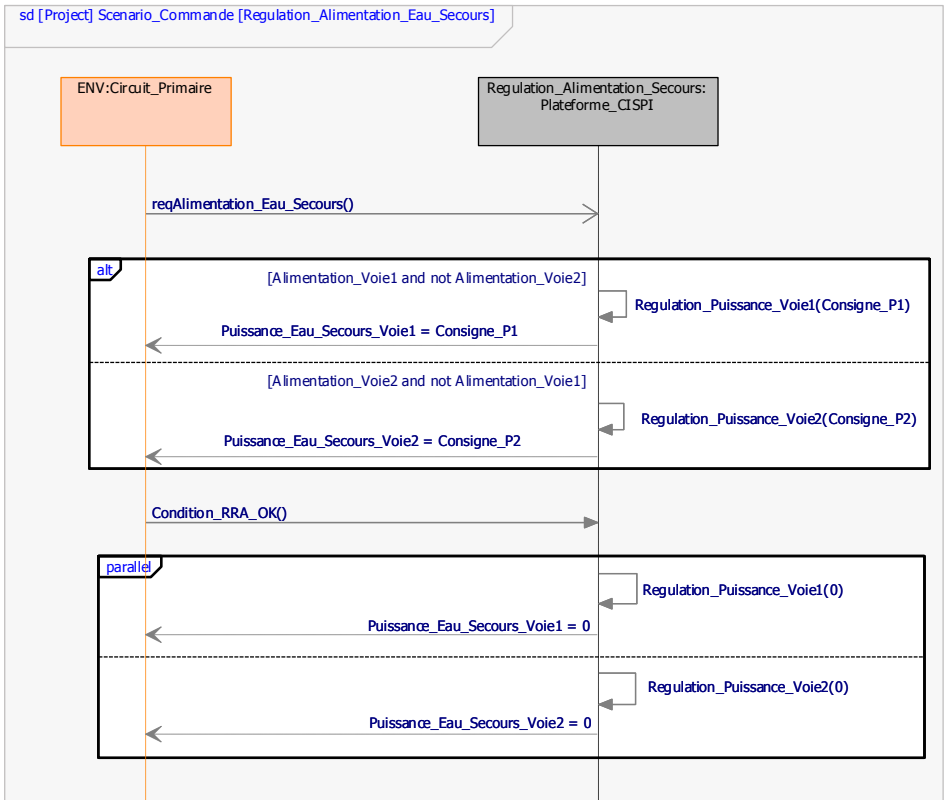


Figure 114 : Diagramme de séquence SysML – Scénario d'alimentation en eau de secours du circuit primaire par la plate-forme CISPI au travers de la voie 1 ou de la voie 2

6.3.2. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction de commande de modulation de débit d'eau

6.3.2.1. Eléments de connaissances d'une interaction de commande

Les connaissances de l'**espace-solution** d'ingénierie d'automatisation en modélisation des systèmes à évènements discrets lui permettent de spécifier directement la structure et le comportement du modèle de commande en langage SysML. En effet, le formalisme des diagrammes d'états SysML est très proche des formalismes utilisés habituellement (*Réseau de Pétri* (Peterson, 1981)*, Grafcet* (NF EN 60848, 2002)*, Automate* (Hopcroft, 1979) ,...) par l'ingénieur d'automatisation. Ces diagrammes décrivent le comportement interne d'un « *block* » selon un automate à états finis. Ils représentent ainsi les séquences possibles d'états et des opérations qu'une instance de bloc « part » peut mettre en œuvre au cours de son cycle de vie en réaction à des événements discrets. Notons que pour le type de système étudié : système critique, les diagrammes à états SysML auraient dû être formalisés en langage formel tel que le langage SCADE. Ce langage de modélisation

utilisé dans l'outil SCADE Suite[59] se base sur le langage Lustre (Halbwachs, Caspi, Raymond, & Pilaud, 1991; Houmb, Islam, Knauss, J\ürjens, & Schneider, 2010). Il offre la possibilité de représenter le modèle de contrôle-commande sous la forme de machines à états synchrones et de modèles de flots de données. De plus, il permet la vérification formelle du modèle ainsi que la génération de code certifié selon le standard IEC 61508 (Niveau SIL 3 (*Safety Integrity Level*)) dont est issu le standard IEC 61513:2011 (*Centrales nucléaires de puissances – Instrumentation et contrôle-commande importants pour la sûreté – Exigences générales pour les systèmes*).

6.3.2.2. Définition d'un modèle exécutable de commande de l'interaction ciblée

Spécification structurelle du modèle de commande

La décomposition hiérarchique de la fonction de commande telle que définie par le patron architectural CMMS-IAMS est représentée dans un diagramme de définition de blocs SysML. Ainsi, le bloc fonctionnel de commande « ***Uc_Alimenter_Eau_Secours*** » se décompose en trois sous-blocs fonctionnels : ***« Acter, Commenter*** et ***Rendre Compte***» (Figure 115). Comme pour le modèle de contrôle, la structure interne du modèle de commande est décrite dans un diagramme interne de blocs SysML (Cf. Annexe 6-2). Ce diagramme représente les messages et flux échangés entre les différentes instances des sous-blocs fonctionnels « Parts » composant le bloc fonctionnel étudié « ***Uc_Alimenter_Eau_Secours*** ».

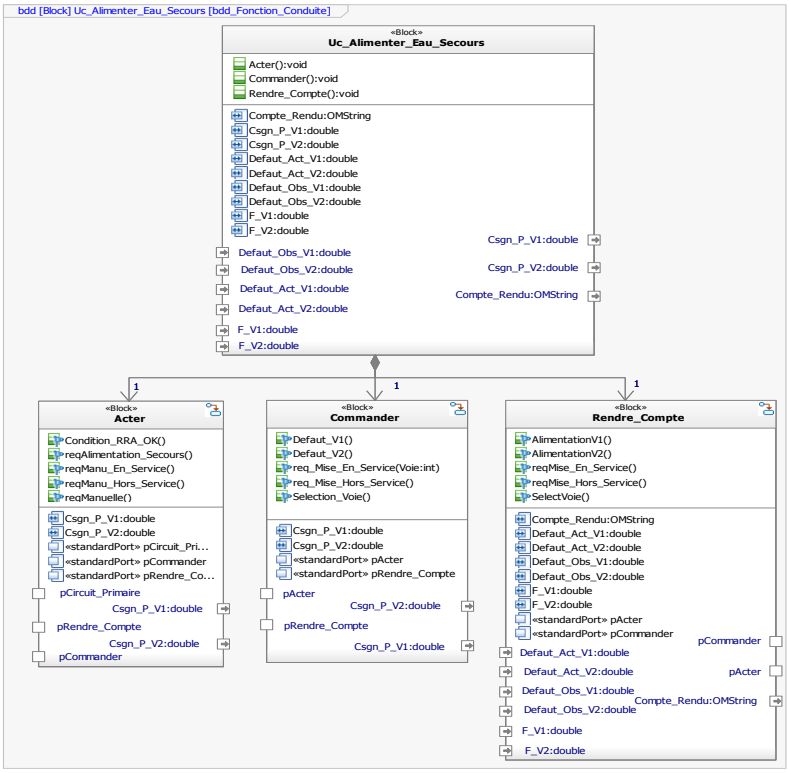


Figure 115 : Diagramme de définition de blocs (bdd) - Architecture de la fonction de commande « *Uc_Alimenter_Eau_Secours* »

[59] SCADE (Safety Critical Application Development Environment) : http://www.esterel-technologies.com/products/scade-suite/

Spécification comportementale du modèle de commande

Une fois l'architecture du modèle de commande définie, l'ingénieur d'automatisation spécifie le comportement de chacun des sous-blocs fonctionnels constituants. Ces comportements sont représentés par des diagrammes d'états SysML. De façon générale :

- le bloc fonctionnel « ***Acter*** » filtre les requêtes externes selon la situation système de conduite. De plus, il initialise la fonction de conduite (*commande + contrôle*) et gère la mise en service ou hors service de cette fonction.
- Le bloc fonctionnel « ***Commander*** » implémente des procédures (*stratégies*) de conduite à travers une approche par états (APE). Selon la situation système de conduite, ces procédures visent à maintenir le système dans un état, ou dans le cas contraire à le faire évoluer vers un état connu et maitrisable (chapitre 1).
- Le bloc fonctionnel « ***Rendre_Compte***» agrège l'ensemble des défauts de contrôle et de commande pour élaborer des comptes rendus de la situation système de conduite à destination du système extérieur. De plus, il fournit des informations au bloc fonctionnel « ***Commander*** » pour la prise en compte des défauts de contrôle.

La situation système de conduite de notre cas d'utilisation se caractérise selon une situation incidentelle/accidentelle survenant alors que le système a atteint l'état de production d'énergie électrique en pleine puissance (« *Etat de fonctionnement en puissance* »). Dans cette situation, le refroidissement du circuit primaire n'étant plus assuré par le circuit secondaire. La stratégie de conduite à adopter consiste alors à alimenter en eau de secours le circuit primaire, afin d'en évacuer la puissance résiduelle jusqu'à ce que le système atteigne un nouvel état sûr de fonctionnement : « *Arrêt intermédiaire diphasique aux conditions RRA connecté* ».

En ce sens, le circuit primaire représenté par un acteur externe SysML (***Circuit_Primaire***) émet deux évènements (***reqAlimentation_Secours***, ***Condition_RRA_OK***) au sous-bloc fonctionnel (***Acter***). La prise en compte de ces requêtes externes est modélisée en SysML par deux opérations de réception d'évènements allouées au sous-bloc fonctionnel (***Acter***). Les réceptions de ces événements définissent les conditions de mise en service ou hors service du sous-système de conduite d'alimentation en eau de secours. Ces deux états du système sont représentés par des états SysML : « ***Hors_Service*** » et « ***En_Service*** ». Notons qu'il est également possible de mettre manuellement en service « ***reqManu_En_Service*** » ou hors service « ***reqManu_Hors_Serive*** » le système de conduite. L'initialisation de la fonction de la conduite est effectuée au travers de deux opérations assignant les consignes de contrôle en puissance : « ***setCsng_P_V1(0), setCsng_P_V1(0)***» à zéro (Figure 116).

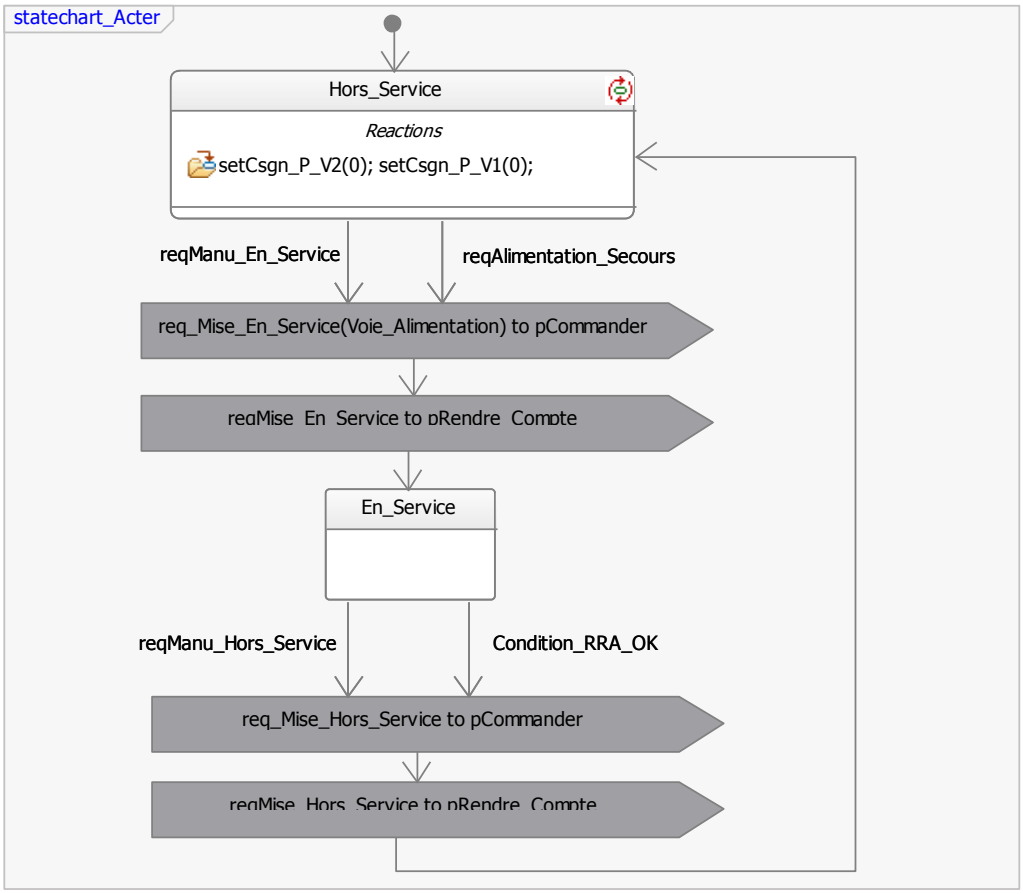


Figure 116 : Diagramme d'états SysML - Comportement du bloc fonctionnel : "*Acter*"

Les conditions de mise en service et hors service au niveau du sous-bloc fonctionnel (***Acter***) sont envoyées sous la forme d'événements (***req_Mise_En_Service, req_Mise_Hors_Service***) au sous-bloc fonctionnel (***Commander***) à travers le « standard port » (***pCommander).*** Lors de la réception de l'évènement (***req_Mise_En_Service***) les consignes de contrôle d'alimentation en eau de secours sont élaborées soit pour la voie 1 (***Alimentation_V1***) soit pour la voie 2 (***Alimentation_V2***). L'assignation et l'envoi des valeurs de la consigne en puissance (***Consigne_P***) au sous-système de contrôle sont modélisés par les opérations (***setCsgn_P_V1(Consigne_P), setCsgn_P_V2(Consigne_P)***). Notons qu'il est également possible de permuter l'alimentation en eau de secours à travers la voie 1 ou la voie 2 lors de la réception de l'évènement (***Selection_Voie***). Enfin, l'apparition d'un défaut (***Defaut_V1*** ou ***Defaut_V2***) sur l'une des voies enclenche automatiquement l'alimentation sur la voie qui n'est pas défaillante (Figure 117).

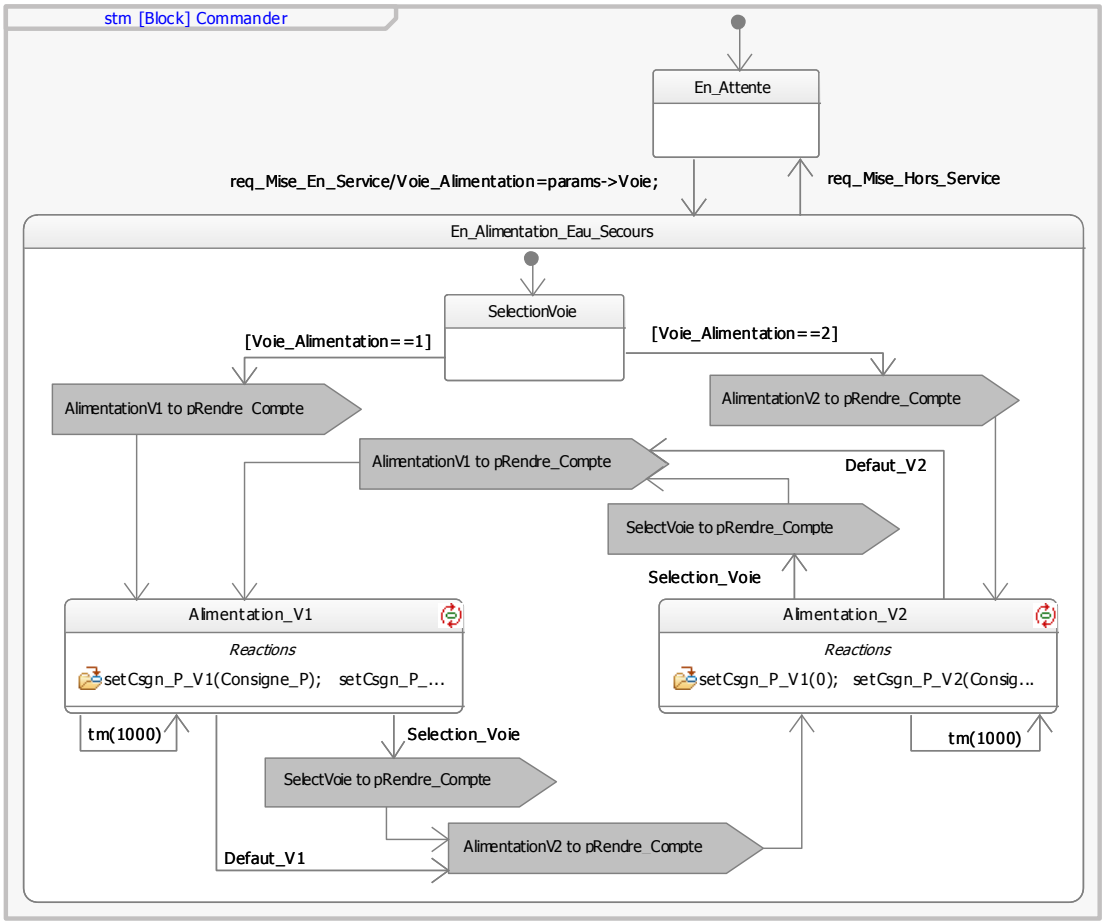


Figure 117 : Diagramme d'états SysML - Comportement du bloc fonctionnel : "*Commander*"

Pour finir, les comptes rendus relatifs à la situation système de conduite d'alimentation en eau de secours sont élaborés au niveau du bloc fonctionnel (***Rendre_Compte***). L'élaboration de ces comptes rendus dépendent de la réception des événements émis soit par la partie commande (***reqMise_En_Service***, ***reqMise_Hors_Service***, ***AlimentationV1***, ***AlimentationV2)*** soit par la partie contrôle au travers de conditions (***Defaut_Obs_V1 ==1***, ***Defaut_Obs_V2 ==1***, ***Defaut_Act_V1 ==1***, ***Defaut_Act_V2 ==1***). L'envoi des comptes rendus au système extérieur est représenté par les opérations (***setCompe_Rendu (« compte rendu »)***) définies au niveau des transitions entre les états représentant la situation système de conduite (***Alimentation_V1***, ***Alimentation_V1***, ***Alimentation_V1_Defaut_V2***, ***Alimentation_V2_Defaut_V1***) (Figure 118).

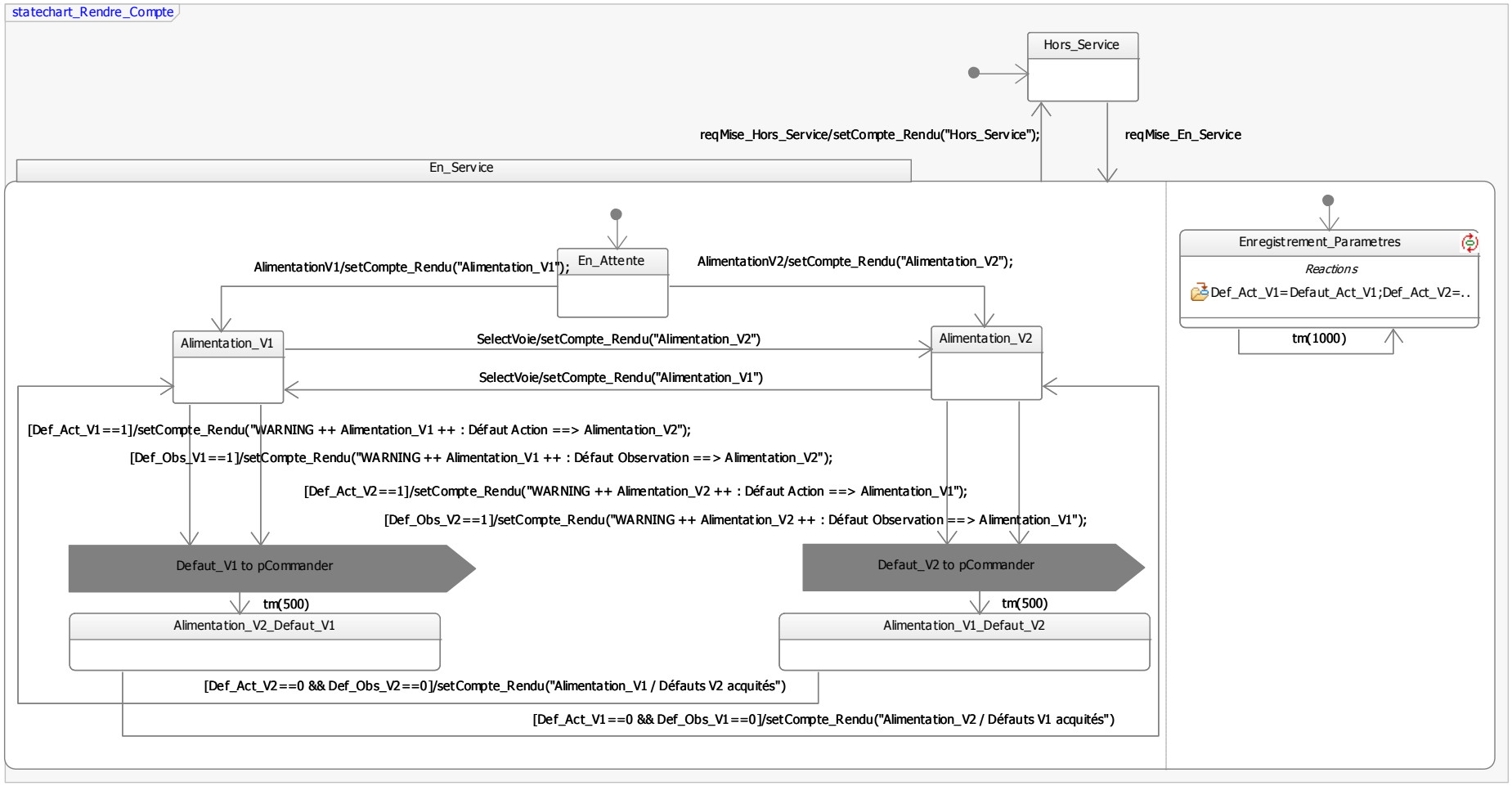
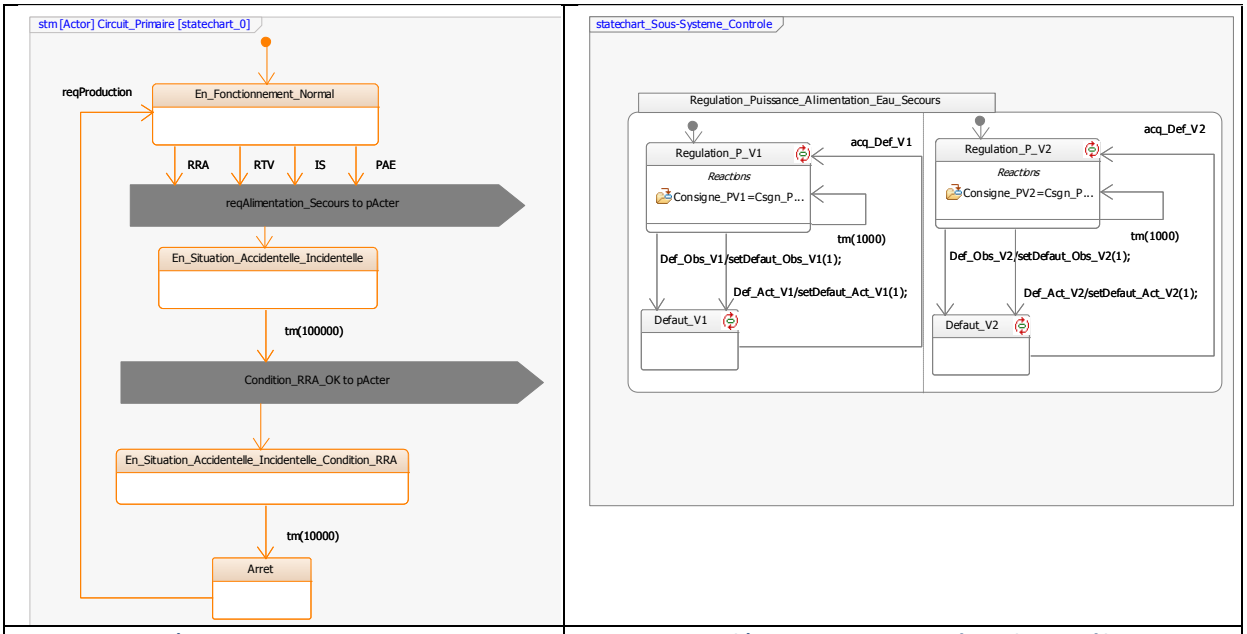


Figure 118 : Diagramme d'états SysML - Comportement du bloc fonctionnel : "*Rendre Compte*"

Vérification de la spécification d'un modèle exécutable de commande

Au cours de cette étape du scénario de co-spécification, l'**espace-solution** d'ingénierie d'automatisation vérifie que le comportement du modèle de commande se comporte tel qu'il est décrit par l'ensemble des exigences systèmes ($R_{IS_Aut\text{-}Comm}$). Une fois la spécification vérifiée, cette dernière est prescrite à l'**espace-problème** d'ingénierie système pour validation. Notons que pour la prescription de la solution, l'ingénieur d'automatisation s'affranchit de la traduction du modèle de commande étant donné que celui-ci a été directement modélisé en langage SysML.

Afin de stimuler les entrées du modèle de commande, l'ingénieur d'automatisation décrit également le comportement des acteurs externes du système (***Circuit_Primaire*** et ***Sous_Systeme_Controle***) selon des diagrammes d'états SysML (Figure 119). Les sorties du modèle de commande sont également connectées aux modèles d'acteurs afin d'observer l'évolution du comportement global du système d'un point de vue commande au regard du comportement requis.



a) Acteur : Circuit Primaire | b) Acteur : Sous-système de contrôle

Figure 119 : Diagrammes d'états SysML - Modèles comportementaux des acteurs externes du sous-système de commande

Les interactions entre les acteurs et le sous-système de commande sont représentées dans un diagramme interne de blocs de SysML (Figure 120). Ce diagramme permet de modéliser deux types d'interactions :

- Les flux de services au travers des « Standard Ports » définissent les opérations requises ou fournies par les blocs. Par exemple, le flux de service entre le port (***pActer***) et le port (***pCircuit_Primaire***) représente l'échange des requêtes de mise en service ou hors service du sous-système de conduite entre le part (***Circuit_Primaire***) et le part (***Sous_Systeme_Commande***). Ces opérations sont définies dans des interfaces, représentant pour notre cas d'utilisation des opérations de réception des évènements : ***reqAlimentation_Secours*** et ***Condition_RRA_OK.***
- Les flux de matières, énergies et informations au travers des « Flow Ports ». Dans notre cas d'utilisation, les flow ports (***F_V1, F_V2, Def_Obs_V1, Def_Obs_V2, Csgn_P_V1,…)*** représentent les données échangées entre les instances de blocs.

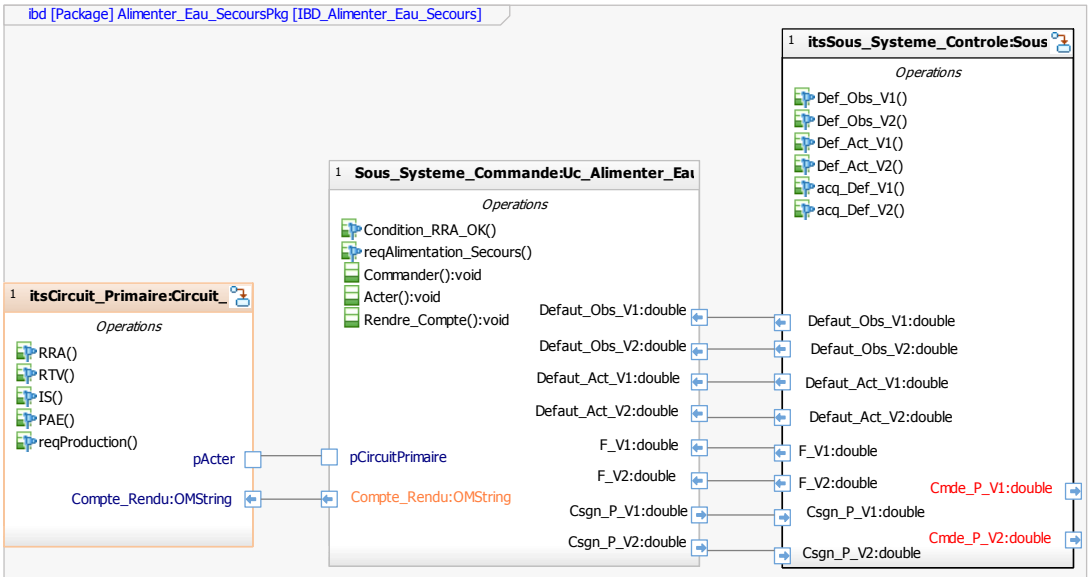
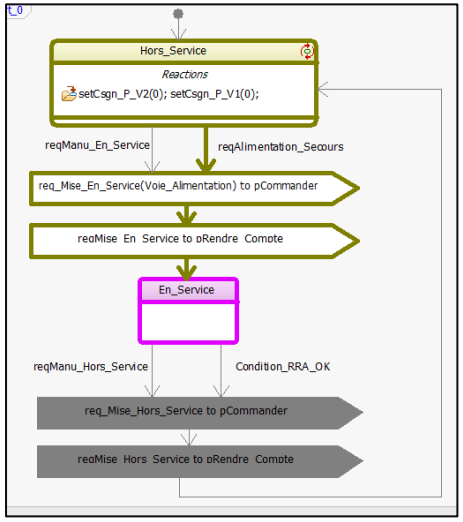


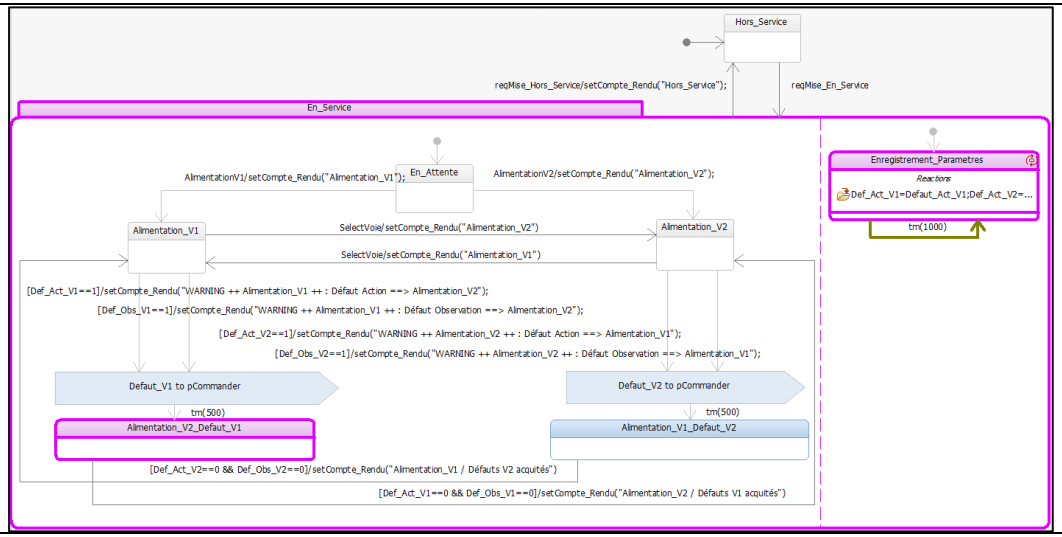
Figure 120 : Diagramme interne de blocs SysML - Interactions entre le sous-système de commande et le système extérieur (*Circuit_Primaire + Sous_Systeme_Controle*)

L'ingénieur d'automatisation est alors en mesure d'exécuter des scénarios de test pour s'assurer de la satisfaction des exigences systèmes ($R_{IS_Aut\text{-}Comm}$). Comme nous l'avons présenté précédemment, l'outil IBM® Rational® Rhapsody® permet l'exécution des diagrammes SysML pour vérifier au plus tôt le comportement du modèle de commande. Il est alors possible de récupérer les traces d'exécution des scénarios de test sous la forme de diagrammes de séquences (cf. chapitre 5), et également de visualiser en temps réel l'état du modèle de commande par animation des diagrammes d'états SysML. Par exemple, au cours de la simulation nous pouvons observer (Tableau 17) que le bloc fonctionnel (***Acter***) se situe dans l'état (***En_Service***) (Tableau 17a) tandis qu'au même instant le bloc fonctionnel (***Commander***) se situe dans l'état (***Alimentation_V2***) (Tableau 17b). Afin de faciliter les interactions avec le modèle SysML, l'ingénieur d'automatisation développe une interface sous la forme d'un « Panel Diagram » permettant de générer un ensemble d'événements stimulant les acteurs du système, ou le système lui-même, mais également d'observer un ensemble de paramètres du modèle (*valeurs des attributs, états du système*) (Tableau 17d). En déroulant un ensemble de scénario de test, l'espace-solution d'ingénierie d'automatisation est ainsi capable de vérifier le comportement du modèle de commande avant de prescrire sa solution à l'**espace-problème** d'ingénierie d'automatisation.

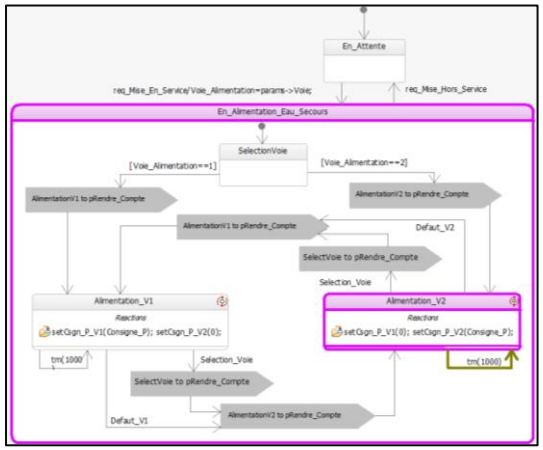
Notons cependant que cette démarche interne au domaine d'ingénierie d'automatisation ne définit qu'un premier niveau de vérification du système. En effet, il aurait été préférable que le comportement des acteurs externes ne soit pas modélisé par l'**espace-solution** d'ingénierie d'automatisation mais plutôt par des **espaces-solutions** d'ingénieries spécialistes. La modélisation de ces acteurs par l'ingénieur d'automatisation est ici fortement influencée par la solution qu'il vient de définir et ne reflète pas en ce sens leurs comportements réels. L'environnement de co-spécification système tel que nous l'avons défini offre une solution à cette problématique. Ainsi, en effectuant des requêtes aux domaines d'ingénieries spécialistes concernés par les acteurs du système, le domaine d'ingénierie d'automatisation aurait pu vérifier le comportement du sous-système de commande en l'immergeant dans un modèle de l'environnement le plus fidèle possible de la réalité. Cette alternative permet alors de gagner en niveau de maturité durant la vérification de la spécification. Nous illustrons cette démarche dans le prochain paragraphe durant la validation du sous-système de commande par l'**espace-problème** d'ingénierie système en co-spécification exécutable avec les **espaces-solutions** d'ingénierie opérationnelle, d'automatisation et du procédé.



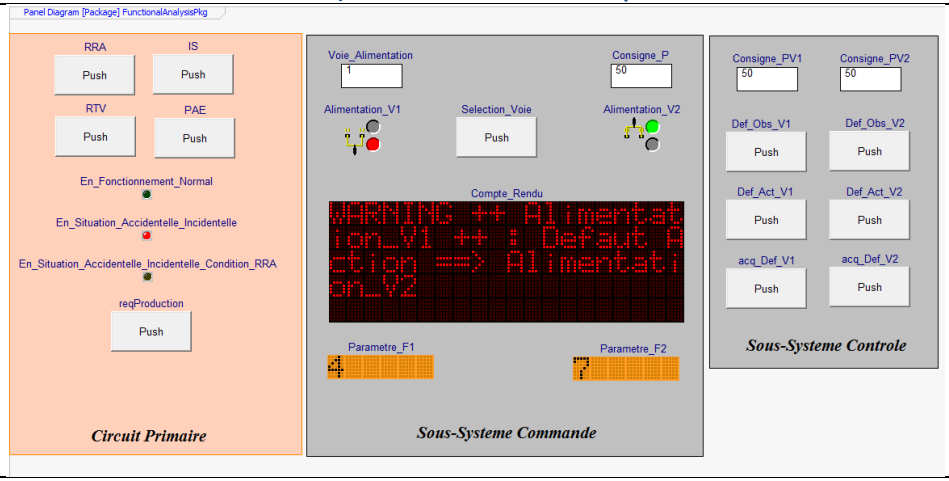
a) Statechart « Acter »



b) Statechart « Rendre Compte »



c) Statechart « Commander »



d) Panel Diagram

Tableau 17 : Vérification du sous-système de contrôle de la partie logique de la plate-forme CISPI par exécution de modèles SysML

6.3.3. Co-Intégration système d'un modèle constitutif d'une interaction de commande de modulation de débit d'eau

A cette étape du scénario de co-spécification système l'**espace-problème** d'ingénierie système valide la spécification du modèle de commande prescrit par l'**espace-solution** d'ingénierie d'automatisation. Cette activité est réalisée par co-exécution de modèles afin de s'assurer que le modèle du sous-système de commande immergé dans son environnement (*modèle de l'environnement opérationnel, du sous-système de contrôle et du sous-système physique*) manifeste les phénomènes requis par ce dernier (***R_{IO_SysML}***) (Cf. Annexe 6-3). Notons que le modèle de l'environnement prescrit par le domaine d'ingénierie opérationnelle reste le même que celui utilisé précédemment pour la validation du sous-système physique.

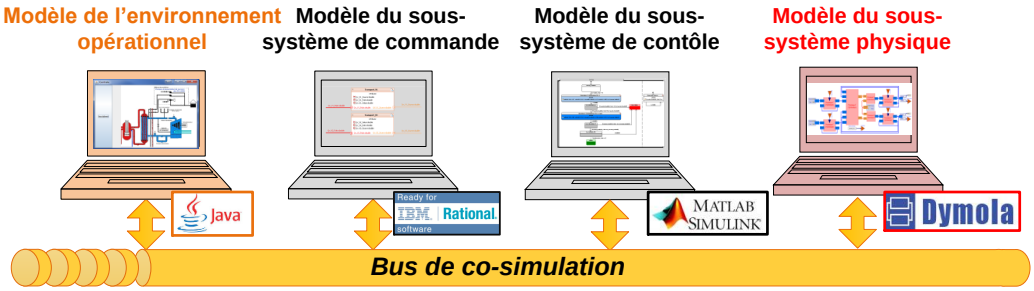


Figure 121 : Environnement de co-simulation système pour la validation de la spécification ($S_{IAut\text{-}Comm}$) d'une fonction de commande de la plate-forme CISPI

Pour ce faire, **l'espace problème** d'ingénierie système architecture l'ensemble des modèles des ingénieries spécialistes. Cette architecture est définie dans un diagramme interne de blocs SysML représentant la plate-forme CISPI selon sa partie logique et sa partie physique, en interaction avec son environnement (***Circuit_Primaire***) (Figure 122).

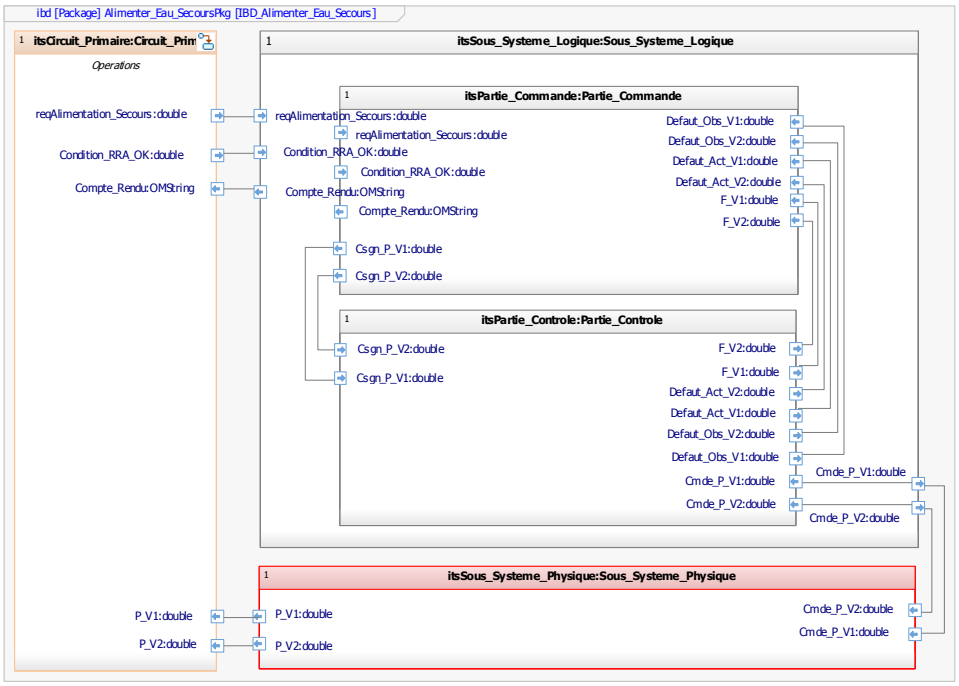


Figure 122 : Seconde itération - Définition de l'architecture de co-simulation système (*Sous-système logique + Sous-système physique*) par l'espace-problème d'ingénierie système

L'ingénieur système est alors en mesure d'exécuter des scénarios de test pour valider la spécification du sous-système de commande de la plate-forme CISPI selon l'analyse des traces d'exécution des différents modèles spécialistes (Figure 123). Considérons le scénario de test suivant décrivant une situation système de conduite pouvant survenir :

> *En situation normale de conduite, la puissance thermique du circuit primaire est constante (1500 Watts), en cas de dysfonctionnement du système principal de refroidissement (situation incidentelle/accidentelle) la température au sein du circuit primaire augmente rapidement et par conséquent la puissance résiduelle augmente également (100 sec < t <120 secondes). Lors de l'apparition du dysfonctionnement (t = 100 sec), le système de sécurité envoie automatiquement une requête au système d'alimentation en eau de secours (plate-forme CISPI) pour refroidir le circuit primaire.*

En analysant les traces d'exécution des sous-systèmes de commande et de contrôle, on observe que dès la réception de la requête d'alimentation en eau de secours le sous-système de commande fixe une consigne de contrôle en puissance de 50 Watts sur la voie d'alimentation 1, le sous- système de contrôle régule alors la puissance d'alimentation en eau de secours autour de cette consigne. Dès que la puissance d'alimentation en eau de secours est suffisante la puissance résiduelle dans le circuit primaire commence à diminuer (t = 120 secondes). Nous observons également que lors d'une apparition de défaillance d'alimentation en eau de secours à travers la voie 1 (t= 205 secondes), le sous-système de commande enclenche automatiquement la régulation en puissance de l'alimentation à travers la voie 2. Une fois les conditions de mise en service du système RRA atteintes (t = 390 sec), le sous-système de commande fixe les consignes de contrôle en puissance à zéro.

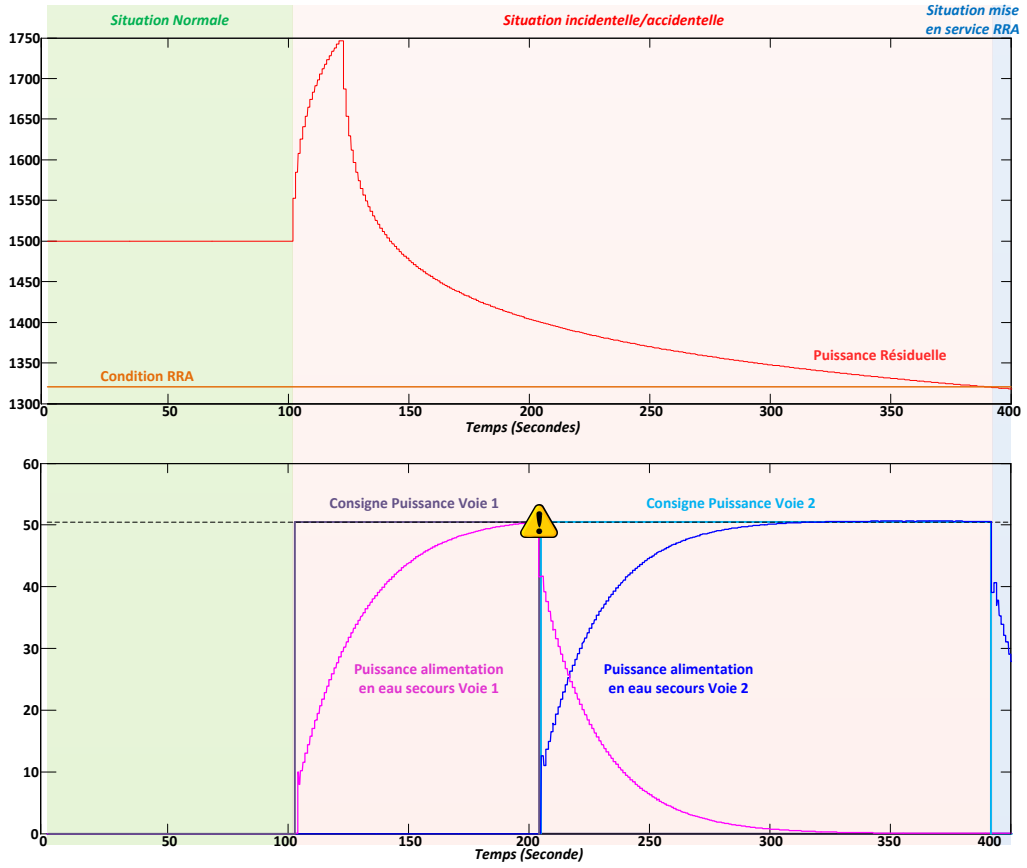


Figure 123 : Trace d'exécution de la situation système de conduite d'alimentation en eau de secours Conclusion

En exécutant ainsi un ensemble de scénarios de test couvrant l'ensemble des situations système de conduite pouvant se produire, l'**espace-problème** d'ingénierie système s'assure que le système manifeste les phénomènes requis par le circuit primaire à savoir l'évacuation de la puissance résiduelle en situation incidentelle ou accidentelle. En ce sens, la spécification du sous-système de commande est validée par le domaine d'ingénierie système.

6.4. Conclusion : Co-spécification exécutable d'un modèle partiel constitutif de la partie logique ciblée

L'objectif de ce chapitre était de définir la partie logique de la plate-forme CISPI dans notre environnement de co-spécification système selon le patron architectural CMMS-IAMS et la triade « **Communication, Control, Command** » du conceptagon.

D'un point de vue « **contrôle »**, nous avons illustré l'intérêt de la Commande Sans Modèle pour spécifier le comportement adaptif d'une fonction de contrôle selon la situation système de conduite. De plus, nous avons montré que l'étude de la dynamique du paramètre F, terme estimé en ligne de la CSM, fournit une information relative à la détection globale d'une dérive du système et permet ainsi de relier les sous-systèmes de conduite, de gestion technique et de maintenance composant notre architecture CMMS-IAMS retenue.

D'un point de vue « **commande »**, nous avons illustré l'intérêt du patron architectural CMMS-IAMS pour spécifier une fonction de commande à l'interface du système extérieure (*environnement*) et du système intérieur (*sous-système de contrôle et sous-système physique*). Le sous-bloc fonctionnel (**Commander**) établit alors les consignes de contrôles permettant de maintenir ou faire évoluer la situation système courante de conduite, en fonction de la situation système requise par l'environnement (sous-bloc fonctionnel (**Acter**)). Le sous-bloc fonctionnel (**Rendre Compte**) élabore les comptes rendus à destination du système extérieur afin de s'assurer du bon déroulement des activités de conduite.

Le point de vue « **communication »** permettant de relier le système intérieur au système extérieur ainsi que les composants du système intérieur entre eux, n'a pas été développé dans ce mémoire. En phase d'ingénierie, nous interprétons le bus de co-simulation système comme étant une spécification du sous-système de communication. En effet, le bus de co-simulation orchestre et supporte les échanges de données entre les différents modèles des sous-systèmes composant la plate-forme CISPI.

Partie 3 : Co-spécification exécutable de constituants de conduite du procédé du système CISPI

Conclusion : Co-spécification d'un modèle total constitutif de l'interaction physico-logique ciblée

Nous venons d'illustrer dans cette partie la construction itérative à un niveau d'abstraction fonctionnel d'un modèle holistique de la plate-forme CISPI du CRAN. Cette construction est guidée par l'étude des triades du conceptagon, à la source des itérations de spécification système entre le domaine d'ingénierie système et les domaines d'ingénieries spécialistes. Plus précisément, ce modèle système est le résultat de deux itérations de co-spécification système consistant à percevoir le système selon un point de vue physique au travers de la triade « **Function, Structure, Process** » et selon un point de vue logique au travers de la triade « **Communication, Control, Command** ».

La première itération de spécification entre l'**espace-problème** d'ingénierie système et l'**espace-solution** d'ingénierie du procédé définit à un niveau d'abstraction fonctionnel le modèle de la partie physique de la plate-forme CISPI selon un modèle processif en puissance et énergie. Ce modèle processif permet ainsi de s'affranchir des contraintes organiques et de respecter les lois d'agencement de la physique. Il est alors vu comme un premier modèle holistique de la plate-forme CISPI autour duquel l'ingénieur système architecture l'ensemble des contributions spécialistes.

La seconde itération de spécification entre l'**espace-problème** d'ingénierie système et l'**espace-solution** d'ingénierie d'automatisation raffine le modèle de la physique d'un point de vue logique pour maintenir ou faire évoluer le processus physique de la plate-forme CISPI afin de manifester les phénomènes requis par le système extérieur. Le modèle de la partie logique architecture, selon le patron CMMS-IAMS, un modèle du contrôleur intégrant la CSM et un modèle de commande, sans présager d'une répartition entre des agents techniques et humains.

Au l'issue de ces deux itérations de co-spécification système, l'exécution du modèle global de la plate-forme CISPI permet à l'**espace-problème** d'ingénierie-système de valider l'architecture système assemblant la partie logique et la partie physique du système. Dans la prochaine partie de ce mémoire, nous explorons les problèmes d'allocation des fonctions de conduite à des agents techniques ou humains en tant que respectivement composants d'actionnement et de mesurage intelligents du système global (Whole-System). Cette répartition des fonctions de conduite nécessite de déterminer une plage de faisabilité que nous détaillons d'un point de vue technique dans le chapitre 7 et d'un point de vue humain dans le chapitre 8.

D'un point de vue co-exécution de modèles, nous avons ainsi montré :

- L'intérêt de permettre à chacun des domaines spécialistes de spécifier et vérifier sa solution dans son propre environnement de modélisation-simulation,
- La validation système de ce constituant (modèle) partiel (M_{IPro_SysML}) du modèle global visé ($M_{IS,\ SysML}$) par intégration avec le procédé réel CISPI.
- La difficulté de translater en langage SysML la connaissance spécialiste suffisante pour orchestrer cette solution dans le comportement d'ensemble.

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage du système de conduite CISPI

Introduction : Interaction collaborative entre domaines d'ingénierie interdisciplinaire centrée technique et centrée physiologique

L'objectif des deux chapitres de cette quatrième et dernière partie est de mettre en œuvre notre heuristique de co-spécification système pour raffiner l'interaction logico-physique entre le sous-système physique et sous-système logique de la plate-forme CISPI. Alors que dans le chapitre précédent nous avons considéré cette interaction comme une fonction identité, nous nous préoccupons dans cette partie à son comportement et sa structure en tant qu'opérateur de nature[60] opérant des transmutations entre des flux logiques et physiques. En ce sens, nous interprétons cette interaction comme un constituant fonctionnel d'actionnement et de mesure intelligent (IAMS) assurant la relation causale entre la partie physique et la partie logique de notre système d'intérêt (Figure 124).

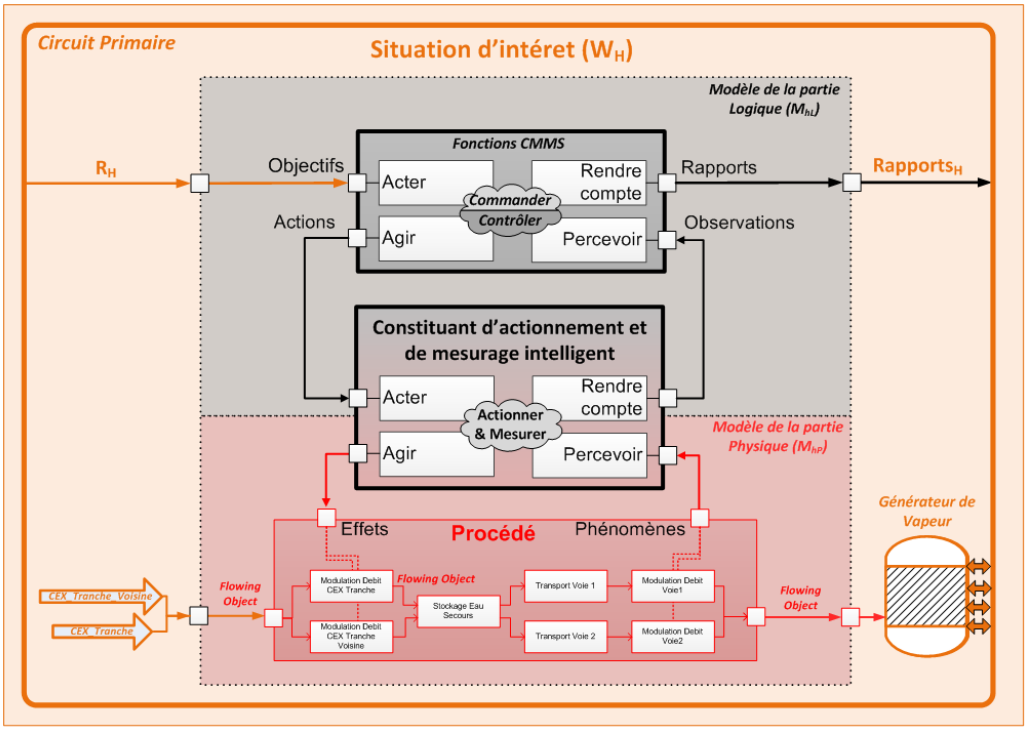


Figure 124 : Patron architectural CMMS-IAMS logique visé

Selon notre approche holistique de co-spécification système, il aurait fallu dans un premier temps définir un modèle du système d'actionnement et de mesurage intelligent comme un TOUT afin d'assurer la cohérence de ce dernier. Dans les deux chapitres suivants de ce mémoire, nous nous intéressons à la co-spécification itérative d'un modèle exécutable de ce système IAMS selon deux modèles constitutifs de constituants (agents) de natures différentes (Figure 125) :

- Agent technique d'actionnement intelligent (IA)
- Agent humain de mesure intelligent (IM)

---

[60] au sens de la systémique (Le Gallou & Bouchon-Meunier, 1992)

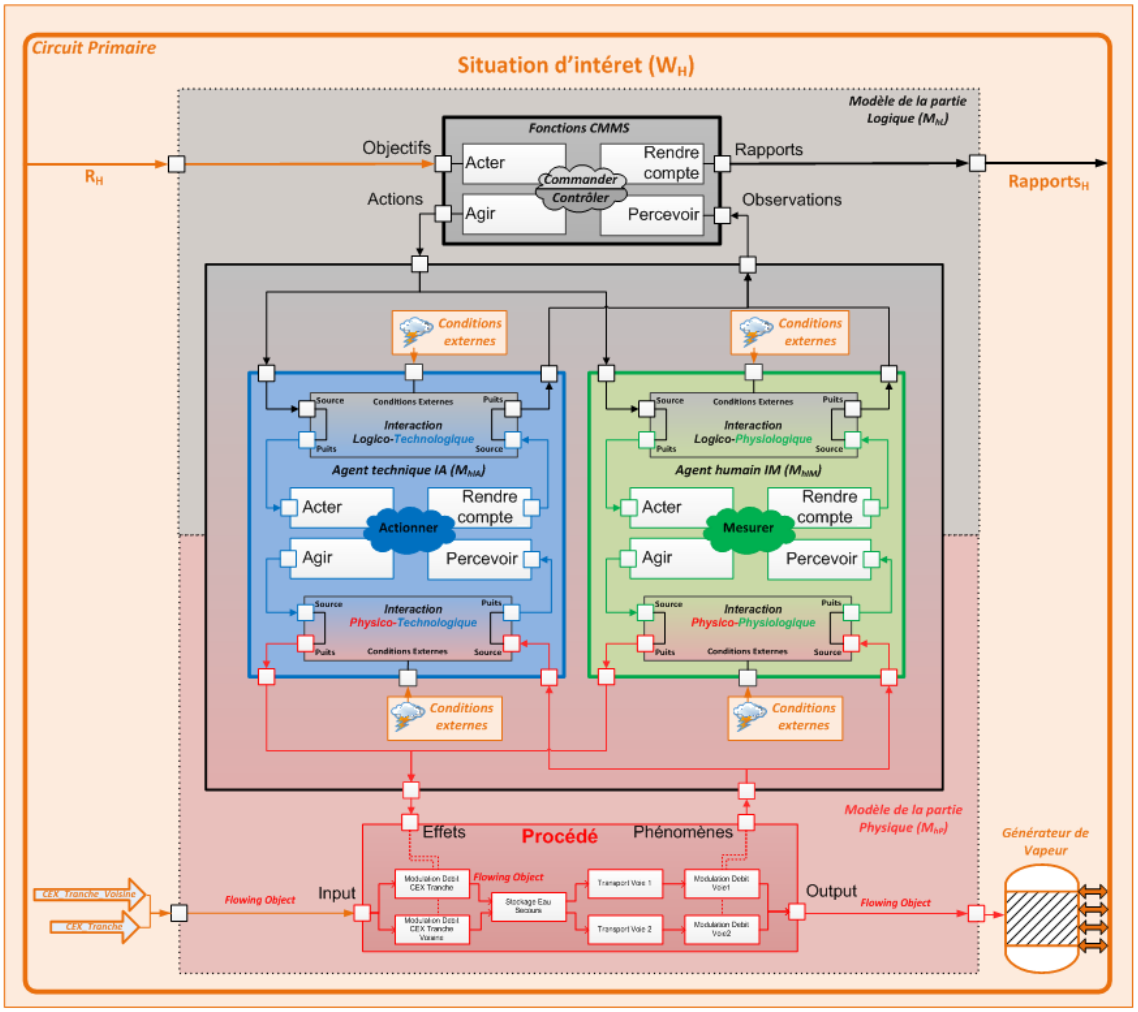


Figure 125 : Répartition de la fonction IAMS entre un agent technique d'actionnement intelligent et un agent humain de mesurage intelligent

L'intégration de l'ensemble des modèles spécialistes des constituants technique d'actionnement et humain de mesurage est effectuée par co-simulation afin de spécifier un comportement global de la plate-forme CISPI assurant ainsi le bon fonctionnement de l'ensemble.

> "*Simulations can be orchestrated linking models together across domains to examine previously unknown design spaces. It is through the linking of these various simulation that more is learned about the dynamical properties of the whole*" (G. A. Boy & Narkevicius, 2014)

Sans se poser le problème de répartition de ce système IAMS sur des agents techniques ou humains (*Chapitre 2*), nous détaillons (Figure 126) :

- Dans le chapitre 7, la co-spécification d'un modèle exécutable d'un constituant d'actionnement intelligent de régulation de puissance (S_{FAI_SysML}), supporté par un agent technique. Ce modèle est co-spécifié entre l'**espace-problème** d'ingénierie système et les **espaces-solution** du procédé et fournisseur d'actionneurs intelligents. Nous évaluons la maturité de cette solution technique par co-exécution de modèles selon une série de tests intégrant des modèles de l'environnement de plus en plus réalistes. Cela nous amène dans un premier temps à raffiner le modèle fonctionnel du processus

CISPI selon un modèle logique (S_{IPro_SysML}) permettant ainsi d'évaluer le modèle d'actionnement intelligent (M_{hIA}) en co-spécification exécutable (*Model In the Loop*). Dans un second temps, ce modèle constitutif d'actionnement intelligent est couplé à notre plate-forme CISPI qui reproduit un environnement représentatif d'une situation de conduite (*Hardware/Software In the Loop*), pour finir avec la réalisation d'un prototype.

- Dans le chapitre 8, la co-spécification d'un modèle exécutable d'un sous-système de mesurage intelligent (S_{IPhy_SysML}) supportée par un agent humain de conduite. Une première itération de co-spécification (S_{IPhy_SysML}) entre l'**espace-problème** d'ingénierie système et un **espace-solution** d'ingénierie physiologique nous permet de démontrer que la bonne exécution d'une fonction de conduite exécutée par un agent humain est conditionnée par la bonne perception (Leger, 1999) sensorielle des phénomènes émis par l'environnement d'intérêt. Il s'agit là d'une des conditions nécessaire mais non suffisante de répartition des fonctions de conduite entre agents techniques et humains en situation opérationnelle de conduite. Nous en faisons un artefact de co-spécification système pour que l'architecte système orchestre au plus tôt les spécifications d'ingénierie technique au regard des faisabilités humaines. En ce sens, nous nous focalisons sur l'interaction technico-physiologique de perception sonore d'un agent technique vers un agent humain

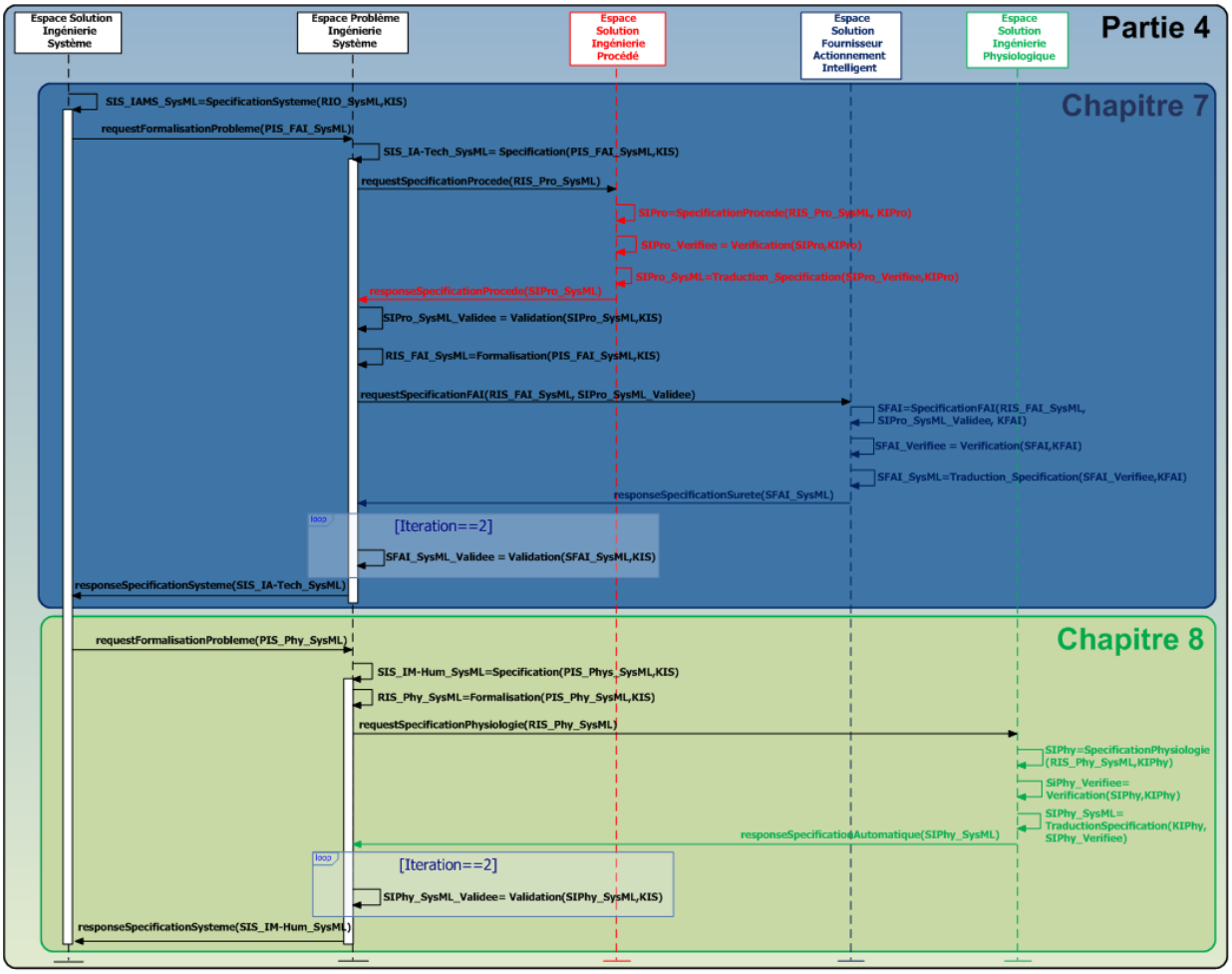


Figure 126 : Diagramme de séquence (seq) SysML - Scénario de co-spécification d'un système d'actionnement et de mesurage intelligent

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage du système de conduite CISPI

7. Chapitre 7 : Co-spécification exécutable d'un constituant technique d'actionnement du système de conduite CISPI

7.1. Introduction : Partie « actionnement intelligent » du système de conduite CISPI

A cette étape de notre scénario de co-spécification système, notre intérêt se porte sur le raffinement d'une fonction d'actionnement intelligent opérée par un constituant technique en interaction avec les parties physique et logique de la plate-forme CISPI (Figure 127). D'un point de vue technique, ce constituant exécute un ensemble d'activités organisées les unes par rapport aux autres (Després, 1991) :

- **Acter :** Le constituant technique d'actionnement intelligent reçoit des objectifs qu'il doit valider en fonction des contraintes liées à la conduite du système (*plage de valeur à respecter, mise en service d'équipements si lignage effectué, situation courante de conduite …*) qui sont capitalisées dans des procédures. Ces objectifs peuvent être émis par des constituants humains mais également par d'autres constituants techniques de niveaux supérieurs fixant les objectifs, c'est-à-dire les valeurs de consigne que le processus physique doit atteindre. Ainsi, ces objectifs permettent de synchroniser d'un point de vue opérationnel différent niveaux hiérarchiques de conduite.
- **Agir :** A partir des objectifs de conduite, le constituant technique d'actionnement intelligent élabore les ordres de contrôle qui vont lui permettre de stimuler les actionneurs (puis effecteurs) sous sa dépendance et avec lesquels il doit se coordonner. Ces ordres permettent in–fine d'agir sur le processus physique afin d'atteindre les objectifs validés précédemment. Des variables de réglage peuvent être également transmises. Elles n'agissent pas directement sur le processus physique mais permettent de choisir la stratégie la mieux adaptées pour répondre aux réglages souhaités.
- **Percevoir :** Le constituant technique d'actionnement intelligent acquiert des informations qui lui permettent d'observer l'effet des actionnements sur le processus physique à conduire au regard des objectifs. Ces informations issues de capteurs permettent d'apprécier l'état et l'évolution de la transformation en cours. Des informations de contrôles signalant une défaillance et diffusées par les actionneurs ou capteurs peuvent être également recueillies, afin d'adapter la stratégie de conduite en fonction de la situation courante de l'installation.
- **Rendre Compte :** Le constituant technique d'actionnement intelligent émet également des comptes rendus. Ces derniers synthétisent l'activité du constituant compte tenu des moyens mis à sa disposition (*rapports synthétiques de l'activité en cours, s'il respecte les objectifs, ou sinon les raisons de l'insuccès et la stratégie de repli adoptée*). Ces informations permettent également de synchroniser de façon opératoire les traitements à réaliser par les différents niveaux supérieurs concernés.
- **Actionner :** Cette activité en interaction avec les 4 précédentes représente l'intelligence (*algorithmique*) du constituant technique d'actionnement. Cette intelligence repose sur un ensemble de procédures de conduite et de retours d'expériences implémentés et constitue une référence pour l'ensemble des autres activités. Cette activité assure également la synchronisation entre les quatre activités présentées précédemment.

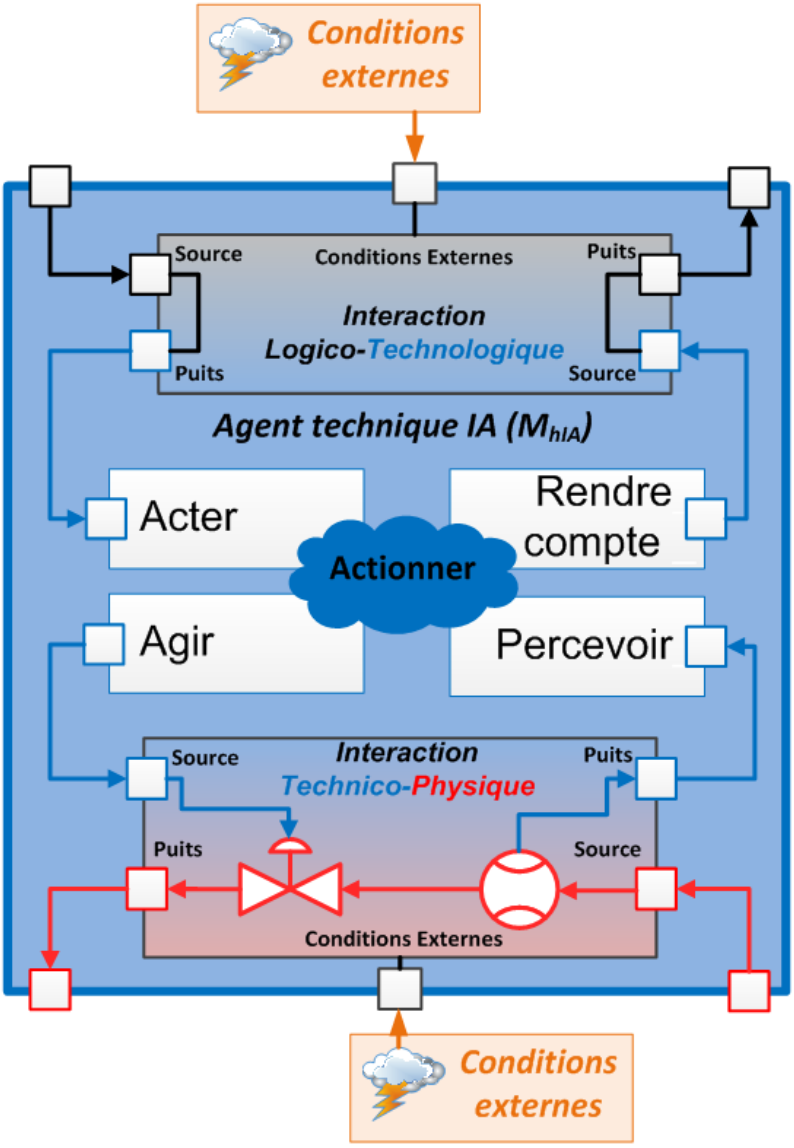


Figure 127 : Patron architectural d'Actionnement Intelligent par un constituant technique

7.2. Co-spécification système d'un modèle d'interaction logico-physique de modulation de débit

Dans la suite de ce chapitre, nous nous focalisons sur une fonction d'actionnement intelligent de modulation de débit implémentée par un constituant technique requis pour satisfaire l'ensemble des exigences systèmes (Cf. Chapitre 6) :

- (R_{IS_Aut1}) : «*En situation incidentelle ou accidentelle, la puissance d'alimentation en eau de secours doit être maintenue à 50 Watts à travers la voie 1* »
- ($R_{IS_Aut\text{-}Cont\ 1\text{-}4}$) : «*La puissance d'alimentation en eau de secours ne doit pas excéder 75 Watts* »
- ($R_{IS_Aut\text{-}Cont\ 1\text{-}5}$) : « *La consigne en puissance doit être atteinte entre 100 et 150 Secondes* ».

En ce sens, nous raffinons le modèle de la fonction de contrôle de la partie logique de la plate-forme CISPI, afin de prendre en compte les contraintes techniques d'un composant d'actionnement intelligent distribué.

7.2.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction physique de modulation de débit

7.2.1.1. Éléments de connaissances d'une interaction physique

De manière générale, les composants techniques d'actionnement intelligent cherchent à attribuer aux actionneurs composant le processus physique d'un système, une certaine autonomie en leur conférant une intelligence technique. Dans un contexte distribué, toutes les sous-fonctions (*Acter, Agir, Percevoir, Rendre compte, Actionner*) du composant d'actionnement intelligent sont directement implantées dans ce dernier afin, d'une part, de réagir plus rapidement à ses évolutions, et d'autre part, fournir à la partie logique des informations cohérentes et fiables (et non plus des données) sur son état. Les fonctions de contrôle-commande de niveau plus global, correspondant à l'intégration du composant d'actionnement intelligent par rapport à son environnement, sont quant à elles implantées dans la partie logique communicant avec le composant d'actionnement intelligent sous la forme d'objectifs et de comptes rendus (Figure 128).

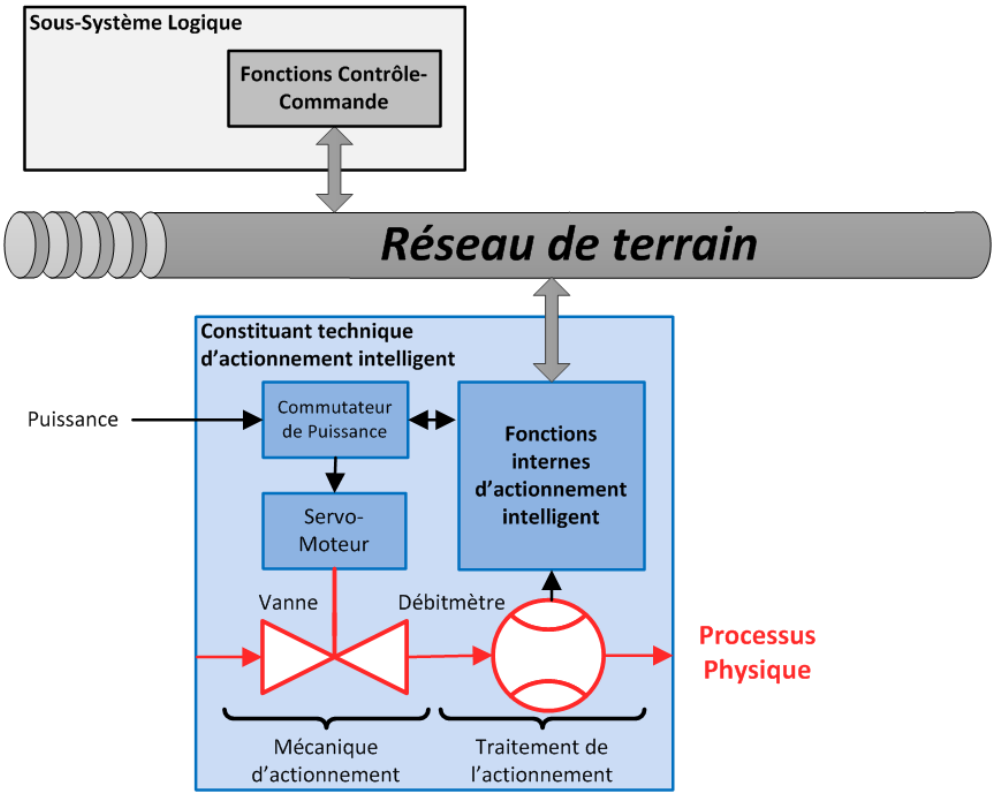


Figure 128 : Principe d'un constituant technique d'actionnement intelligent (G. Morel, 1999)

De façon plus détaillée, la partie logique de contrôle-commande est chargée des actions à entreprendre sur le processus physique au regard de la finalité du système. Pour cela il envoie des objectifs à des actionneurs qui les convertissent en actions effectives sur le processus en modulant certaines variables de ce dernier (pression,

débit d'un fluide, position d'une pièce,…). En retour les capteurs observent les réactions du processus physique et fournissent les données pour les sous-fonctions internes.

Dans notre scénario de co-spécification système, le composant technique d'actionnement intelligent constitue la partie logique du processus physique de conversion (*transformation de la puissance*) permettant de moduler en puissance l'alimentation en eau de secours à travers la voie 1 (Cf. Chapitre 5) (Figure 129). D'un point de vue technique, cette modulation en puissance s'effectue via un actionneur et un capteur. Notons que dans le cadre de l'évolution de CISPI, une contrainte budgétaire nous impose de réutiliser le maximum de composants de la plate-forme existante. La question est alors de savoir si les vannes de régulations et débitmètres présents sur la plate-forme (Figure 130) sont suffisants pour moduler la puissance d'alimentation en eau de secours en sortie de la voie 1.

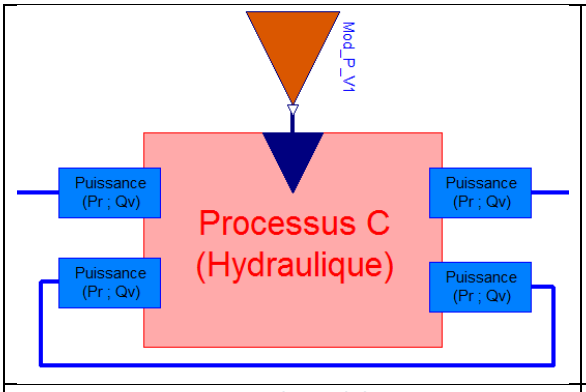


Figure 129 : Processus C de modulation en puissance en sortie de la voie 1 d'alimentation en eau de secours

Figure 130 : Vanne de régulation et débitmètre de la voie 1 de la plate-forme CISPI

Considérons le processus C, la puissance hydraulique en sortie est exprimée par le couple des variables généralisées (Pr, Qv) avec :

$$P_{Hydraulique} = Qv * Pr \qquad (26)$$

$P_{Hydraulique}$ = Puissance hydraulique (W) Pr = pression (Pascal)
Q = débit (m^3/s)

Dans notre cas d'étude, la cuve émulant le générateur de vapeur est une cuve ouverte, en ce sens la pression à la sortie de la voie correspond à la pression atmosphérique. Considérant la pression atmosphérique comme étant constante sur la durée de nos expériences, la puissance hydraulique à la sortie de la voie 1 est proportionnelle au débit. Il est ainsi possible de moduler la puissance d'alimentation en eau de secours en modifiant le débit d'alimentation via la vanne de régulation existante, mais également d'observer les effets de cet actionnement au travers du débitmètre. D'autres études nous ont également permis de valider le dimensionnement de la vanne de régulation et du débitmètre existants au regard des exigences systèmes présentées dans le chapitre suivant.

7.2.1.2. Définition des exigences de l'interaction physique ciblée

A cette étape du scénario de co-spécification système, l'**espace-solution** d'ingénierie système ne possède pas l'ensemble des **connaissances** requises pour spécifier un composant technique d'actionnement intelligent et se pose notamment la question de l'intégration de ce composant technique à la plate-forme CISPI existante. En ce sens, l'**espace-problème** d'ingénierie système formalise un ensemble d'exigences systèmes : $\{R_{IS_FAI1}, R_{IS_FAI2}, R_{IS_FAI3}, R_{IS_FAI4}\}$ formalisant ce problème (Figure 131).

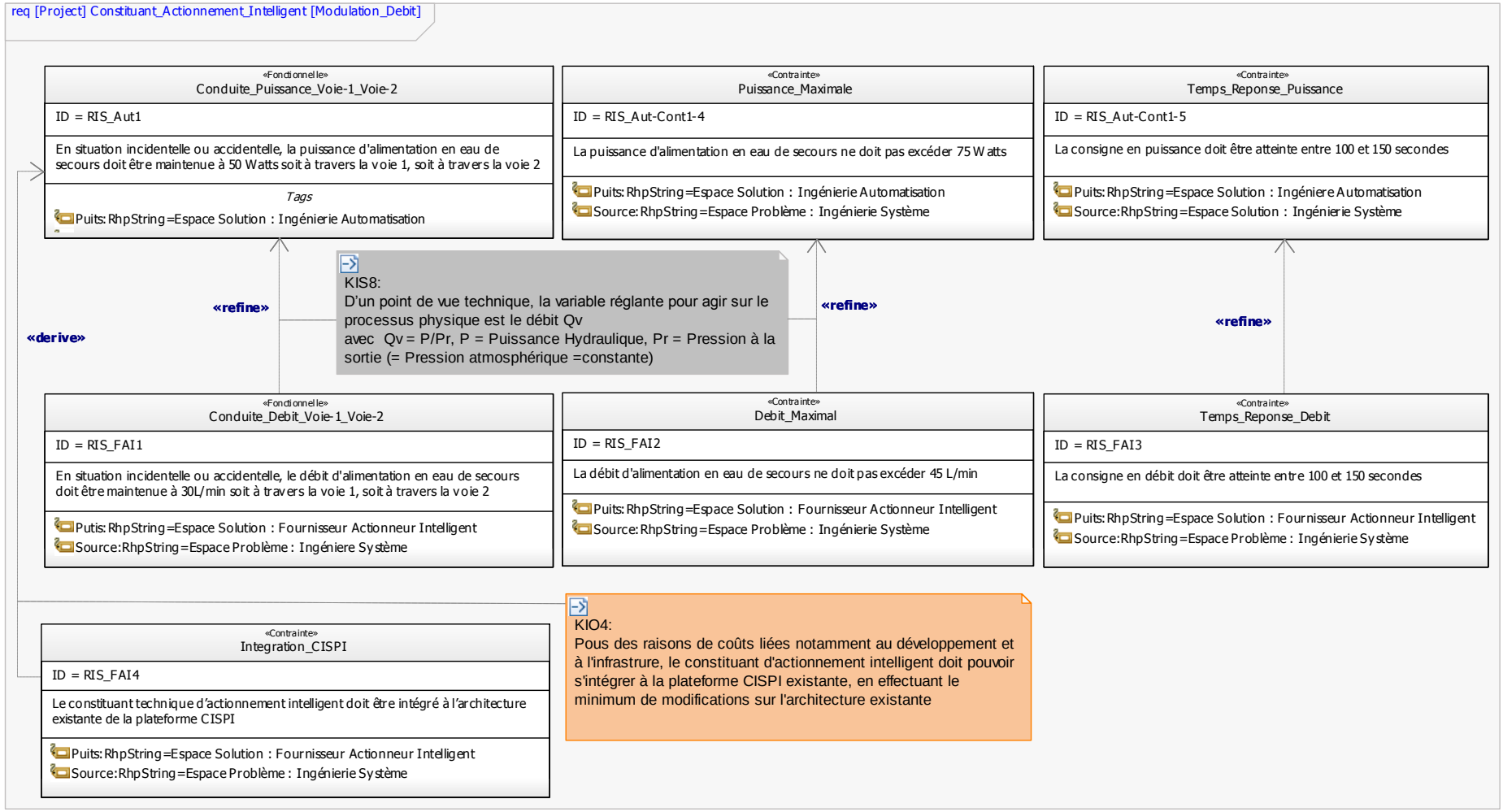


Figure 131 : Modèle d'exigences système ($R_{IS_FAI_SysML}$) relatives au constituant technique de modulation de débit

L'**espace-solution** d'ingénierie spécialiste, retenu pour recevoir ces exigences par notification à travers le bus de co-spécification système, est un Fournisseur d'Actionneur Intelligent (FAI) ayant contribué à la conception de la plate-forme CISPI existante. Ce dernier est alors responsable de la prescription d'une solution satisfaisant l'ensemble de ces exigences systèmes.

De façon plus détaillée, les exigences systèmes (R_{IS_Aut1}, $R_{IS_Aut\text{-}Cont\ 1\text{-}4}$, et $R_{IS_Aut\text{-}Cont\ 1\text{-}5}$) de contrôle-commande en puissance sont raffinées pour prendre en compte les contraintes techniques d'actionnement intelligent relatives au contrôle du débit d'alimentation en eau de secours (R_{IS_FAI1}, $R_{IS\text{-}FAI2}$, R_{IS_FAI3}). Une nouvelle exigence (R_{IS_FAI4}) est également dérivée pour prendre en compte l'intégration du constituant technique d'actionnement intelligent à l'architecture globale de la plate-forme CISPI existante.

7.2.2. Spécification prescriptive d'un modèle d'exigences constitutif d'une interaction logique de modulation de débit

7.2.2.1. Eléments de connaissance d'une interaction logique

De façon plus détaillée, le constituant d'actionnement intelligent étudié fait évoluer des objets passifs OMT (*Vanne et Débitmètre*) en un objet actif OLT (Objet Logique Technique) devant s'intégrer à l'architecture existante SIAC de la plate-forme CISPI. Afin de garantir cette intégration, cet OLT doit satisfaire un certain nombre de caractéristiques :

- Possède une identification unique (Wong, McFarlane, & Agarwal, 2002)
- Est capable de communiquer avec son environnement (Kintzig, Poulain, Privat, & Favennec, 2002)
- Peut mémoriser et gérer des informations propres
- Dispose d'un langage de dialogue et d'échange d'informations et d'états. Nos travaux dans le chapitre 5, nous amène à définir le langage d'interopération d'un point de physique en énergie et en puissance en tant que langage pivot entre les différents sous-domaines de la physique (mécanique, hydraulique, thermique,…).
- Participe au processus de prise de décisions concernant son état
- Peut surveiller et contrôler son environnement
- Peut réaliser des interactions à travers les services qu'il offre (Bajic, 2005)

Ces caractéristiques constituent une partie de la **connaissance** (K_{FAI}) de l'**espace-solution** fournisseur d'actionnement intelligent.

7.2.2.2. Définition des exigences de l'interaction logique ciblée

Cette première itération de spécification a pour objectif d'évaluer la faisabilité d'une solution technique satisfaisant l'ensemble des exigences systèmes {R_{IS_FAI1}, R_{IS_FAI2}, R_{IS_FAI3}, R_{IS_FAI4}}, mais aussi de contractualiser la relation de spécification entre l'**espace-solution** FAI et l'**espace-problème** d'ingénierie système. En ce sens, le FAI analyse les exigences systèmes à partir de ses **connaissances** métiers (K_{FAI}) afin de prescrire à l'**espace-problème** d'ingénierie système un premier modèle d'exigences du constituant d'actionnement intelligent de régulation de débit de la plate-forme CISPI. Ainsi, l'ensemble des exigences systèmes {R_{IS_FAI1}, R_{IS_FAI2}, R_{IS_FAI3}, R_{IS_FAI4}} sont raffinées en exigences techniques afin de prendre en compte les contraintes techniques de réalisation d'un constituant d'actionnement intelligent s'intégrant à l'architecture SIAC de la plate-forme CISPI existante.

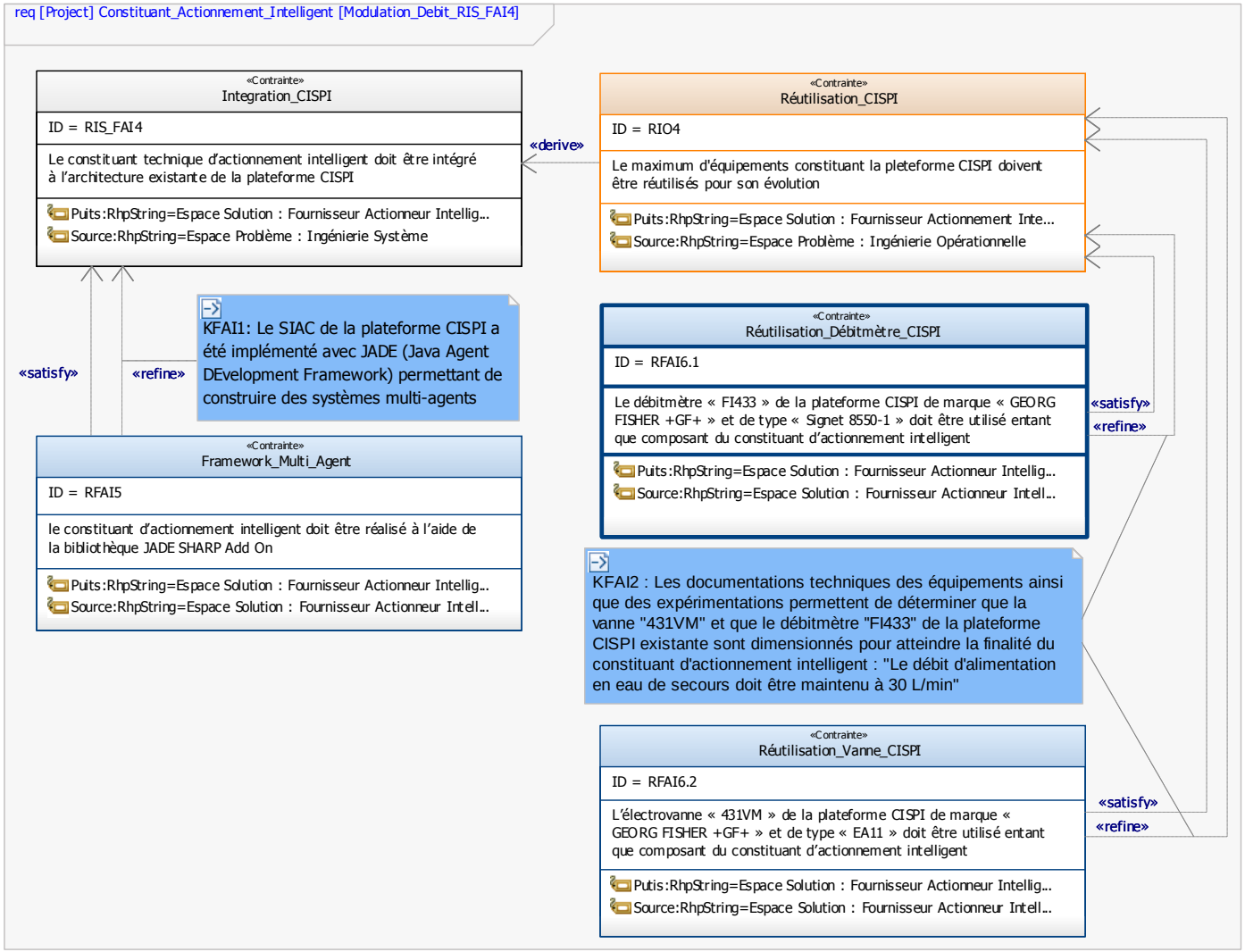


Figure 132 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence système (R_{IS_FAI4}) par l'espace-solution fournisseur d'actionneur intelligent

A titre d'exemple, nous détaillons ici le raffinement de l'exigence (R_{IS_FAI4}) (Figure 132), le raffinement des autres exigences systèmes décrites à l'**espace-solution** FAI étant représenté en annexe (Cf. Annexe 7-1). L'exigence système (R_{IS_FAI4}) est raffinée à partir de la connaissance (K_{FAI1}) pour produire l'exigence technique (R_{FAI5}) contraignant l'implémentation du constituant d'actionnement intelligent selon la bibliothèque « JADE SHARP Add On », utilisée pour définir la plate-forme multi-agents de CISPI. Notons également que l'**espace-solution** d'ingénierie opérationnelle dérive, à partir de l'exigence (R_{IS_FAI4}), une nouvelle exigence technique (R_{IO4}) requérant qu'un maximum d'équipements composant la plate-forme CISPI existante soit réutilisé. Cette exigence (R_{IO4}) est à son tour raffinée en deux exigences techniques ($R_{FAI6.1}$) et ($R_{FAI6\cdot 2}$) à partir de la connaissance (K_{FAI2}) du fournisseur d'actionneur intelligent pour préciser quels équipements de la plate-forme CISPI existante devront être utilisés pour la réalisation du constituant d'actionnement intelligent.

L'ensemble de ces exigences techniques spécifiées par le FAI définissent un premier modèle du constituant d'actionnement intelligent et peuvent ainsi être prescrites à l'**espace-problème** d'ingénierie système sous forme d'un diagramme d'exigences SysML pour validation. Néanmoins ce premier niveau descriptif de modélisation n'est pas suffisant pour notre heuristique de co-spécification système qui requiert de prescrire un modèle constitutif d'une exécution d'ensemble avec d'autres modèles, afin de valider le comportement global de plate-forme CISPI en spécification.

7.2.3. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction logico-physique de modulation de débit

7.2.3.1. Eléments de modélisation exécutable d'une interaction logico-physique

Spécification Structurelle

Afin de spécifier la structure interne du constituant d'actionnement intelligent, l'**espace-solution** FAI se réfère au patron architectural Figure 127 en tant qu'architecture d'un agent technique (IAMS) interopérables avec le SIAC de la plate-forme CISPI. Cela l'amène à interpréter ce patron architectural pour prendre en compte les caractéristiques liées à la conception du constituant d'actionnement intelligent de modulation en débit et à son intégration avec le système existant (Figure 133).

Ainsi, le bloc « ***d'interaction logico-technologique*** » réalise l'interface avec le bus d'informations numérique de la plate-forme CISPI. Il reçoit des objectifs en puissance du sous-système logique de contrôle-commande et lui fournit des rapports sur l'exécution globale de la fonction de modulation en puissance. Ce bloc « ***d'interaction logico-technologique*** » est également en charge de la conversion de l'objectif en puissance vers un objectif en débit. Le bloc « ***Acter*** » reçoit les objectifs en débit et valide ou non ces objectifs en fonction de la situation courante de conduite. Une fois les objectifs validés, le bloc « ***Agir*** » élabore les ordres de contrôle en position d'ouverture de la vanne réglante composant le bloc « ***d'interaction Technico-Physique*** ». Ce dernier est également composé d'un débitmètre qui fournit les observations de débit au bloc « ***Percevoir*** » s'assurant que les observations sont cohérentes avec la situation courante de conduite. Enfin le bloc « ***Rendre Compte*** » élabore des rapports concernant la modulation en débit à travers la voie 1 qui sont ensuite transposer en puissance par le bloc « ***d'interaction logico-technologique*** » pour être interpréter par le sous-système logique de contrôle-commande.

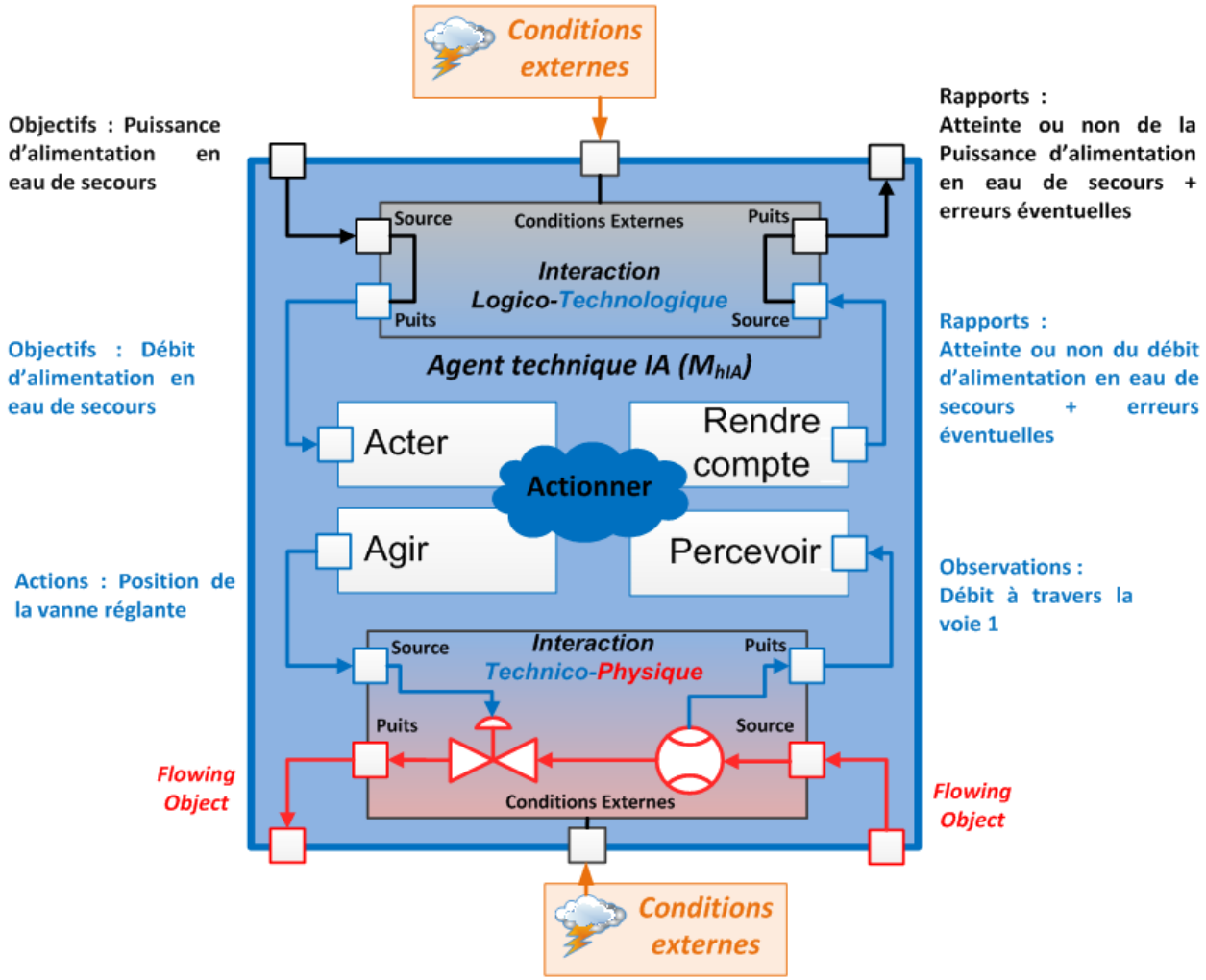


Figure 133 : Interprétation du patron architectural d'Actionnement Intelligent pour la modulation en débit à travers la voie 1 de la plate-forme CISPI

Spécification comportementale

En parallèle de la définition de l'architecture du constituant d'actionnement intelligent, **l'espace-solution** FAI définit le comportement de ce dernier. Dans ce cas, le comportement du constituant a été spécifié avec les outils Simulink et Dymola, permettant ainsi au fournisseur d'actionneur intelligent de profiter de la puissance d'expression et de modélisation de chacun de ces outils, mais également de vérifier le comportement du constituant par co-exécution de modèle. En ce sens, la majeure partie du constituant d'actionnement intelligent a été modélisé avec l'outil Simulink, pour représenter la structure interne du constituant (Figure 134) sous la forme de bloc Simulink (***Interaction Logico-Technologique, Acter, Agir, Percevoir, Rendre Compte, Actionner***) en interaction les uns avec les autres. Le comportement interne de chacun de ces blocs est décrit en Annexe 7-2 (Tableau 24). Notons que pour la partie contrôle du constituant d'actionnement intelligent, l'**espace-solution** FAI a fait le choix d'implémenter l'algorithme de régulation intelligente CSM (Commande Sans Modèle) (Cf. Chapitre 6) paramétré pour l'occasion à la modulation de débit à travers la voie 1. Ce choix se justifie du fait que la CSM présente l'avantage de prescrire un algorithme de régulation sans connaissances précises du système (*bloc d'interaction technico-physique*) à contrôler.

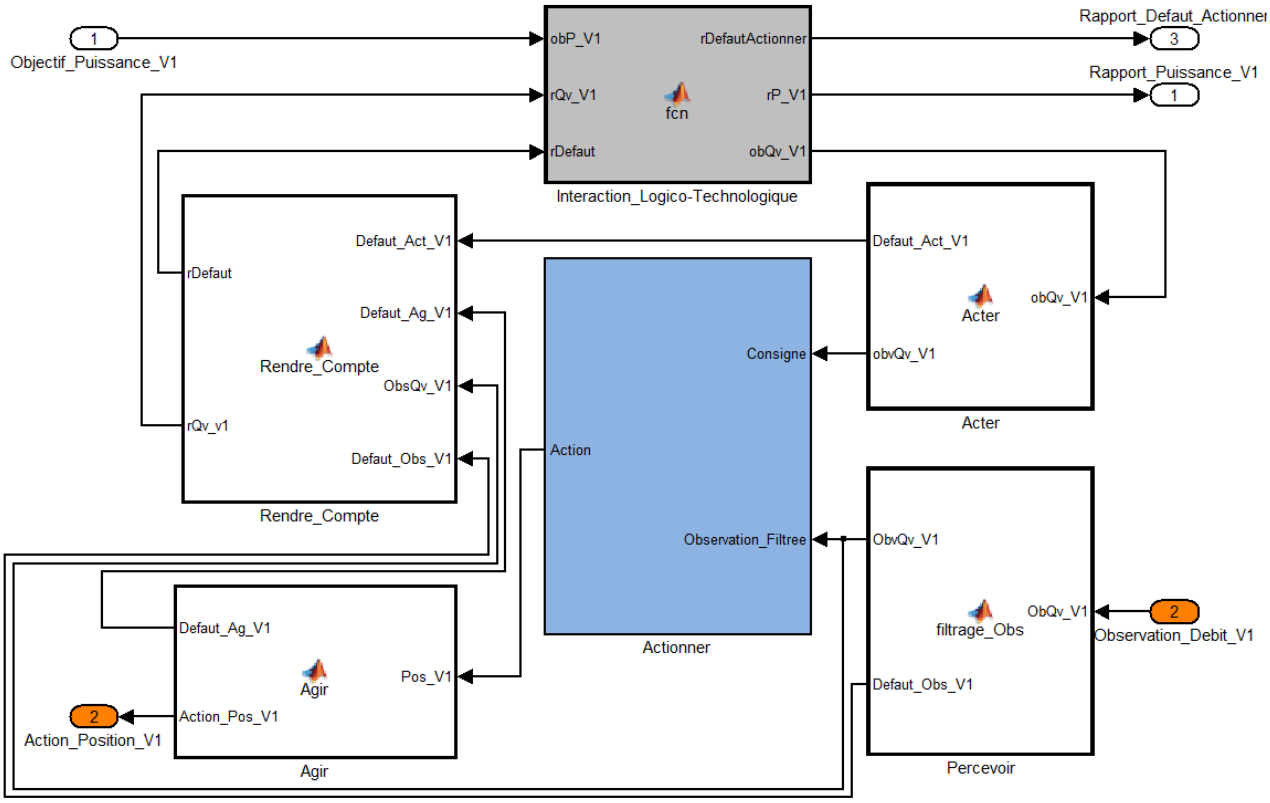


Figure 134 : Modèle (Simulink) des blocs : « interaction Logico-Technologique », « Acter », « Percevoir », « Agir » et « Rendre-Compte » composant le constituant d'actionnement intelligent

Concernant le bloc d'interaction Technico-Physique, son comportement a été spécifié dans l'outil Dymola afin de profiter des bibliothèques de Modelica regroupant un ensemble de composants développés par des spécialistes du domaine. Ainsi le modèle du bloc d'interaction Technico-Physique se compose d'un modèle logique de vanne (***vrVoie1***) et d'un modèle logique de débitmètre (***dVoie1***) reliés entre eux par un tuyau (***p3Voie1***) (Figure 134).

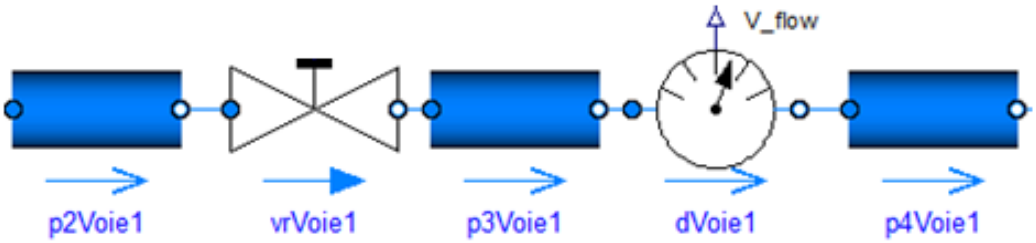


Figure 135 : Modèle logique (Modelica) du bloc : « interaction technico-physique » composant le constituant d'actionnement intelligent

Vérification par co-exécution du modèle d'interaction

Une fois le modèle du constituant d'actionnement intelligent spécifié dans les outils Simulink et Dymola, le fournisseur d'actionneur intelligent est en mesure de vérifier le comportement global du constituant par co-exécution de modèles (Figure 136) et selon différents scénarios de test. Notons qu'à des fins de simulation il a fallu rajouter au niveau du modèle Modelica une « Source » et un « Puits » pour modéliser l'écoulement de l'eau à travers la vanne et le débitmètre composant le constituant d'actionnement intelligent (Cf. Annexe 7-3 - Figure 216).

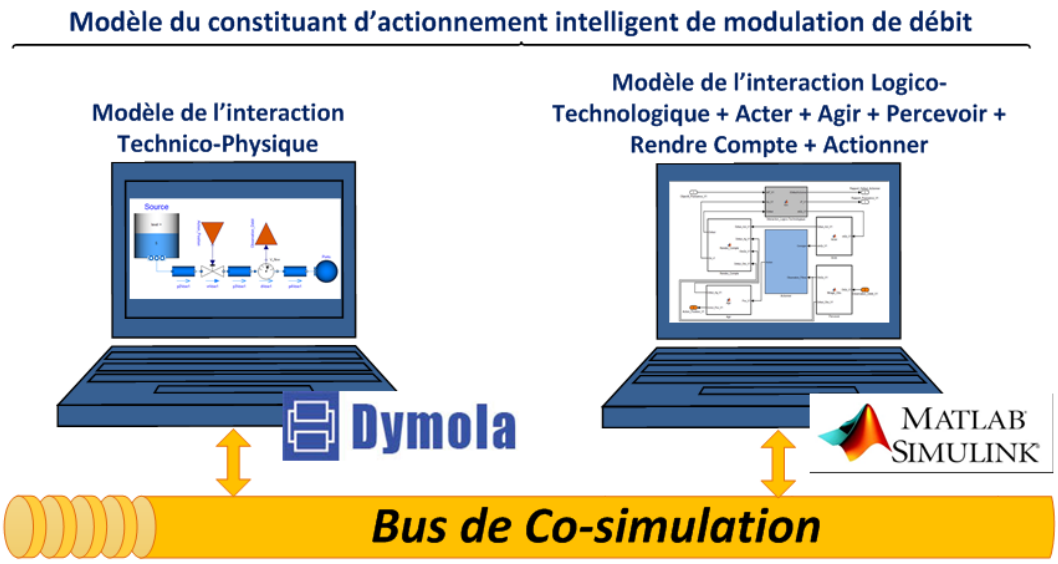


Figure 136 : Environnement de co-simulation pour la vérification de la spécification (S_{FAI}) du constituant d'actionnement intelligent de modulation de débit

Afin d'illustrer les activités de vérification, nous détaillons ici un scénario de test parmi d'autres. Ce scénario consiste à vérifier par co-exécution de modèles que pour différentes hauteurs dans la cuve « Source » le constituant d'actionnement intelligent arrive à maintenir le débit d'eau (*à 30 L/min*) tout en satisfaisant les performances (*régulation, plage de fonctionnement et temps de réponse*) requises par les exigences systèmes $\{R_{IS_FAI1}, R_{IS_FAI2}, R_{IS_FAI3}\}$. L'étude des traces d'exécution (Figure 137) de la co-exécution des modèles, nous permet ainsi de vérifier le comportement global du constituant d'actionnement intelligent, mais également de dériver de nouvelles exigences à destination du domaine d'ingénierie système. En effet, en observant les traces d'exécution nous pouvons remarquer que selon la hauteur d'eau dans la cuve de stockage (***Hauteur = 0,5 m / 1 m / 1,5 m / 2m***), le temps de réponse pour atteindre le débit (ou la puissance) requis varie. Il apparait que pour une hauteur supérieure à 1,5 mètres et inférieure à 0,5 mètres le temps de réponse prescrit par l'exigence ($R_{IS\text{-}FAI3}$) n'est pas satisfait (*La consigne en débit doit être atteinte entre 100 et 150 secondes*). Cela peut avoir des conséquences graves sur le système :

- Si le générateur de vapeur est refroidi trop lentement, il peut y avoir une surchauffe importante des équipements pouvant entrainer leurs destructions.
- Si le générateur de vapeur est refroidi trop rapidement, il peut se former des chocs thermiques pouvant également conduire à une dégradation des équipements.

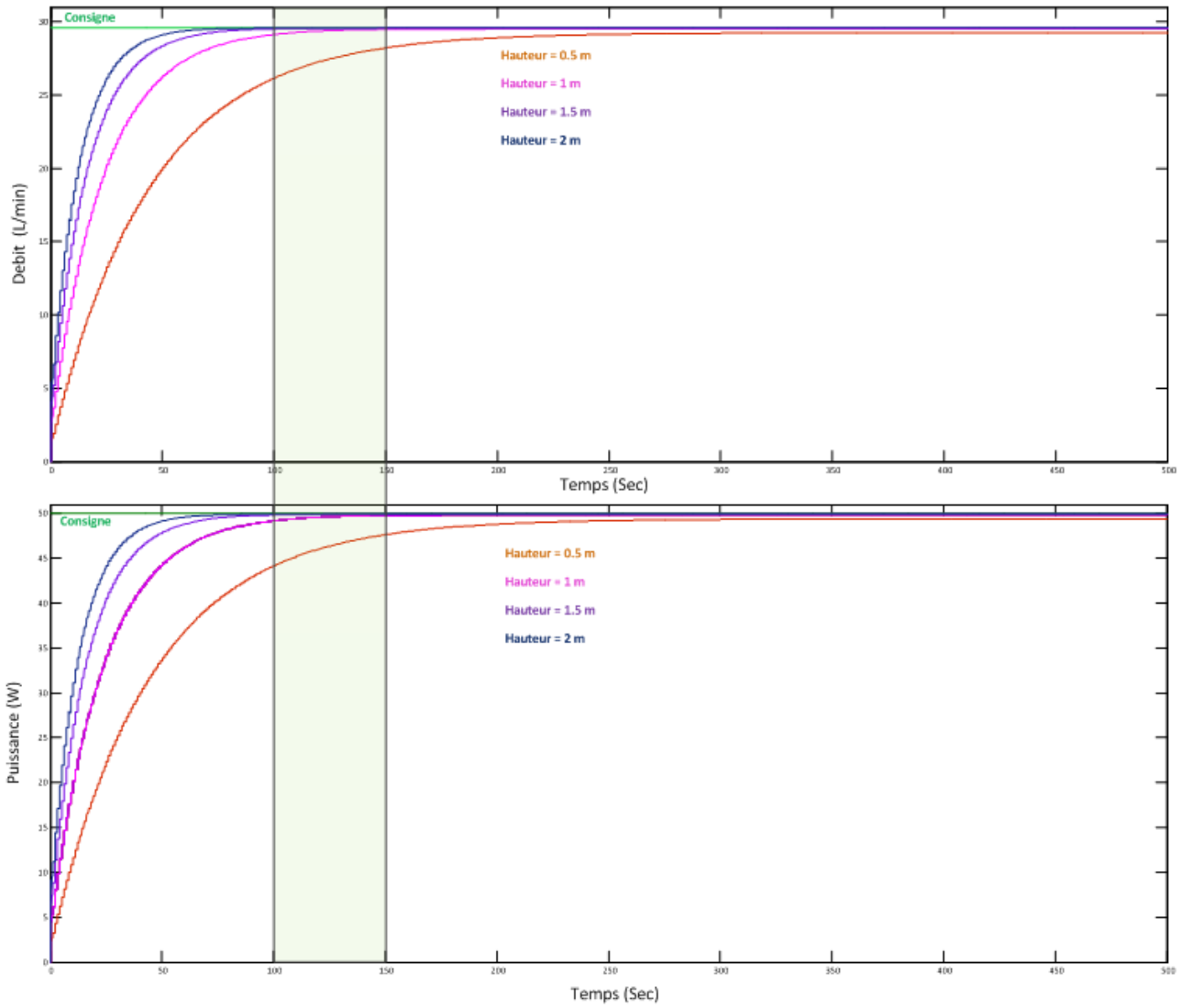


Figure 137 : Traces d'exécution de la co-simulation entre le modèle Simulink et Dymola du constituant d'actionnement intelligent

En ce sens, l'**espace-solution** FAI raffine l'exigence ($S_{IPro1.0}$) afin de prendre en compte les contraintes technologiques imposées par le constituant d'actionnement intelligent (Figure 138). Cette exigence sera ensuite prescrite à l'**espace-problème** d'ingénierie système en même temps que la spécification du modèle du constituant d'actionnement intelligent.

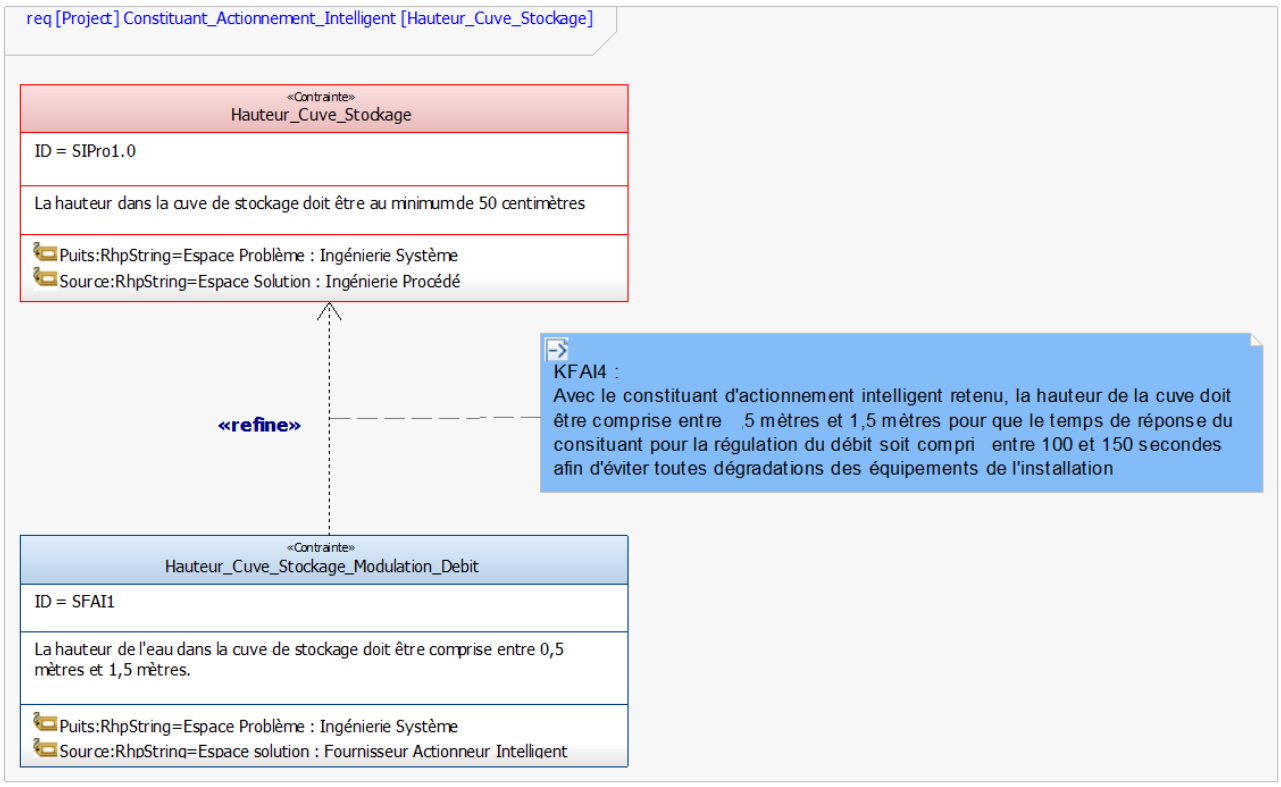


Figure 138 : Raffinement de l'exigence $S_{IPro1.0}$ à partir de la connaisse K_{FAI4} de l'espace-solution fournisseur d'actionnement intelligent

7.2.3.2. Définition d'un modèle exécutable de l'interaction logico-physique ciblée

Une fois la spécification du constituant (S_{FAI}) vérifiée, l'**espace-solution** FAI doit prescrire à minima à l'**espace-problème** d'ingénierie système les interfaces de son modèle afin que ce dernier puisse l'architecturer avec l'ensemble des modèles spécialistes (M_h) constitutifs du modèle global (M_H) de notre système d'intérêt CISPI. Cette architecture entre les différents modèles définit également la configuration de l'environnement de co-simulation système requis pour la validation de la spécification du constituant d'actionnement intelligent (S_{FAI}). En ce sens, le fournisseur d'actionnement intelligent translate une partie de sa connaissance contenue dans le modèle Simulink, vers un modèle SysML partagé entre l'ensemble des parties prenantes du projet pour définir les interfaces de son modèle avec les autres modèles spécialistes. Pour ce faire, le fournisseur d'actionnement intelligent étend sa connaissance (K_{FAI}) avec la table de correspondance présentée Chapitre 6 entre les éléments de modélisation Simulink et SysML.

Notons qu'à cette étape du processus de co-spécification système, la prescription en langage SysML du comportement du constituant d'actionnement intelligent n'est pas nécessaire à l'ingénieur système pour architecturer l'ensemble des modèles. En ce sens, le fournisseur d'actionneur intelligent restreint la translation de son modèle (M_{hIA}) d'un point de vue structurel :

- Dans un diagramme de définition de blocs SysML afin de prescrire les interfaces externes du modèle du constituant d'actionnement intelligent (Figure 137).
- Dans un diagramme interne de blocs SysML afin de prescrire les interfaces entre les modèles Simulink et Dymola composant le modèle global du constituant d'actionnement intelligent (Figure 140).

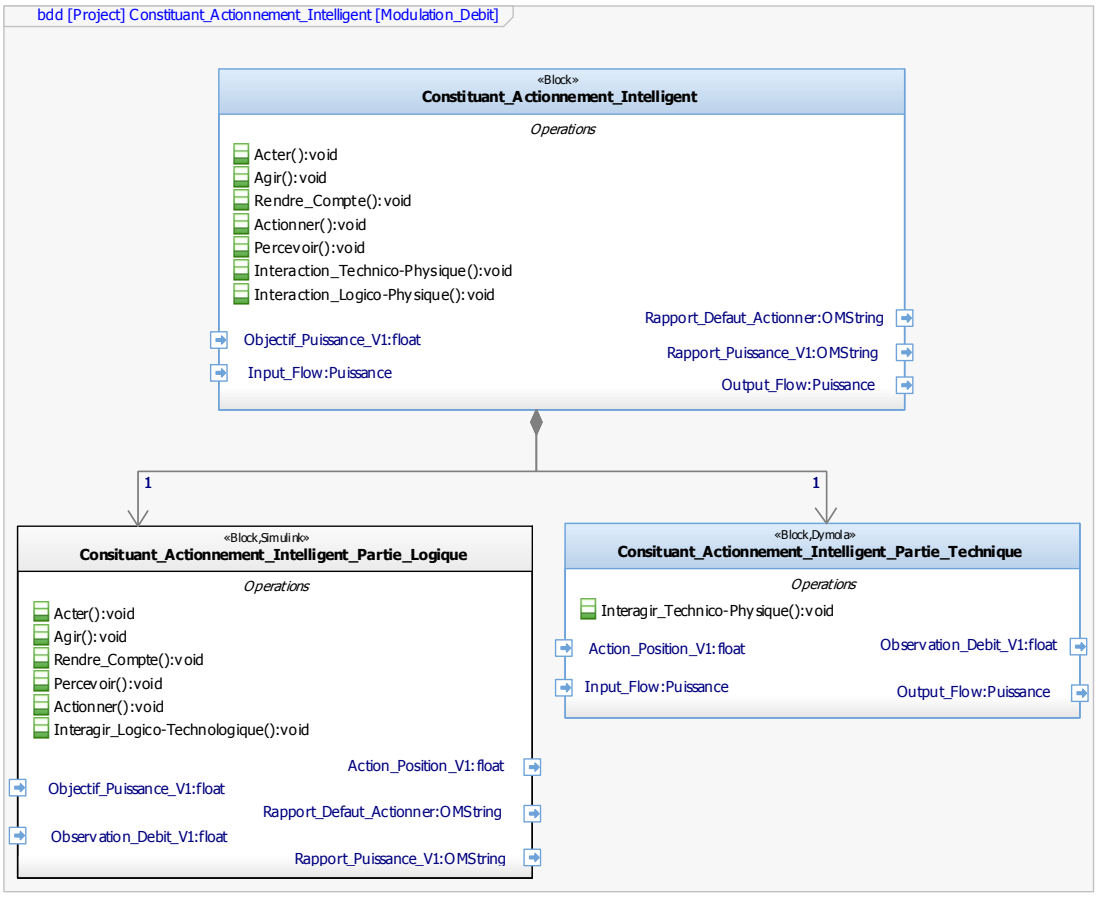


Figure 139 : Diagramme de définition de blocs (bdd) - Architecture externe du constituant d'actionnement intelligent de modulation de débit

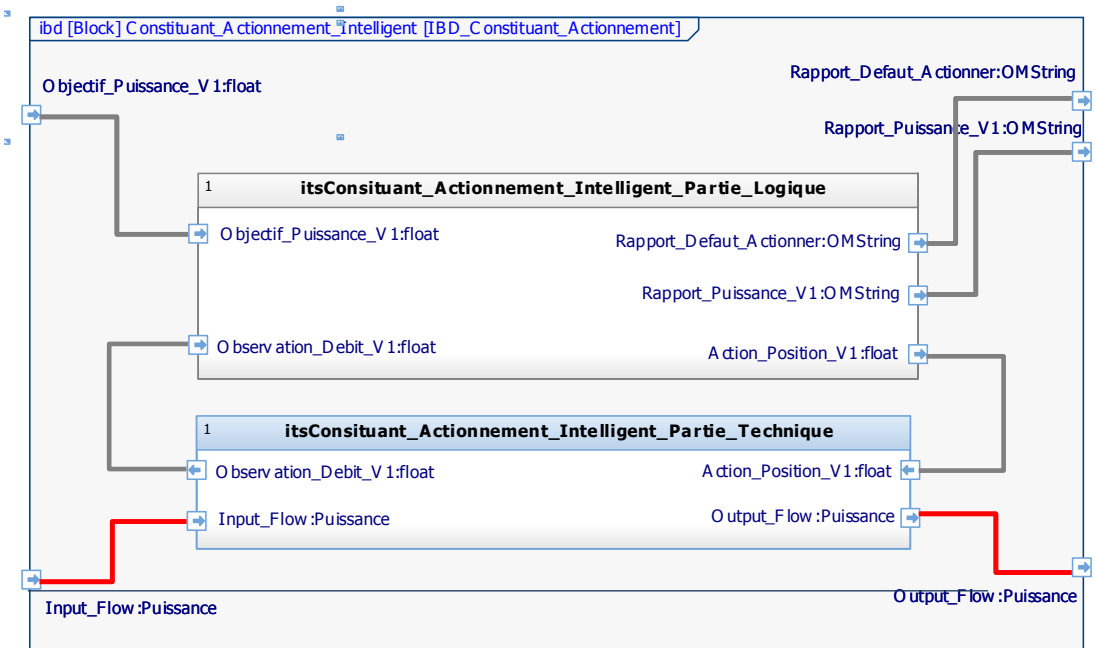


Figure 140 : Diagramme interne de bloc (ibd) - Structure interne du constituant d'actionnement intelligent de modulation de débit

7.3. Co-intégration système d'un constituant d'interaction logico-physique de modulation de débit

La validation système d'un élément, modèle ou composant, constitutif d'un modèle global, requiert son intégration dans une situation réelle conformément à la définition formelle d'une spécification ou la reflétant au mieux en pratique par niveaux de maturité technologique[61]. En ce sens, nous présentons dans le prochain paragraphe, la validation de la spécification (S_{FAI}) du constituant d'actionnement intelligent immergé dans un premier temps dans un environnement simulé puis dans l'environnement réaliste que constitue la plate-forme CISPI. Dans le second paragraphe, nous détaillons la réalisation du constituant d'actionnement intelligent en tant qu'agent de conduite intégré au SIAC de la plate-forme CISPI.

7.3.1. Intégration d'un modèle constitutif de l'interaction ciblée

Validation du modèle dans un environnement simulé

A cette étape du scénario de co-spécification système, l'**espace-problème** d'ingénierie système est en charge de valider la spécification (S_{FAI}) du constituant d'actionnement intelligent de modulation de débit. Cette activité est réalisée par co-exécution de modèles (Figure 141) afin de s'assurer que le modèle du constituant d'actionnement intelligent immergé dans le modèle de son environnement opérationnel manifeste les phénomènes requis par les exigences systèmes ($R_{IS_FAI_SysML}$).

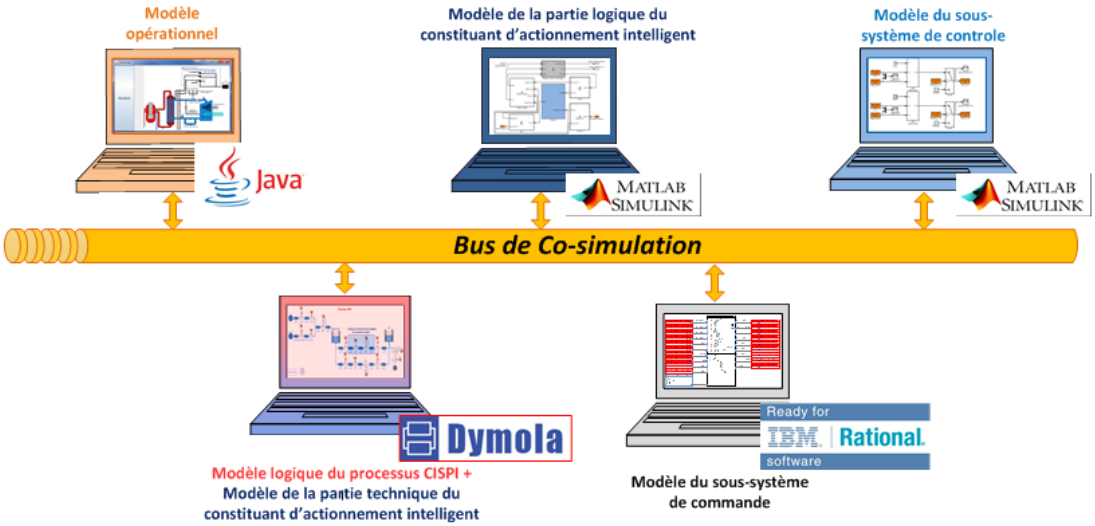


Figure 141 : Environnement de co-simulation système pour la validation de la spécification (S_{FAI}) du constituant d'actionnement intelligent de modulation de débit

Cela amène l'**espace-problème** d'ingénierie système à notifier à l'**espace-solution** d'ingénierie du procédé de raffiner son modèle du processus CISPI à un niveau fonctionnel (Cf. Chapitre 5) vers un modèle à un niveau logique, afin d'intégrer le modèle de la partie technique du composant d'actionnement intelligent dans l'environnement opérationnel de conduite. Notons que pour des raisons techniques liées à la co-simulation, les deux modèles (*environnement et constituant d'actionnement intelligent*) ont dû être intégrés au sein d'un même modèle (Figure 142).

[61] Ces travaux se situent aux niveaux 4 et 5 d'une échelle TRL

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage du système de conduite CISPI

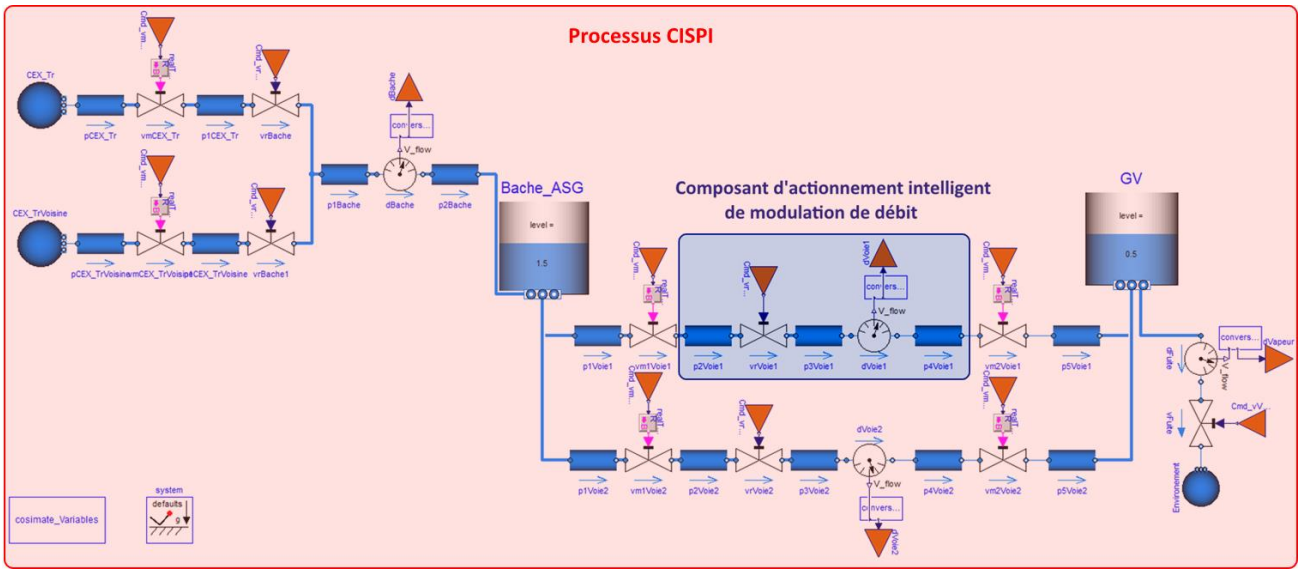


Figure 142 : Modèle logique du processus CISPI intégrant la partie physique du constituant d'actionnement intelligent

Une fois le modèle du processus CISPI raffiné à un niveau logique, l'**espace-solution** d'ingénierie du procédé prescrit à l'**espace-problème** d'ingénierie système un diagramme interne de blocs SysML décrivant les interfaces externes du modèle de processus de la plate-forme CISPI, ainsi que les flux échangés entre le constituant d'actionnement intelligent et le processus CISPI.

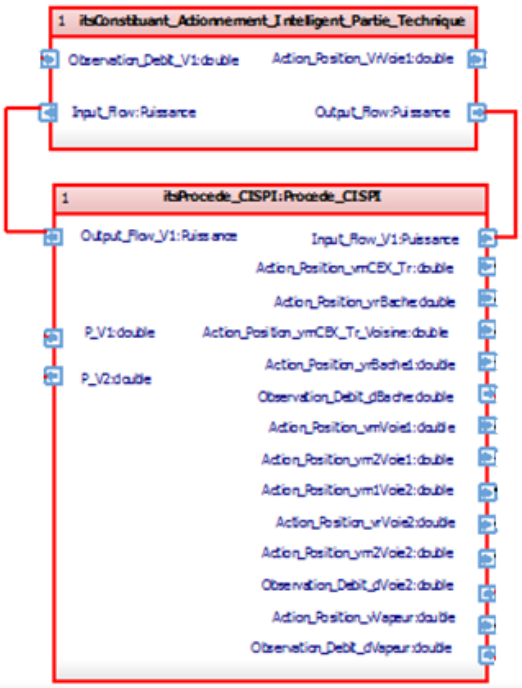


Figure 143 : Spécification (S_{IPhy_SysML}) des interfaces externes du processus CISPI à un niveau d'abstraction logique

En ce sens, l'**espace-problème** d'ingénierie système possède suffisamment de connaissances sur l'ensemble des modèles d'ingénieries spécialistes constitutifs de la plate-forme CISPI pour les architecturer les uns avec les autres afin de définir un modèle global du système étudié. Cette architecture est modélisée dans un diagramme interne de blocs SysML (Figure 144).

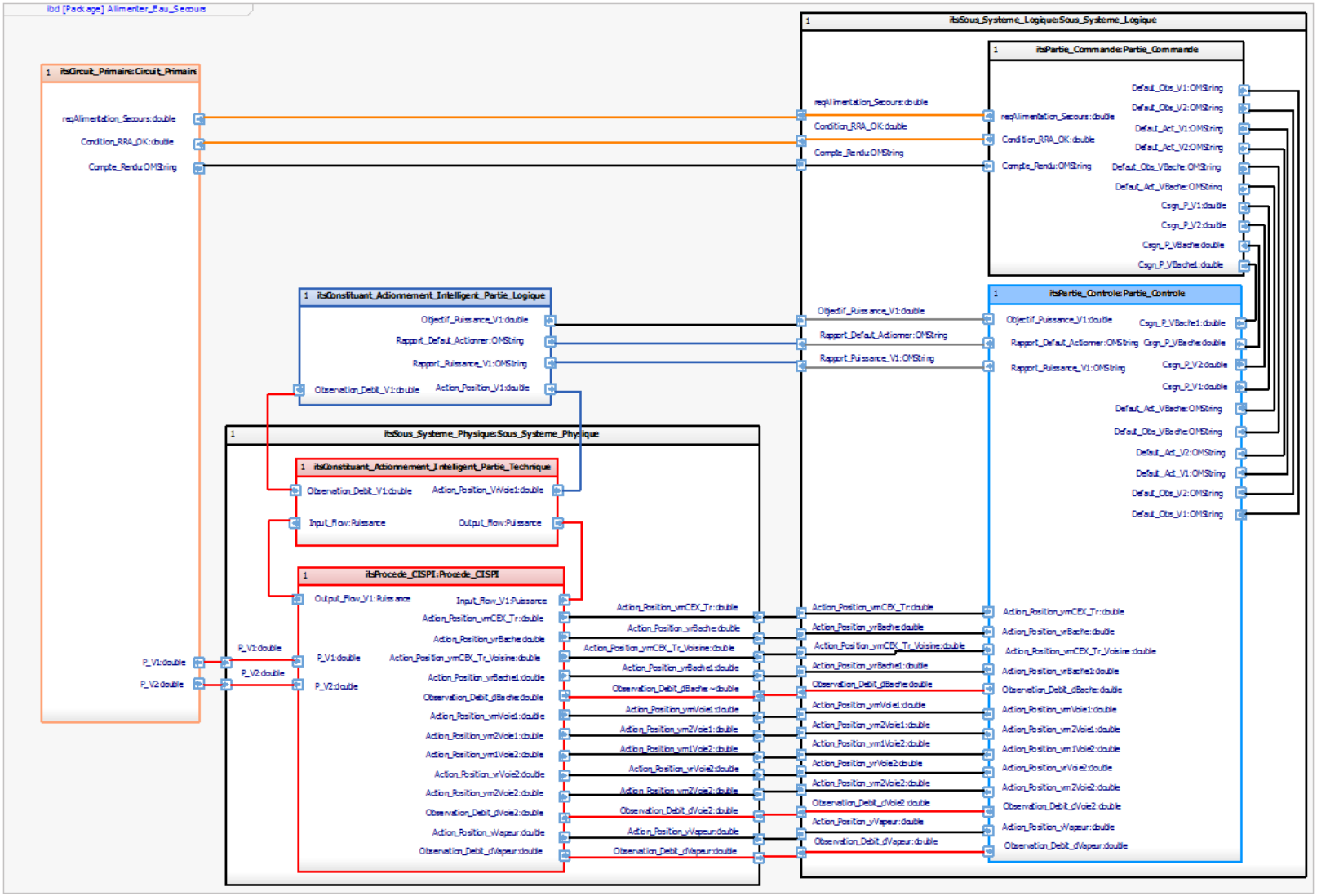


Figure 144 : Définition de l'architecture de co-simulation système (*Bloc d'interaction de détection sonore + bloc logique + bloc physique*) par l'espace-problème d'ingénierie système

Le comportement du constituant d'actionnement intelligent est alors validé par co-exécution de modèles selon différents scénarios de tests afin que l'ingénieur système s'assure que le système global manifeste les événements requis par l'environnement (***R_{IS_FAI1}***, ***R_{IS_FAI2}***, ***R_{IS_FAI3}***, ***R_{IS_FAI4}***). Nous présentons ci-dessous, un scénario de test relatif à l'évaluation des performances du constituant d'actionnement intelligent lors de la régulation de puissance d'alimentation en eau de secours à travers la voie 1 de la plate-forme CISPI en cas de situation incidentelle ou accidentelle.

En analysant les traces d'exécution (Figure 145) de la co-simulation, nous constatons que lors de l'apparition d'un événement caractérisant une situation accidentelle ou incidentelle (*correspondant ici à une augmentation soudaine de la puissance résiduelle à l'intérieur du circuit primaire à t = 100 secondes*), le constituant d'actionnement intelligent est automatiquement mis en service pour faire évoluer et maintenir la puissance d'alimentation en eau de secours autour de 50 Watts (***Objectif_Puissance_V1***). Le constituant régule alors la puissance d'alimentation (***Rapport_Puissance_V1***) autour de cette consigne, avec un temps de réponse satisfaisant les contraintes opérationnelles (*temps de réponse = 150 secondes*) ce qui entraine une diminution de la puissance résiduelle. Une fois les conditions de mise en service du système RRA (*Refroidissement du Réacteur à l'Arrêt*) atteinte, le constituant d'actionnement intelligent se met automatiquement hors service (*à t = 375 secondes,* ***Objectif_Puissance_V1 = 0***). Ainsi, en exécutant un ensemble de scénarios de test, **l'espace-solution** d'ingénierie système valide le comportement du constituant d'actionnement intelligent.

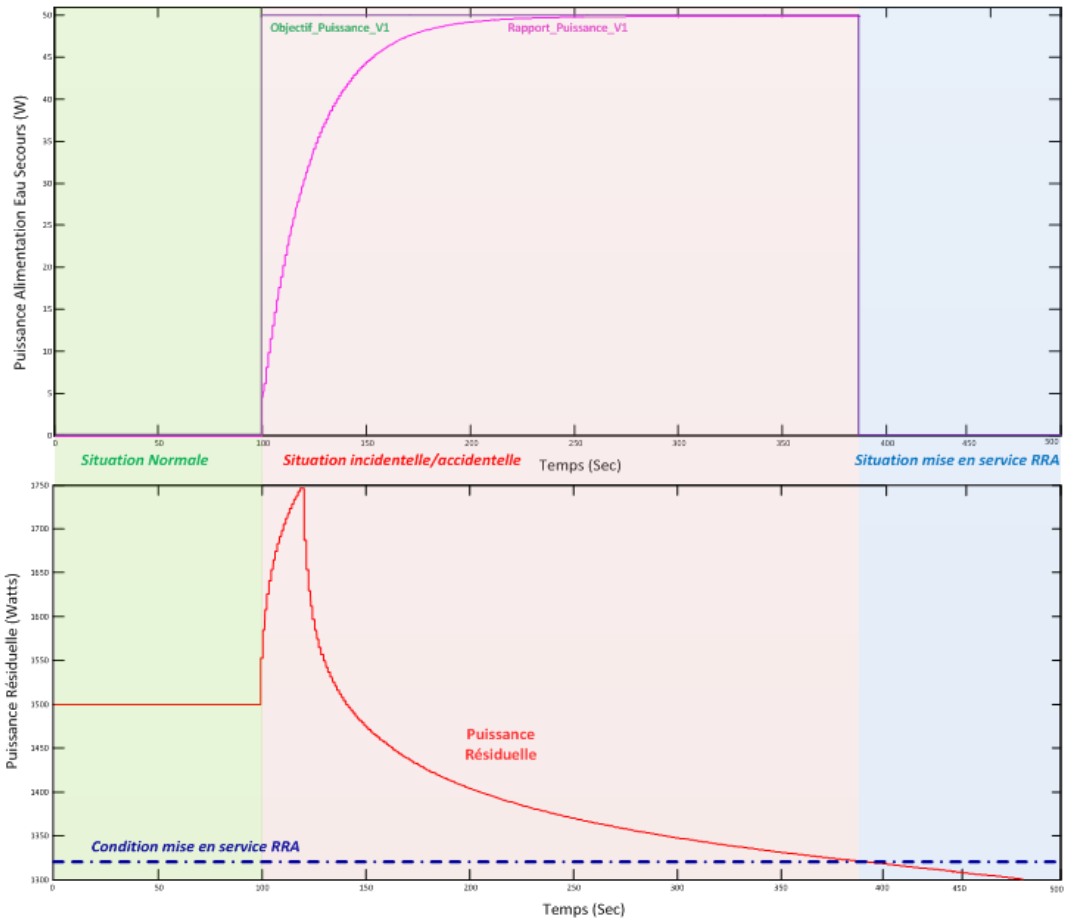


Figure 145 : Trace d'exécution du scénario de test pour la validation du constituant d'actionnement intelligent de modulation de puissance

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage intelligents du système CISPI

Validation du modèle d'actionnement intelligent dans un environnement représentatif de la réalité

Notre environnement de co-spécification système exécutable basée sur des modèles offre à l'ingénieur système des moyens techniques pour la validation de la spécification du constituant d'actionnement intelligent. Dans un processus d'ingénierie conduisant à l'intégration de ce constituant technique en tant que composant de la plate-forme CISPI, il est nécessaire d'éprouver la technologie dans des environnements de tests de plus en plus réalistes afin d'en évaluer son degré de maturité (*selon l'échelle TRL* (*Technology Readiness Level (Mankins, 1995)*). Cela nous a conduits à remplacer, dans notre environnement de co-simulation, le modèle[62] du processus de la plate-forme CISPI (*Model in the Loop*) par le vrai processus physique en exploitation (*Hardware/Software in the Loop*) (Figure 146). Cette étape de validation nous permet alors d'atteindre le niveau 5 (*Validation du composant dans un environnement représentatif*) de l'échelle TRL.

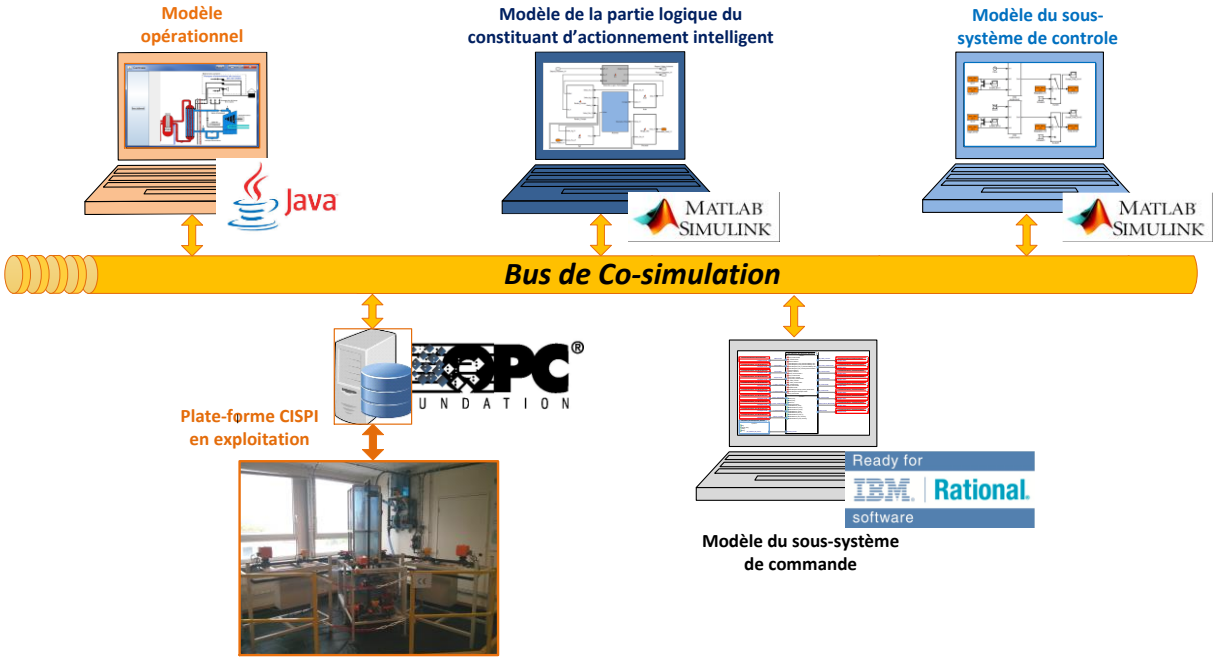


Figure 146 : Environnement de co-validation système pour la validation et l'évaluation du niveau de maturité de la spécification (S_{FAI}) du constituant d'actionnement intelligent de modulation de débit

D'un point de vue technique pour effectuer le couplage entre le processus existant de la plate-forme CISPI avec le bus de co-simulation, il a été nécessaire de développer un module complémentaire se connectant au serveur OPC[63] de la plate-forme CISPI. Ce module a pour mission d'aller lire ou écrire les variables automates définies au niveau du serveur OPC (OLE for Process Control) et de diffuser ces données à travers le bus de co-simulation système. Ce module a été développé en langage Java en faisant appel à la librairie « Cosimate » permettant d'établir la connexion avec le bus de co-simulation retenu pour notre cas d'application (Cf. Annexe 7-5).

[62] Un modèle est une représentation homomorphe du système étudié, qui en donne une vue partielle plus ou moins abstraite et permet d'en étudier certaines caractéristiques. En ce sens, le modèle de l'environnement du constituant d'actionnement intelligent représentant le processus CISPI, ne modélise pas l'ensemble des phénomènes émergents.

[63] Un serveur OPC fournit une méthode permettant à différents logiciels d'accéder aux données de dispositifs de contrôle de processus comme un automate programmable industriel

En ce sens, l'**espace-problème** est en mesure d'éprouver le comportement du constituant d'actionnement intelligent de modulation de puissance en co-simulation, en couplant le modèle logique du constituant avec le sous-système technique (vanne + débitmètre) elle-même reliée à la plate-forme CISPI. Cette phase de test, a conduits à une nouvelle itération de spécification entre l'ingénieur système et le fournisseur d'actionnement intelligent. En effet, le modèle du processus de la plate-forme CISPI (Figure 142) ne prenant pas en compte les frottements de l'eau dans les canalisations ainsi que le temps de réponse d'ouverture et de fermeture de la vanne (*environ cinq secondes pour l'ouverture complète*), le paramétrage de la fonction de contrôle composant le constituant d'actionnement intelligent n'était pas optimisée (Figure 147a). Cette nouvelle itération a donc permis de prendre en compte ces contraintes techniques en raffinant la spécification (S_{FAI}) du constituant d'actionnement intelligent afin de satisfaire les exigences systèmes de performances (R_{IS_FAI1}, R_{IS_FAI2}, R_{IS_FAI3}) (Figure 147b).

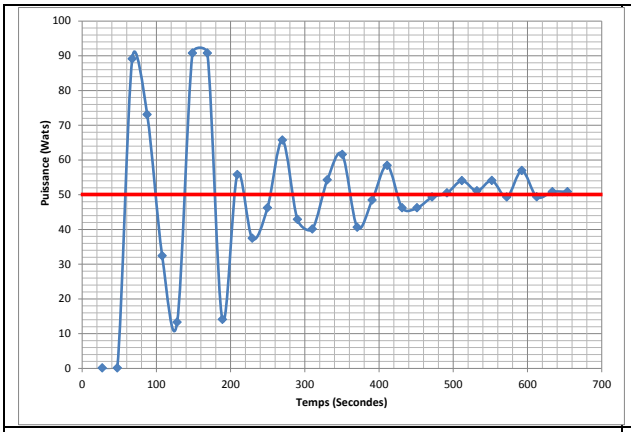


Figure 147a : Avant la prise en compte des contraintes techniques de la plate-forme CISPI

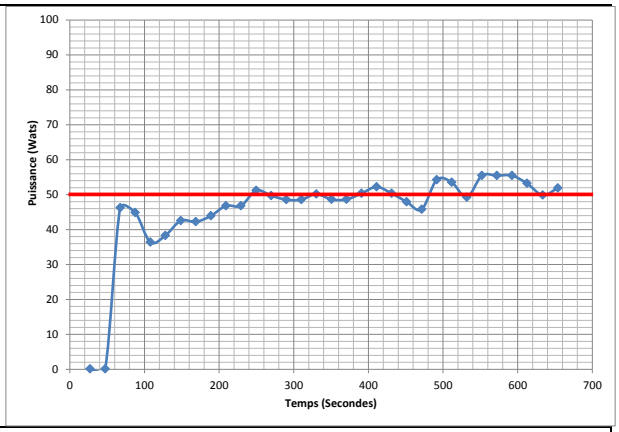


Figure 147b : Après la prise en compte des contraintes techniques de la plate-forme CISPI

Figure 147 : Trace d'exécution du constituant d'actionnement intelligent de modulation de puissance

7.3.2. Intégration d'un composant constitutif de l'interaction ciblée

Cette nouvelle itération de spécification, entre l'**espace-problème** d'ingénierie système et l'**espace-solution** fournisseur d'actionneur intelligent, a pour objectif de faire évoluer la maturité du constituant technique d'actionnement au niveau 6 (*Démonstration d'un prototype dans un environnement représentatif*) de l'échelle TRL. Ainsi, en se basant sur les différentes exigences systèmes, ainsi que les précédents modèles de spécification, le fournisseur réalise un prototype du constituant d'actionnement intelligent de modulation de puissance. Ce prototype s'intégrant au SIAC est immergé dans l'environnement opérationnel de conduite de la plate-forme CISPI afin de s'assurer qu'il manifeste les événements requis par l'environnement.

Notons que les exigences (Figure 132) :

- ($R_{FAI6.1}$) : « *Le débitmètre « FI433 » de la plate-forme CISPI de marque « GEORG FISHER +GF+ » et de type « Signet 8850-1 » doit être utilisé en tant que composant du constituant d'actionnement intelligent*»
- ($R_{FAI6.2}$) : « *L'électrovanne « 431VM » de la plate-forme CISPI de marque « GEORG FISHER +GF+ » et de type « EA11 » doit être utilisée en tant que composant du constituant d'actionnement intelligent*»

prescrivent comment doit être réalisée la partie technique du constituant d'actionnement intelligent. En ce sens, nous nous focalisons sur la réalisation de la partie logique du constituant, contraint par l'exigence R_{FAI5} : « *Le constituant d'actionnement intelligent doit être réalisé à l'aide de la bibliothèque JADE SHARP Add On* ». En effet, le SIAC implémenté sur la plate-forme CISPI a été développé durant les travaux de thèse de (Dobre, 2010) en se basant sur la plate-forme multi-agents JADE[64] (Java Development Framework Agent) (Bellifemine, Poggi, & Rimassa, 1999). En d'autres termes, le FAI doit réaliser un prototype d'OLT (Objet Logique Technique) en tant qu'agent JADE du SIAC et reflétant le comportement des OMTs (Objets Matériels Techniques) correspondant au débitmètre « FI433 » et à l'électrovanne « 431VM » de la plate-forme CISPI.

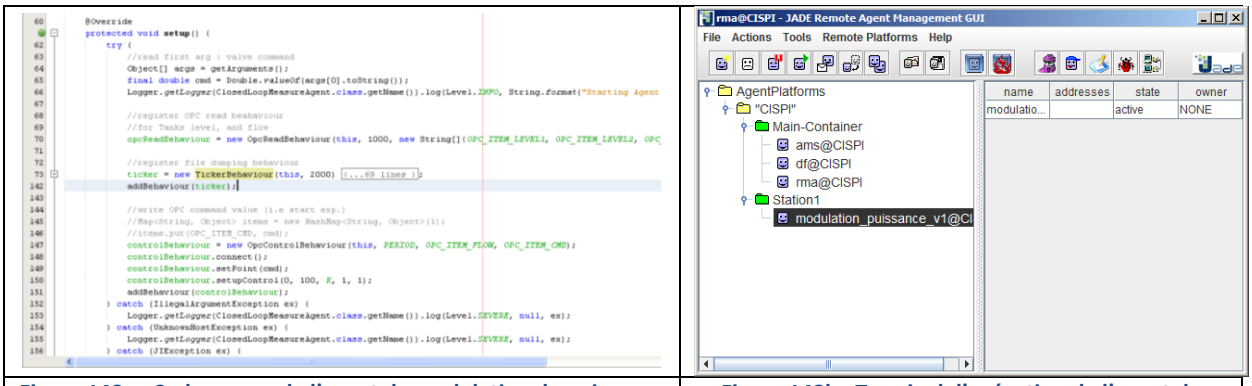


Figure 148a : Code source de l'agent de modulation de puissance | **Figure 148b : Terminal d'exécution de l'agent de modulation de puissance**

Figure 148 : Réalisation de la partie logique du constituant d'actionnement intelligent en tant qu'agent de la plate-forme JADE implémentant le SIAC

Une fois le prototype du composant d'actionnement intelligent de modulation de puissance réalisé (Figure 148), l'ingénieur système procède à son intégration au niveau de la plate-forme CISPI, afin de fournir la fonction de modulation de puissance d'alimentation en eau de secours requise par l'environnement. Ainsi la partie logique du constituant (*Agent de contrôle*) est implémentée au niveau de la plate-forme multi-agent JADE s'exécutant sur le poste de « ***Conduite et de Supervision*** » de la plate-forme CISPI. Quant à la partie technique, le FAI réutilise l'actionneur (*électrovanne « 431VM »*) et le capteur (*débitmètre « FI433 »*) de la station 1 de la plate-forme CISPI (Figure 149). Le déroulement des différents scénarios de tests permet ainsi à l'**espace-problème d'ingénierie**-système d'évaluer la maturité du constituant d'actionnement intelligent dans un environnement représentatif d'une situation de conduite.

[64] JADE est un cadre logiciel mise en œuvre dans le langage JAVA, pour simplifier la mise en œuvre de système multi-agents à travers un middleware simplifie qui est conforme aux spécifications du FIPA (Foundation for Intelligent Physical Agents) et à travers un ensemble d'outils graphiques qui prennent en charge les phases de débogage et de déploiement. (http://jade.tilab.com/)

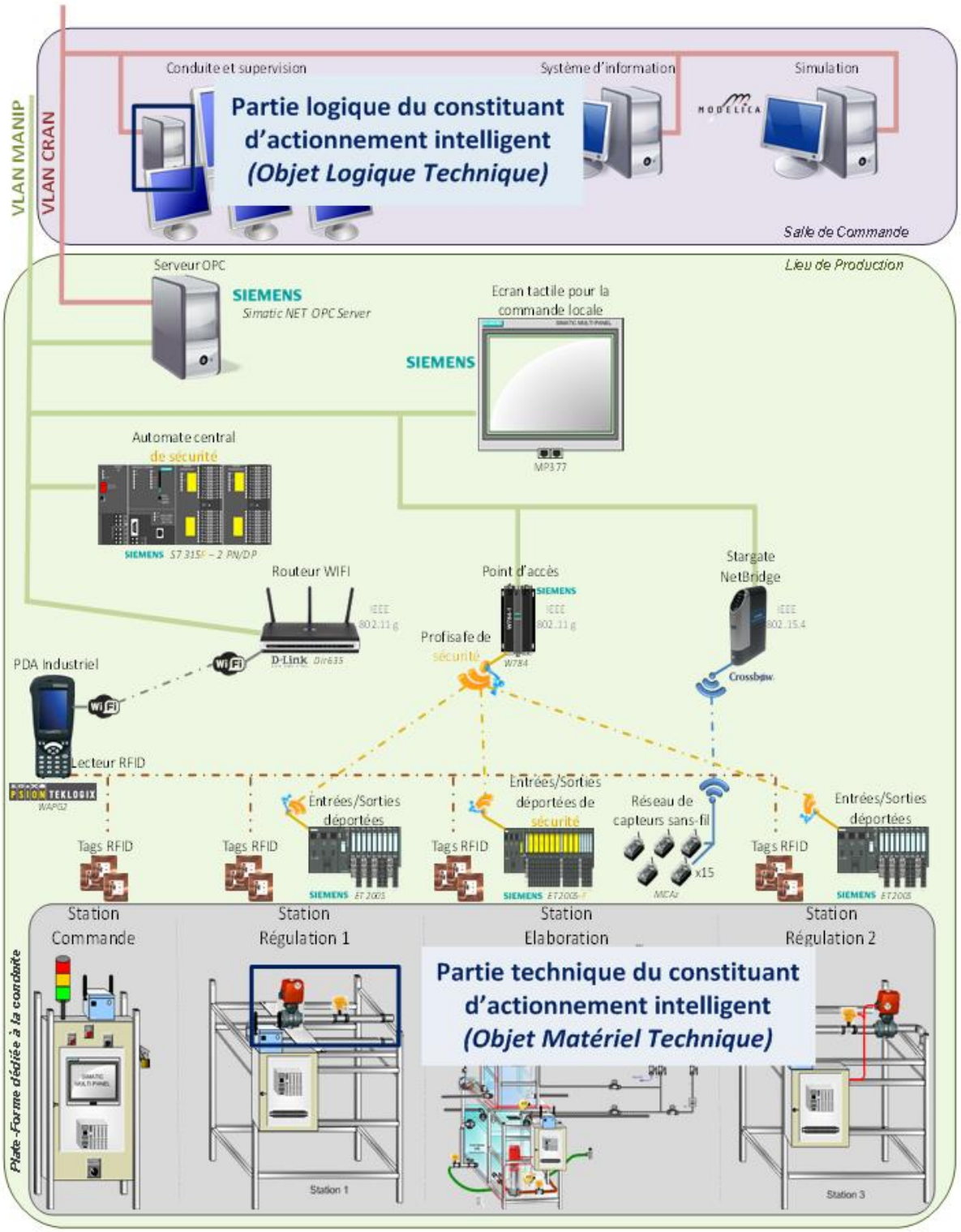


Figure 149 : Intégration du composant d'actionnement intelligent avec le SIAC de la plate-forme CISPI

7.4. Conclusion : Co-spécification exécutable d'un modèle partiel constitutif de la partie d'« actionnement intelligent » ciblée

L'objectif de ce chapitre était de spécifier un élément sous forme de modèles puis de composants, qui soient constitutifs d'une interaction logico-physique du système CISPI en se limitant à une fonction de modulation de puissance d'alimentation en eau de secours. Ce constituant technique d'actionnement intelligent a été co-spécifié entre l'**espace-problème** d'ingénierie système et l'**espace-solution** FAI, nécessitant également l'intervention de l'**espace-solution** d'ingénierie du procédé pour raffiner le modèle du processus CISPI. Cette spécification est le résultat du raffinement d'une interaction logico-physique selon une interprétation du patron architectural CMMS-IAMS ayant conduit à la définition d'un agent de conduite s'intégrant au SIAC de la plate-forme CISPI. Au cours de cette définition, nous avons montré l'intérêt de la CSM pour définir rapidement et sans connaissances du processus à conduire le sous-système logique du constituant d'actionnement intelligent.

Les technologies numériques existantes en termes de co-simulation nous ont permis de valider et d'évaluer le degré de maturité de ce constituant d'actionnement intelligent immergé dans des environnements de plus en plus réalistes, pour atteindre le niveau 6 de l'échelle TRL à travers la réalisation d'un prototype.

De la même manière que nous venons de co-spécifier un constituant technique d'actionnement intelligent, nous nous focalisons dans le prochain chapitre sur un constituant physiologique de détection sonore constitutive de l'interaction logico-physiologique étudiée.

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage intelligents du système CISPI

8. Chapitre 8 : Co-spécification exécutable d'un constituant physiologique de mesurage du système de conduite CISPI

8.1. Introduction : Partie « mesurage intelligent » du système de conduite CISPI

A cette étape de notre scénario de co-spécification système, notre intérêt se porte sur le raffinement d'une fonction MI opérée par un agent humain en tant qu'observateur d'une situation de conduite. Notre interprétation de cette fonction MI s'appuie principalement sur des travaux relatifs à la « spécification d'exigences physico-physiologiques en ingénierie d'un système support de maintenance aéronautique » auxquels nous avons contribué en tant que spécialiste de modélisation-simulation (Lieber, Dupont, Bouffaron, & Morel, 2013) dans le cadre du doctorat de (Lieber, 2013). Plus précisément, notre étude se focalise sur le comportement de l'interaction (Figure 150) entre les parties logique et physiologique du système de conduite en co-spécification afin de s'assurer au plus tôt que les phénomènes affordés soient bien détectés pour être bien perçus (Hartson, 2003).

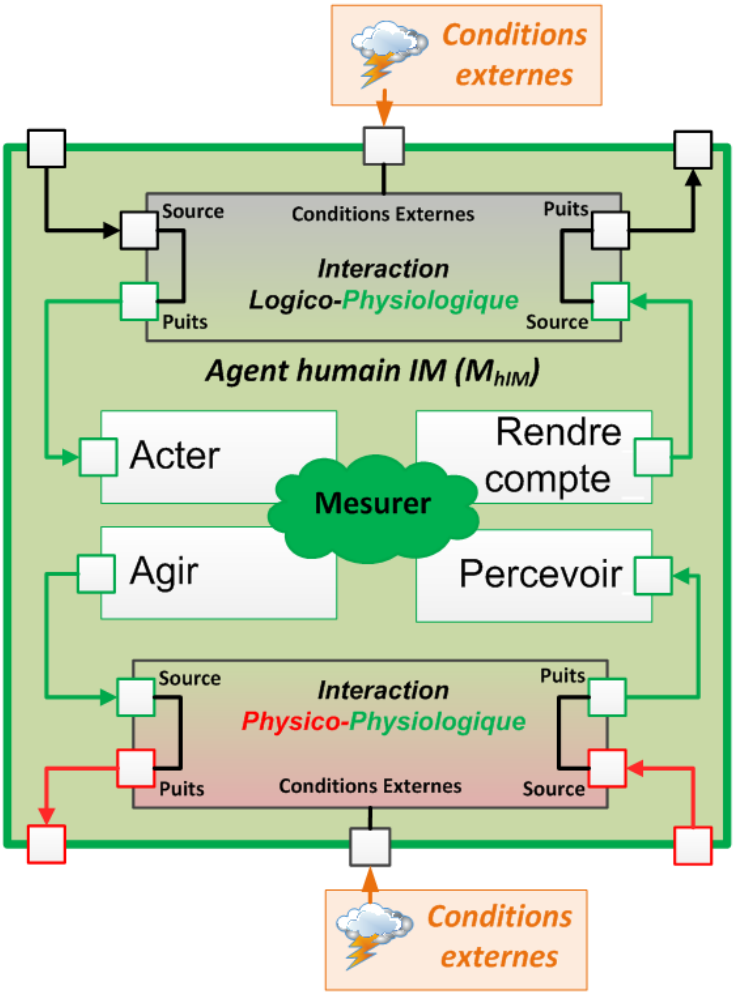


Figure 150 : Patron architectural de Mesurage Intelligent par un Agent Humain

De façon complémentaire, cette « bonne détection » pour « bien percevoir » afin de « bien agir » s'applique aussi à l'interaction entre les parties physique et physiologique du système de conduite. L'ensemble de ces

activités contribue à la construction du schéma mental de conduite dont l'importance a été soulignée par (Galara, 2011) :

- **Acter** : Au cours de la conduite, les opérateurs reçoivent des objectifs pouvant provenir d'autres opérateurs en salle de commande ou sur le terrain, d'alarmes, de procédures, d'écrans de contrôle… Compte tenu de ces objectifs, l'opérateur se réfère à son schéma mental de conduite, pour supprimer les ambiguïtés de transmission (*fonctionnelles, temporelles, ordre contre ordres*) et s'assurer de leurs cohérences vis à vis des contraintes de conduite. Un certain nombre de ces objectifs requièrent que l'opérateur observe son environnement afin de percevoir la situation globale de conduite. Ceci est particulièrement vrai lors des rondes d'observations effectuées par les agents de terrain.
- **Percevoir** : A partir des objectifs, l'opérateur observe la situation globale de conduite pour se construire une représentation de l'état dans lequel se situe l'installation, un équipement, le procédé,... Pour élaborer ces schémas mentaux, les opérateurs sont amenés à prendre en compte des mesures issues de capteurs ou des perceptions sensorielles (*ouïe, odorat, vue, toucher, gout*) de leur environnement de travail.
- **Agir** : La perception est dépendante de l'action. Plus précisément le concept d'affordance, issu de la théorie écologique de la perception de (Gibson, 1977), souligne le caractère actif de la perception, c'est-à-dire orientée en fonction d'un but à atteindre. Ainsi, pour se construire une représentation du réel, les opérateurs se déplacent dans leur environnement, balayent du regard l'espace, orientent leurs sens en direction des sources sonores, touchent les objets,….
- **Rendre compte :** En se basant sur son schéma mental et sur les objectifs, l'opérateur restitue sa perception de la situation réelle de l'installation, des équipements ou du procédé. Notons que cette restitution reflète le réel selon le point de vue de l'opérateur. Il faut donc une communication pertinente entre les opérateurs, justifiant numériquement le PLM, pour s'assurer que les équipements, les sources d'informations, l'état présumé et les niveaux de compréhension soient cohérents pour que la situation soit comprise de la même manière.
- **Mesurer :** Cette activité en interaction avec les 4 précédentes concerne les processus cognitifs mis en œuvre par un agent humain en tant qu'observateur (mesureur). Nous interprétons cette activité comme une situation d'activité lente au sens de (Berthoz, 1997) laissant une place centrale au cerveau qui reçoit (cortex auditif, visuel …), traite, analyse et synthétise les informations lui permettant de simuler l'action et de prédire ses conséquences avant de commander l'action. Ces simulations et prédictions reposent ainsi sur un ensemble de règles d'interprétations, issues de l'apprentissage et de la mémorisation des actions passées.

En ce sens, nous avons exploré au cours de nos travaux de thèse un ensemble d'interactions élémentaires de perceptions sensorielles entre un agent humain de conduite et son environnement de travail :

- Interaction tactile :
    - Perception des vibrations permettant de caractériser l'écoulement de l'eau dans des conduites, le bon fonctionnement d'un moteur en rotation, ….
    - Perception thermique (Schmidt, 1999): En cas d'obstruction d'une conduite à la sortie d'une pompe, la pression du liquide refoulé par cette dernière va augmenter et par conséquent la température du liquide aussi. La perception d'une différence de température par rapport aux conditions nominales, peut ainsi révéler une dégradation, ou une défaillance des équipements.
- Interaction visuelle : Perception de la couleur et du clignotement des boutons TPL (Tourner-Pousser Lumineux) représentant l'état de fonctionnement d'un équipement (Prouillac, Lerat, & Janior)

- Interaction sonore :
  - Perception de sifflements lors des opérations d'éventages en phase de maintenance
  - Perception des bruits des pompes pouvant caractériser une dégradation de l'équipement (J. Morel, 1992)
  - Perception d'alarme pour alerter des situations incidentelles ou accidentelles
  - Perception de situation normale de conduite : Par exemple, en continuité du chapitre 7, nous aurions pu nous intéresser au bruit de l'écoulement de l'eau à travers une conduite en tant que retour d'observation de l'action d'ouverture ou de fermeture de la vanne pouvant être contrôlée à distance.

Dans la suite de ce chapitre, nous nous sommes focalisés sur l'une des interactions sonores. En effet, les agents humains en situation d'exploitation sont amenés à percevoir des sons susceptibles de déclencher des procédures apprises. Il peut s'agir d'une alarme sonore, d'un bruit caractéristique d'un quelconque dysfonctionnement ou tout simplement d'une signature sonore de bon fonctionnement.

8.2. Co-spécification système d'un modèle d'interaction physico-physiologique de détection sonore

Pour des raisons opérationnelles de démonstration, et en tenant compte des données disponibles d'un point de vue biologique, nous nous sommes intéressés à une interaction -physico-physiologique de détection d'une alarme sonore entre un constituant technique « ***Source*** » et un constituant humain[65] « ***Puits*** » (Figure 151). Nous dépassons la simple étude d'une interface technico-humaine pour spécifier le comportement d'un bloc « interaction » sous la forme d'un modèle exécutable dans notre processus global de co-spécification système de la plate-forme CISPI. L'objectif de cette étude est de spécifier les propriétés physiques requises par le constituant technique « ***Source*** » afin de stimuler correctement le constituant humain « ***Puits*** » au regard de ses capacités physiologiques. Bien que cette étude ne soit pas suffisante pour prendre en compte l'ensemble des facteurs humains, elle définit cependant un premier niveau d'exigences dans la définition d'un modèle exécutable constitutif d'une interaction que nous qualifions de physico-physiologique en raison de la nature des milieux traversés.

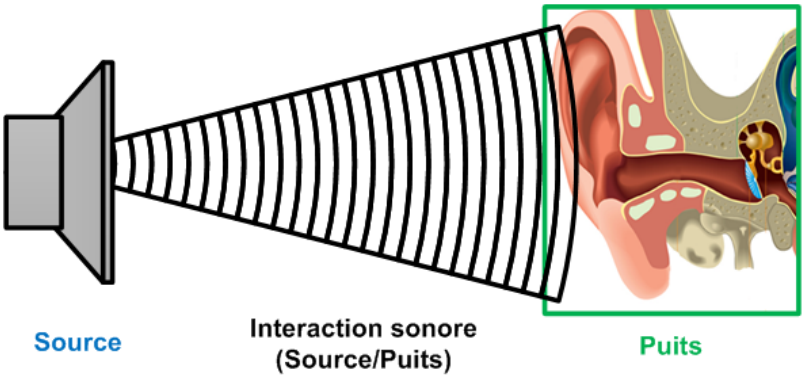


Figure 151 : Interaction logico-physiologique de détection sonore

[65] Nous substituons « constituant » à « agent » pour marquer que ces éléments et leurs modèles sont constitutifs du système CISPI et de son modèle global

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage intelligents du système CISPI

Nous présentons dans la suite de ce chapitre la spécification descriptive d'un modèle d'exigences de cette interaction par l'**espace-problème** d'ingénierie système requérant d'un **espace-solution** d'ingénierie physiologique un ensemble de connaissances (K_{IPhy}). La spécification prescriptive résultante est dérivée de la construction d'un modèle exécutable de la perception sonore chez l'humain, afin de déterminer les conditions nécessaires, mais non suffisantes, au bon stimulus des activités perceptives de conduite. Ce modèle exécutable est ainsi constitutif de cette interaction dans l'ensemble des modèles constituant notre modèle global du système CISPI.

8.2.1. Spécification descriptive d'un modèle d'exigences constitutif d'une interaction sensorielle de détection sonore

8.2.1.1. Eléments de connaissances d'une interaction sensorielle

En tant qu'ingénieur système, cette interaction physico-physiologique nous pose le problème de savoir comment fonctionnaliser un objet technique « Source » pour tenir compte des contraintes physiologiques de l'humain. Cette interprétation se base sur un ensemble de travaux (G. Morel, et al., 2013) situant le cadre théorique des affordances (Gibson, 1977), (Gibson, 2014) en tant que **connaissance** du domaine d'ingénierie système, et soulignant l'importance des degrés d'affordance que doivent porter les constituants techniques pour interagir avec les constituants humains (Cf. Annexe 8-1 - Figure 225) .

Plus précisément une affordance fonctionnelle requiert une condition nécessaire pour l'affordance sensorielle relative à la quantité physique requise pour bien stimuler « la perception » (Tableau 18) (Hartson, 2003). En interprétant ces travaux, nous pouvons reformuler le problème à résoudre selon : « comment bien fonctionnaliser le constituant technique pour qu'il afforde le bon signal au constituant humain ? »

Type d'affordance	Description	Exemple
Cognitive	Caractéristique de conception qui aide un utilisateur à savoir quelque chose	Le timbre caractéristique d'une alarme sonore permet de prévenir l'opérateur d'une situation incidentelle ou accidentelle.
Physique	Caractéristique de conception qui aide un utilisateur à effectuer une action	Le rythme de l'alarme est le paramètre le plus pertinent pour décrire différentes catégories d'urgence allant de l'alerte à la confirmation d'un évènement (Suied, 2007)
Sensorielle	Caractéristique de conception qui aide un utilisateur à percevoir quelque chose	La puissance sonore d'une alarme permet de caractériser les conditions nécessaires mais non suffisantes de perception de l'alarme (Cf. suite du chapitre)
Fonctionnelle	Caractéristique de conception qui aide un utilisateur à accomplir une procédure de conduite	En situation incidentelle, l'alarme sonore est déclenchée pour alerter l'opérateur de conduite (*qui va suivre une procédure de conduite pour rétablir le système dans un état sûr de fonctionnement*).

Tableau 18 : Types d'affordances pour une interaction physico-physiologique de détection sonore. Adapté de (Hartson, 2003)

8.2.1.2. Définition des exigences de l'interaction sensorielle ciblée

A cette étape du scénario de co-spécification système, le domaine d'ingénierie système étudie la situation globale de détection d'une alarme sonore par un agent humain. En ce sens, il se pose le problème de caractériser l'affordance sensorielle d'interaction sonore selon des critères mesurables en tant que conditions nécessaires mais non suffisantes pour l'affordance fonctionnelle. L'**espace-solution** d'ingénierie système n'ayant pas les connaissances pour spécifier ce type d'interaction formalise alors ce problème ($P_{IS_Phy_SysML}$) **dans son espace-problème** sous la forme d'un diagramme d'exigences ($R_{IS_Phy_SysML}$) (Figure 101). Au cours d'une première itération de spécification, ces exigences systèmes sont notifiées à l'**espace-solution** d'ingénierie physiologique qui est alors en charge de prescrire une solution (S_{IPhy_SysML}) dérivée d'un modèle exécutable (M_{hIM}) de l'interaction étudiée.

De façon plus détaillée, les exigences décrites à l'**espace-solution** d'ingénierie physiologique {$R_{IS_Phy1.1}$, $R_{IS_Phy1.2}$} sont raffinées à partir d'une exigence système ($R_{IS_Phy1.1}$) selon un ensemble de connaissances {K_{IO3}, $K_{IO4.1}$, $K_{IO4.2}$} provenant de l'espace solution d'ingénierie opérationnelle. Cette exigence ($R_{IS_Phy1.1}$) est une exigence physiologique de conduite décrivant l'interaction de perception sonore entre la plate-forme CISPI et les opérateurs de conduite en cas de situation incidentelle : « *débordement de la cuve de stockage* ». Notons également qu'une exigence de contrôle ($R_{IS_Aut_Cont2}$) est dérivée de cette exigence ($R_{IS_Phy1.1}$) afin de déterminer les conditions de déclenchement de l'alarme sonore.

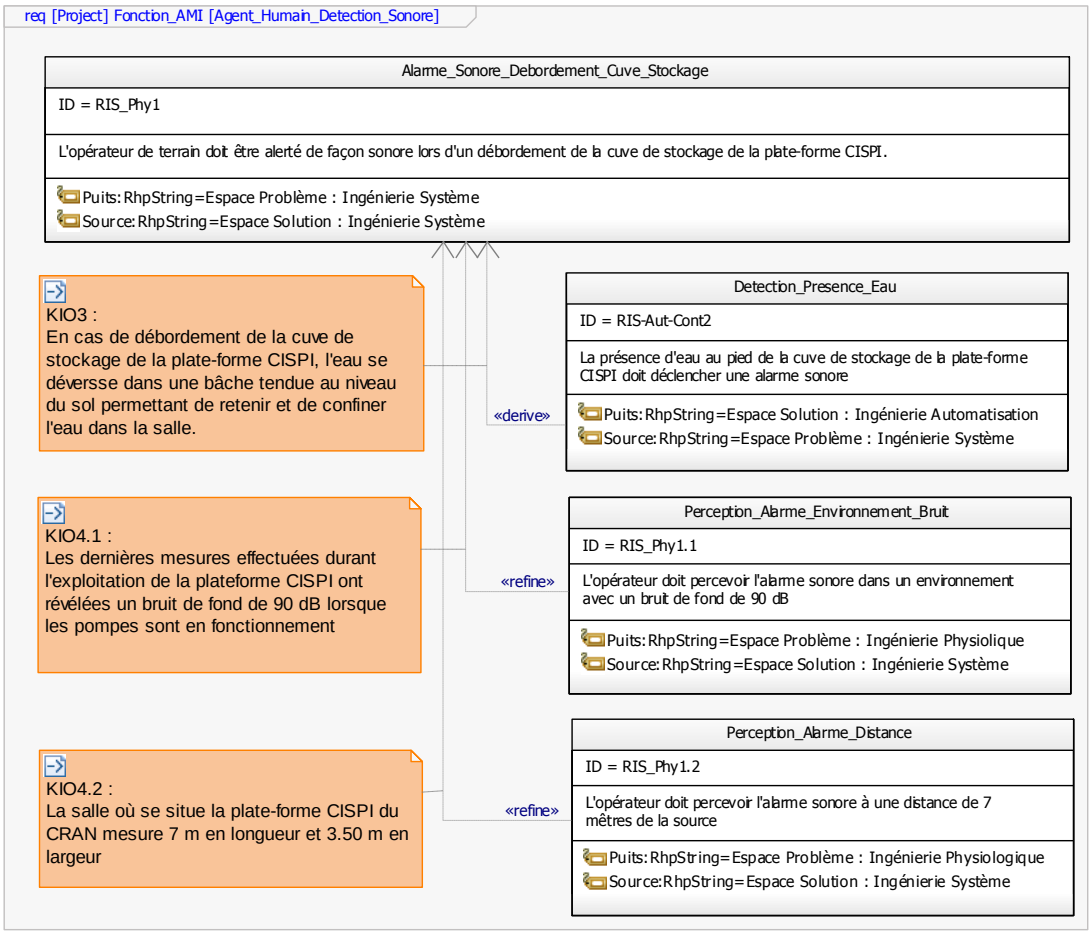


Figure 152 : Modèle d'exigences système ($R_{IS_Phy_SysML}$) relatives au constituant sensoriel de mesurage intelligent

8.2.2. Spécification prescriptive d'un modèle d'exigences constitutif d'une interaction physiologique de détection sonore

8.2.2.1. Eléments de connaissances d'une interaction physiologique

Il a été montré dans (Lieber, 2013) qu'une condition nécessaire mais non suffisante d'affordance sensorielle est aussi relative à la quantité de données physiques transportées jusqu'au capteur sensoriel de l'humain afin de stimuler l'ouverture du potentiel d'action des processus physiologiques déclenchant à leur tour les processus cognitifs de prises de décision et d'action (Cf. Annexe 8-1 - Figure 226). L'explication de la propagation de cette quantité physique parmi différentes unités structurales[66] biologiques s'appuie sur la théorie mathématique de la physiologie intégrative (MTIP[67]) définie par (Chauvet, 1993), (Chauvet, 1995). Cette théorie, constituant une partie de la connaissance (***K_{IPHY}***) de l'**espace-solution** d'ingénierie physiologique, s'appuie sur une mathématisation ayant pour objectif la construction de modèles exécutables décrivant les processus physiologiques comme une combinaison d'« interactions fonctionnelles » non symétriques et non locales entre unités structurales situés dans un espace hiérarchique (Figure 153).

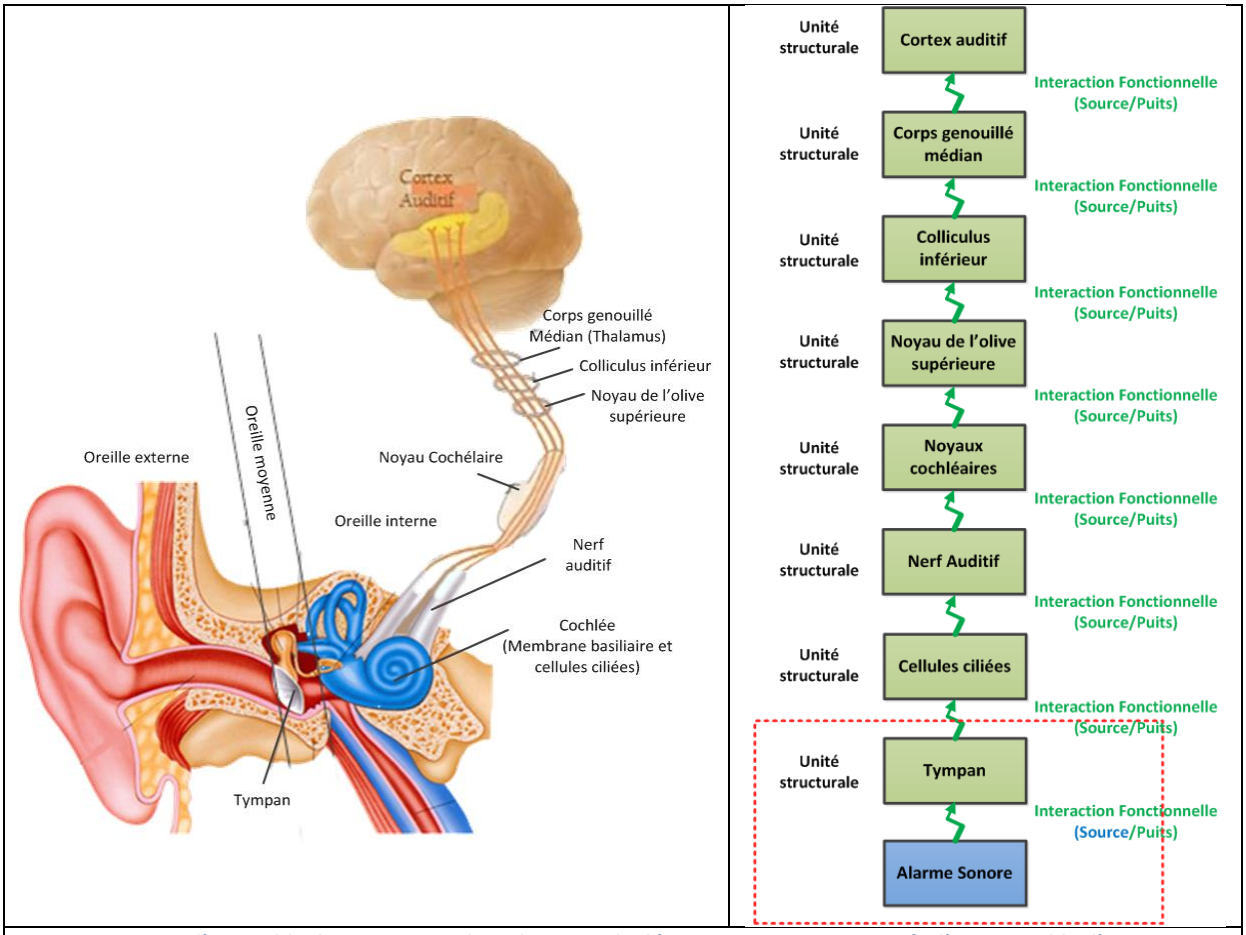


Figure 153 : Vue d'ensemble des processus physiologiques de détection sonore constitutifs d'un ensemble d'interactions fonctionnelles (Source/Puits)

[66] Unité structurale = Structure physique, agencement de molécules dans l'espace physique, par exemple un neurone localisé à un endroit du cerveau

[67] http://www.admiroutes.asso.fr/gilbertchauvet/

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage intelligents du système CISPI

Les ondes sonores qui parviennent à l'oreille font vibrer le tympan. Ces vibrations produisent dans le liquide de l'oreille interne des ondes de très faibles amplitudes qui stimulent des cellules ciliées situées à la surface de la membrane basilaire. Ces cellules ciliées constituent les récepteurs auditifs primaires (analogues au photorécepteur de l'œil). Les oscillations de la membrane basilaire provoquent l'émission de potentiels d'action par les cellules ciliées qui activent les fibres du nerf auditif innervant les noyaux cochléaires du tronc cérébral. Les fibres ascendantes gagnent le cortex auditif après avoir fait relais dans le colliculus inférieur et dans le corps genouillé médian du thalamus. D'après (Gazzaniga, Ivry, & Mangun, 2000) *et* (Bear, Paradiso, Connors, & Neuroscience, 1996). ***Nos travaux se focalisent sur l'interaction fonctionnelle qui stimule de l'extérieur le nerf auditif.***

Afin de faciliter la compréhension de la construction du modèle de détection sonore par l'**espace-solution** d'ingénierie physiologique, nous détaillons dans la suite de ce paragraphe le fondement théorique de la MTIP essentiel pour nos travaux qu'est l'« i**nteraction fonctionnelle** ». La MTIP est une approche de physiologie théorique qui envisage l'organisme vivant dans sa **globalité**[68], c'est-à-dire comme faisant partie d'un système général intégré. Sa mathématisation a pour objet de définir l'intégration des mécanismes biologiques en interaction afin de décrire le fonctionnement du système humain à partir de ces sous-systèmes, tout en se référant aux lois de la physique. En effet, pour Chauvet, l'organisme, c'est à dire l'ensemble des processus physiologiques qui prennent place entre les différentes structures biologiques, est un système hiérarchique continu et fini de structures ainsi qu'une combinaison d'interactions fonctionnelles entre ces structures (Cf. Annexe 8-1 - Figure 227).

Interaction Fonctionnelle

L'interaction fonctionnelle est considérée comme l'atome élémentaire d'un processus physiologique. Elle se définit comme « l'action d'une structure biologique sur une autre» par l'intermédiaire d'un support physique (pouvant être des ions, des molécules, des photons,…) permettant de diffuser ainsi des données d'une unité structurale source[69] vers une unité structurale puits[70] (Figure 154). Cette interaction présente plusieurs propriétés spécifiques :

- Non-symétrie : l'interaction fonctionnelle s'exerce d'une unité de structure source vers une autre unité de structure puits. Elle représente une action unidirectionnelle ; ainsi à un même niveau d'organisation, le signal ne rétroagira pas du puits vers la source. Dans notre cas d'étude, la source est l'alarme technique sonore
- Causalité événementielle : la relation de cause à effet est due à l'existence d'un événement ; c'est parce que cet événement a existé comme cause à un instant antérieur (***t***) que l'effet va exister à un instant (***t'***). C'est un effet non local, ou encore à distance. Dans notre cas, la détection sonore de l'alarme est la cause ou l'événement qui déclenche l'interaction fonctionnelle entre le cortex auditif et le cortex cognitif
- Non-instantanéité : la vitesse de transport d'une fonction élémentaire est finie. Dans notre cas, la stimulation de l'élément de connaissance par la détection sonore de l'alarme est de l'ordre de quelques millisecondes voire secondes.
- Non-localité : Une fonction élémentaire agit à distance et crée des couplages entre structures éloignées. Le produit échangé est transporté d'un endroit à un autre non voisin en se propageant au

[68] D'où notre intérêt pour construire un « Whole Model »
[69] Source : Unité structurale cause de l'interaction fonctionnelle
[70] Puits : Unité(s) structurale(s) qui reçoivent l'effet (ion ou molécules) émis par la source

travers de discontinuités structurales[71]. Dans notre cas, nous limitons cette propagation à un flux uniquement de nature physique.

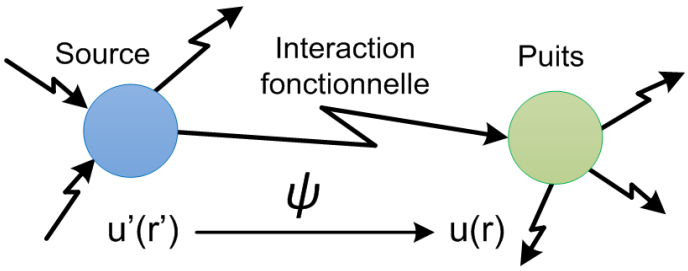


Figure 154 : Interaction Fonctionnelle ψ entre la source u'(r'), unité de structure à la position r', et le puits u(r), unité de structure à la position r. Puits et source peuvent devenir tour à tour source ou puits (Chauvet, 1995).

En ce sens, une interaction fonctionnelle traduit un mécanisme de passage d'un produit entre au moins deux unités structurales, par exemple le cortex auditif et le cortex cognitif. Ce mécanisme de passage est dépendant à la fois du temps et de l'espace. Il peut être représenté mathématiquement dans le cadre théorique de la MTIP par des opérateurs de champ non locaux appelés : S-Propagateur et notés $\boldsymbol{\Phi}$ (Chauvet, 2002). De façon didactique, (Dupont, Lieber, Morel, Méry, & Bouffaron, 2012) proposent un diagramme de type « Thyristor » pour expliquer le comportement de l'interaction fonctionnelle, en combinant les travaux de la « physiologie intégrative » avec ceux de la « physiologie de la perception et de l'action (Berthoz, 2009)» afin de mettre en évidence l'importance d'une « bonne spécification » du flux physique « **Stimulus-source** » ($\psi^{(i)}(r^{(i)\prime})$).

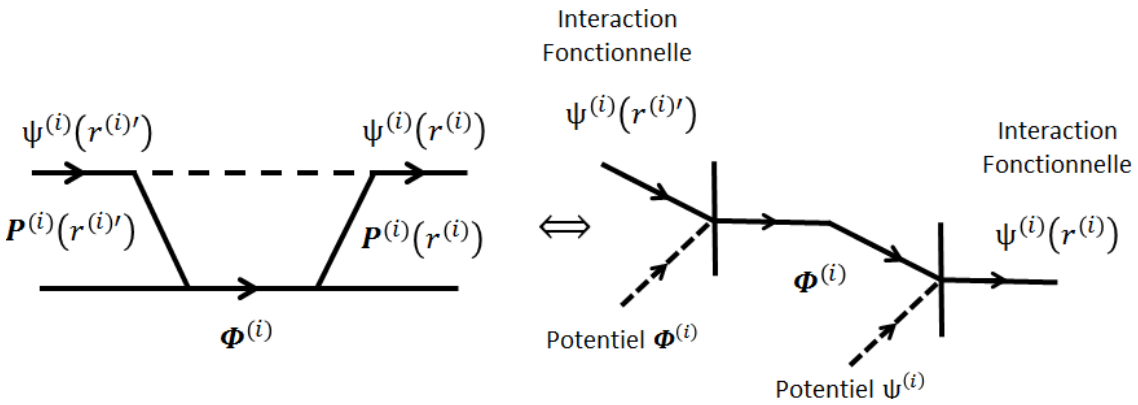


Figure 155 : Modèle explicatif de l'interaction fonctionnelle : Analogie entre la représentation de Chauvet et un schéma électrique montrant l'importance de la stimulation d'un certain potentiel d'action pour ouvrir les « portes » de propagation du flux d'interaction physico-physiologique (Lieber, et al., 2013)

De façon plus détaillée, une interaction fonctionnelle ψ située en r' stimule une interaction fonctionnelle $\boldsymbol{\Phi}$ d'un niveau inférieur (premier thyristor) qui elle-même va stimuler une interaction fonctionnelle située en r du

[71] Unité structurale qui est le siège d'un processus fonctionnel différent de celui existant dans les unités structurales au niveau supérieur. Les discontinuités structurales sont à l'origine de l'existence de l'organisation hiérarchique

même niveau supérieur, permettant ainsi d'assurer de r' à r le passage de ψ avec un certain délai dépendant de sa vitesse de propagation (second thyristor). Appliqué à notre exemple, ceci met en exergue que la propagation du produit physiologique jusqu'au cortex cognitif est contrainte par certaines conditions nécessaires telle la stimulation initiale liée au son de l'alarme (en amont du cortex auditif et de l'oreille). Dit autrement, il faut qu'une certaine quantité physique liée à la détection du son de l'alarme se propage pour stimuler une connaissance stockée chez l'humain, telle que celle matérialisée par les procédures de conduites en situation incidentelle ou accidentelle, permettant de rétablir l'installation dans un état sûr de fonctionnement.

Notons cependant que la construction de ces modèles exécutables dans le formalisme de la MTIP requiert un grand nombre de données relevant à la fois de l'anatomie (densité cellulaire par exemple) et de la physiologie (mécanismes biochimiques par exemple) qui, à l'heure actuelle ne sont pas disponibles faute d'expérimentations techniquement difficiles. Ces précédents travaux ont cependant permis de montrer que le facteur d'échelle (densité synaptique) requis pour la mise en œuvre de la MTIP en modélisation numérique n'était pas forcément nécessaire en regard de celui plus macroscopique exigé par la nature collaborative d'un processus d'ingénierie système. En ce sens, nous interprétons l'interaction physico-physiologique de détection sonore en tant qu'interaction physique, signifiant qu'un flux physique est propagé d'un constituant technique source vers un constituant physiologique puits tant que le milieu traversé ne nécessite pas une transmutation en flux biologique.

8.2.2.2. Définition des exigences de l'interaction physiologique ciblée

Lors de la modélisation de l'interaction de détection sonore, il est apparu opportun aux domaines de l'ingénierie système et de l'ingénierie physiologique de traiter la quantité physique transportée en puissance pour viser une ingénierie centrée système facilitant la prise en compte d'éléments d'ingénierie centrée humain par l'ingénierie courante centrée technique. En effet, les conditions physico-physiologiques de perception sensorielle exprimées en puissance assurent une meilleure intégration entre le monde de la physique (Cf. chapitre 5) et le monde de la physiologie. Cela amène le domaine d'ingénierie physiologique à spécifier les sons[72] complexes (décomposables en une superposition de sons purs sinusoïdaux (bruit aléatoire)) à travers le concept d'onde de pression proportionnelle à la puissance acoustique. En ce sens, il est important de noter que la requête d'un modèle physiologique exécutable, qui soit constitutif de l'ensemble des modèles d'un seul et même système, a inféré une base différente de modélisation que celle qu'aurait mise en œuvre le spécialiste de façon isolée.

A cette étape du scénario de co-spécification, l'**espace-solution** d'ingénierie physiologique doit appliquer le patron architectural d'interaction fonctionnelle (Figure 154) pour étudier l'interaction de détection sonore selon trois constituants correspondant au constituant physiologique « puits », l'interaction fonctionnelle « source/puits » et le constituant technique « source » (Figure 165). Il convient alors de s'intéresser dans un premier temps au constituant « puits » pour définir la plage de perception auditive chez l'humain et délimiter ainsi dans un second temps les caractéristiques physiques du constituant technique « source » afin de garantir la « bonne » détection de l'alarme sonore.

[72] Un son est une perturbation locale de pression qui se propage de proche en proche.

Constituant physiologique de détection sonore

Ce constituant « puits » délimite l'espace de bonne perception auditive en fonction de la fréquence du son appartenant à une fenêtre spectrale comprise entre 16 Hz (infra-sons) et 20 kHz (ultra-sons) et son intensité sonore (I_{sonore}) exprimée en décibel (dB) (Goldstein, 2009).

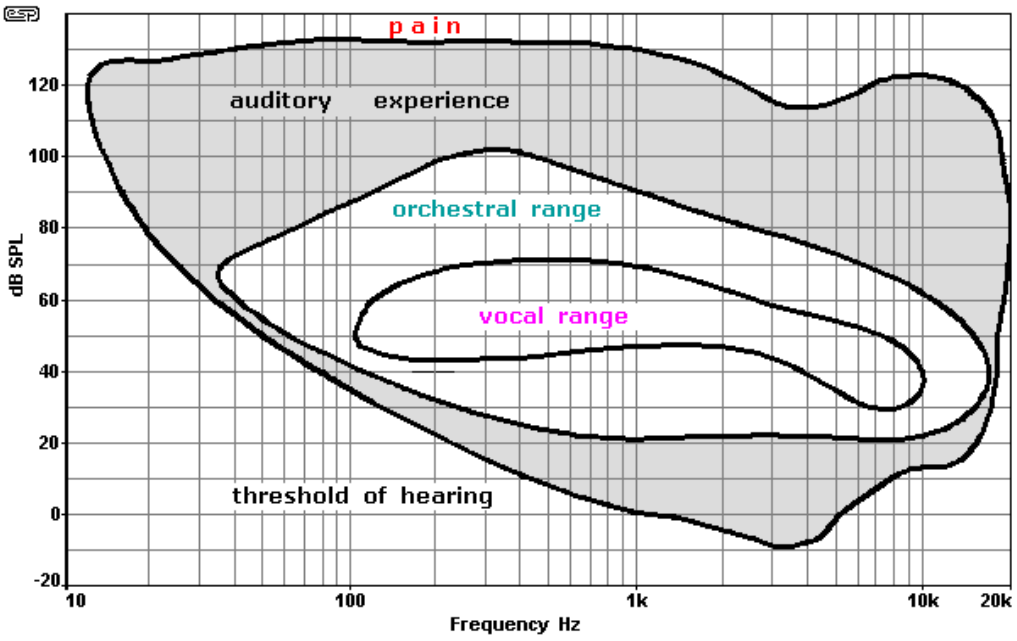


Figure 156 : Plages typiques du champ auditif humain[73]

A partir de cette connaissance, l'**espace-solution** d'ingénierie physiologique dérive une nouvelle exigence mesurable ($I_{Phy\text{-}H1}$) introduisant l'intensité sonore (I_{sonore}) du signal exprimée en décibel selon les inégalités suivantes (Figure 157) :

«Requirement» Capacité_Physiologique_Humain
ID = RIPhy-1.1
L'onde de pression reçue au niveau du canal auditif (Isonore) doit satisfaire par son intensité sonore résultante le champ auditif humain (courbe d'audibilité humaine) : HearingRangesMin(w) < Isonore < HearingRangesMax(w)
Puits:RhpString=Espace Problème : Ingénierie Système Source:RhpString=Espace Solution : Ingénierie Physiologique

Figure 157 : Exigence physiologique ($R_{IPhy\text{-}1.1}$) : plage de perception sonore chez l'humain

[73] http://sound.westhost.com/articles/fadb.htm

Dans ce cas, HearingRangesMin(w) et HearingRangesMax(w) sont deux fonctions mathématiques retournant respectivement les valeurs minimale et maximale de l'intensité sonore définies par le champ auditif de l'humain en fonction de la fréquence (w) du signal sonore. De plus, à partir de cette exigence (***$I_{Phy-1.1}$***), **l'espace-solution** d'ingénierie en physiologie dérive une nouvelle exigence (***$I_{Phy-1.2}$***) (Figure 158) spécifiant selon la fréquence du signal (w), la durée minimale du signal sonore à émettre afin de garantir la « bonne » perception par un humain « puits ».

«Requirement» Duree_Alarme_Sonore
ID = RIPhy-1.2
La durée (Delta_T) du son entendu par l'opérateur humain doit être au moins égale à l'inverse de la fréquence (w) du son (considéré comme pur) : DeltaT > 1/w
Puits:RhpString=Espace Problème : Ingénierie Système Source:RhpString=Espace Solution : Ingénierie Physiologique

Figure 158 : Exigence ($R_{IPhy-1.2}$) : Durée minimale alarme sonore

Enfin, afin de faciliter la « bonne » détection auditive de l'alarme sonore, l'ingénieur en physiologie recommande en plus que le sous-système binaural (les deux oreilles) de l'opérateur soit aligné avec la source sonore, introduisant ainsi une nouvelle exigence ($R_{IPhy1.3}$) de nature anthropométrique (Figure 226).

«Requirement» Alignement_Antropometrique_Alarme
ID = RIPhy1.3
Les axes anthropométriques de l'opérateur doivent être alignés pour satisfaire le système binaural pour assurer la localisation de la source sonore
Puits:RhpString=Espace Problème : Ingénierie Système Source:RhpString=Espace Solution : Ingénierie Physiologique

Figure 159 : Exigence physiologique ($R_{IPhy-1.3}$) : Alignement anthropométrique entre les constituants technique « source » et humain « puits »

Interaction fonctionnelle de propagation sonore

Ce constituant « Source/Puits » contextualise l'interaction de détection sonore d'un son pur *(et plus généralement d'un signal sonore quelconque)* en définissant mathématiquement la dépendance de l'intensité sonore à la pression sonore selon :

$$I_{sonore} = 10Log_{10}\left((P_{rms}/P_0)\right)^2 = 20Log_{10}(P_{rms}/P_0) \tag{27}$$

$$P_{rms} = \left(\frac{1}{\Delta T}\int_{t}^{t+\Delta T}\left(P(r,t)\right)^2 dt\right)^{1/2} \tag{28}$$

$$P(r,t) = \left(\frac{\rho_0 c\, E}{4\pi r^2}\right)^{1/2} sin(\omega t) \tag{29}$$

Avec :

- P_{rms} = pression efficace
- P = pression sonore dépendant des coordonnées d'espace (r) et de temps (t) (Bjørner, 2008)
- E = puissance sonore de la source sonore
- c = vitesse du son
- ρ0 = masse volumique de l'air
- r = distance source-puits

Notons que les paramètres « c » et ρ0 dépendent de la température, de l'humidité et de la pression atmosphérique évaluées dans le milieu de la propagation du son. De plus, l'équation (29) montre que la puissance sonore diminue en fonction de la distance avec la source. En ce sens, l'interaction sonore apparait fortement contextualisée en dépendant de paramètres liés à l'environnement de propagation du son. Ceci est d'autant plus vrai dans un environnement industriel (*tel que celui de la conduite d'une centrale de production électrique*) où plusieurs sources sonores peuvent se superposer. En se référant aux standards en vigueur tels que (NF S32 001, 1975), l'**espace-solution** d'ingénierie physiologique dérive une nouvelle exigence (***R_{IPhy2}***) spécifiant que l'intensité d'une alarme sonore (***I_{Sonore}***) doit être supérieure de 10 dB au niveau sonore ambiant (***I_{Bruit}***).

«Requirement» Bruits_fonds
ID = RIPhy2
L'intensité sonore Isonore doit être 10dB supérieure au niveau sonore ambiant : Isonore - Ibruit > 10 dB
Puits:RhpString=Espace Problème : Ingénierie Système Source:RhpString=Espace Solution : Ingénierie Physiologique

Figure 160 : Exigence physiologique (R_{IPhy-2}) : Intensité signal sonore – bruits de fonds

Constituant technique d'émission sonore

En tenant compte des deux spécifications précédentes, ce constituant « source » doit satisfaire, entre autres, à l'exigence (***$R_{IPhy1.1}$***), relative à la quantité d'onde de pression requise pour détecter un son, l'**espace-solution**

d'ingénierie physiologique dérive une nouvelle exigence (***R_{IPhy3}***) contraignant l'objet technique « source » au regard des contraintes de l'environnement opérationnel selon la loi définie par (Müller & Möser, 2012).

«Requirement» Fréquence_Alarme_Sonore
ID = RIPhy3
La source sonore doit générer une onde sonore de fréquence (w) comprise entre 16Hz et 20kHz avec une puissance « sonore » dépendante de la fréquence suffisante.
Puits:RhpString=Espace Problème : Ingénierie Système Source:RhpString=Espace Solution : Ingénierie Physiologique

Figure 161 : Exigence technique (R_{IPhy-2}) : Intensité signal sonore – bruits de fonds

L'ensemble de ces exigences d'ingénierie physiologique définissent un premier modèle de l'interaction physico-physiologique de détection sonore d'une alarme et peuvent ainsi être prescrites à l'**espace-problème** d'ingénierie système sous forme d'un diagramme pour validation (Cf. Annexe 8.2 - Figure 228). Néanmoins ce premier niveau descriptif de modélisation n'est pas suffisant pour notre heuristique de co-spécification système qui requiert de prescrire un modèle constitutif d'une exécution d'ensemble avec d'autres modèles.

8.2.3. Spécification prescriptive d'un modèle exécutable constitutif d'une interaction physico-physiologique de détection sonore

8.2.3.1. Eléments de modélisation exécutable d'une interaction physico-physiologique

Afin de construire un modèle exécutable de l'interaction de détection sonore, l'**espace-solution** d'ingénierie physiologique a dû sélectionner le langage et l'outil de modélisation le plus adapté à notre contrainte de co-simulation système. Dans un premier temps, nous avons évalué l'outil PhysioMatica™ (Bennani, Chauvet, Chauvet, Dupont, & Jouen, 2009) pour lequel notre expert en physiologie a contribué au développement au sein de VFS-Bio™. Cet outil, basé sur le cadre de modélisation de la MTIP avec pour objectif la modélisation et la simulation des fonctions physiologiques intégrées du vivant, n'a pas eu le degré de maturité technique suffisant pour l'environnement visé par nos travaux. Dans un second temps, nous nous sommes intéressés aux bibliothèques Modelica dédiées à la modélisation en physiologie (Mateják, et al., 2014) . Cependant, il s'est avéré que le niveau de représentation retenu pour ces librairies était à un niveau logique/organique ce qui ne couvrait pas notre besoin de représentation à un niveau d'abstraction fonctionnel. Ces différents constats ont motivé le choix de l'**espace-solution** d'ingénierie physiologique de sélectionner l'outil Simulink®, afin de lui permettre la mise en équation sous forme de blocs constitutifs de l'interaction physico-physiologique de détection sonore, tout en offrant les capacités de se coupler au bus de co-simulation: *Cosimate*. En ce sens, nous présentons dans la suite de ce paragraphe, la formalisation des blocs technique « source » et physiologique « puits » ainsi que de l'interaction fonctionnelle, sous la forme de diagramme fonctionnel Simulink®.

Bloc physiologique de détection sonore

Ce **bloc** physiologique **« puits »** est caractérisé par sa position (***X_Puits_Value, Y_Puits_Value***) ainsi que par une sous-fonction : « ***fncPerception*** » (Cf. Annexe 8-3 – Fonction Perception) dont l'objectif est de calculer la bonne détection (ou non) de l'alarme sonore (booléen *Perception*) tenant compte de la fréquence (***Frequence_Sonore***) et de la puissance (***Puissance_Physiologique***) de la source sonore, mais également du bruit ambiant (***Bruits_Fonds***) (Figure 162a).

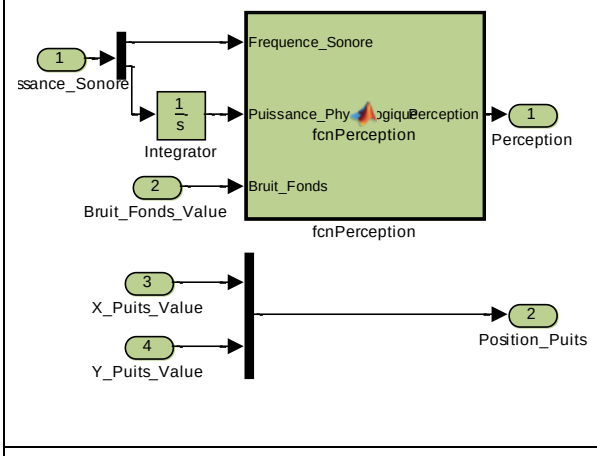


Figure 162a : Structure interne du bloc « puits »

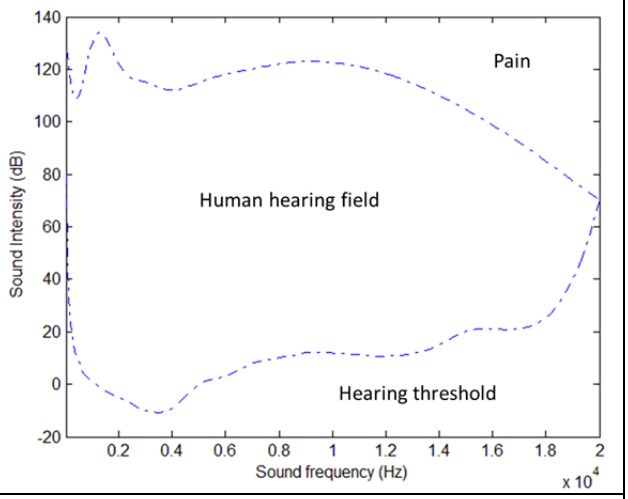


Figure 162b : Représentation de la formalisation dans Simulink du champ auditif de l'humain

Figure 162 : Bloc Simulink – constituant physiologique « puits » de détection sonore

De façon plus détaillée, cette fonction (***fncPerception***) concrétise l'exigence physiologique ($R_{IPhy1.1}$) en comparant le résultat du calcul de l'intensité sonore reçu au niveau du canal auditif avec la limite haute (***HearingRangeMax(w)***) et la limite basse (***HearingRangeMin(w)***) du champ auditif humain.

Bloc d'interaction de propagation sonore

Ce bloc d'interaction fonctionnelle « Source/Puits » (Figure 163) formalise la propagation de l'onde sonore à travers le milieu contextualisé de la conduite des centrales de production. En ce sens, il prend en compte la vitesse (ou célérité (***c***)) du son variant en fonction de la température et de la masse volumique du milieu de propagation (***rho***), ainsi que la distance entre le constituant technique source (***X_Source, Y_Source***) et humain puits (***X_Puits, Y_Puits***) pour évaluer l'amplitude du signal source au niveau du puits (Cf. Annexe 8-3 – Fonction Amplitude).

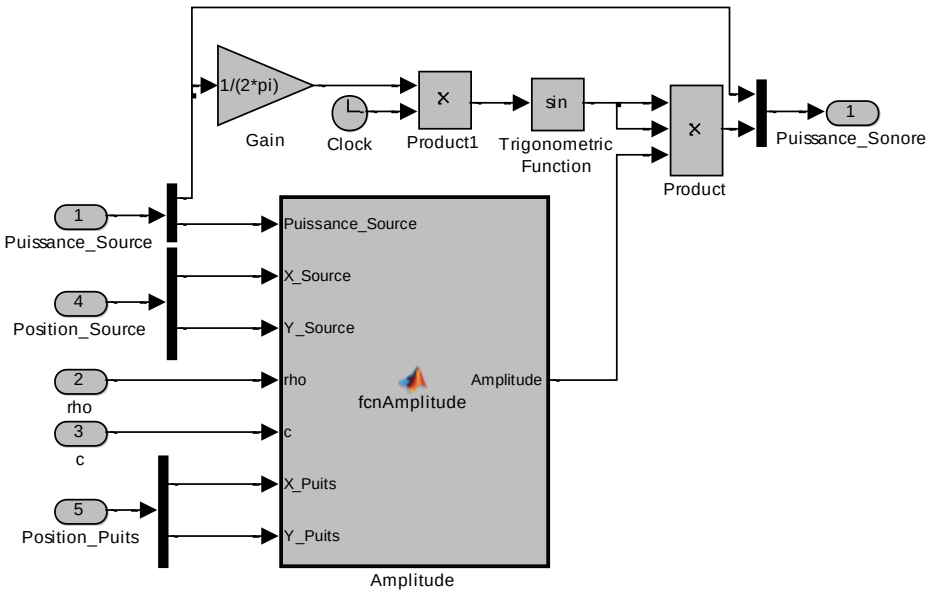


Figure 163 : Bloc Simulink – Interaction fonctionnelle de propagation sonore

Bloc technique d'émission sonore

Ce bloc technique « source » d'émission sonore est caractérisé par sa position (***X_Source, Y_Source***). Quant au son émis par la source technique, celui-ci est caractérisé par sa fréquence (***Frequence_Source_Value***) et sa puissance (***Puissance_Value***). Le block « ***switch*** » permet d'activer ou de désactiver l'émission du son en fonction de la valeur de la variable « ***reqAlarme*** » (Figure 164).

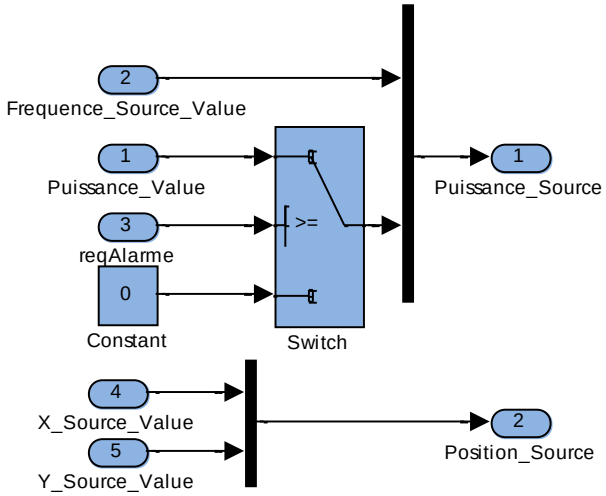


Figure 164 : Bloc Simulink – bloc technique « source » d'émission sonore

Une fois l'ensemble des blocs constituant l'interaction de détection sonore spécifiés dans l'outil Simulink, l'espace-solution d'ingénierie physiologique les architecture les uns par rapports aux autres selon le patron architectural d'une interaction fonctionnelle (Figure 154).

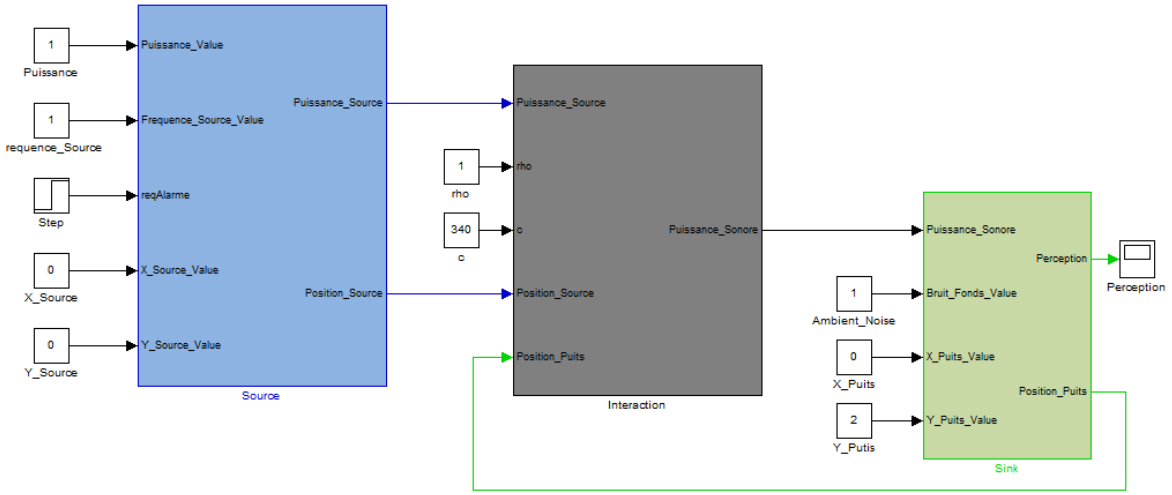


Figure 165 : Diagramme Simulink de l'interaction de détection sonore entre un constituant technique « source » et un constituant physiologique « puits ».

Vérification par exécution du modèle d'interaction

L'ingénieur en physiologie est ainsi en mesure de vérifier la cohérence intrinsèque de son modèle d'interaction de détection sonore en simulant le modèle avec l'outil Simulink selon différents scénarios de tests. En ce sens, nous présentons dans la suite de ce paragraphe quelques scénarios que nous illustrons à travers les traces d'exécution du modèle.

Le premier scénario de test consiste à vérifier que la puissance sonore reçue par le bloc humain « puits » décroit en fonction de la distance avec le bloc technique « source ». Cette vérification montre que, pour une puissance et une fréquence d'émission constante, la puissance sonore reçue par le bloc « puits » décroit en même temps que la distance entre les deux blocs augmente (Figure 166). Notons également qu'à partir d'un certain seuil (Puissance Sonore < 11 Watts) l'alarme sonore n'est plus détectée par le bloc humain « puits ».

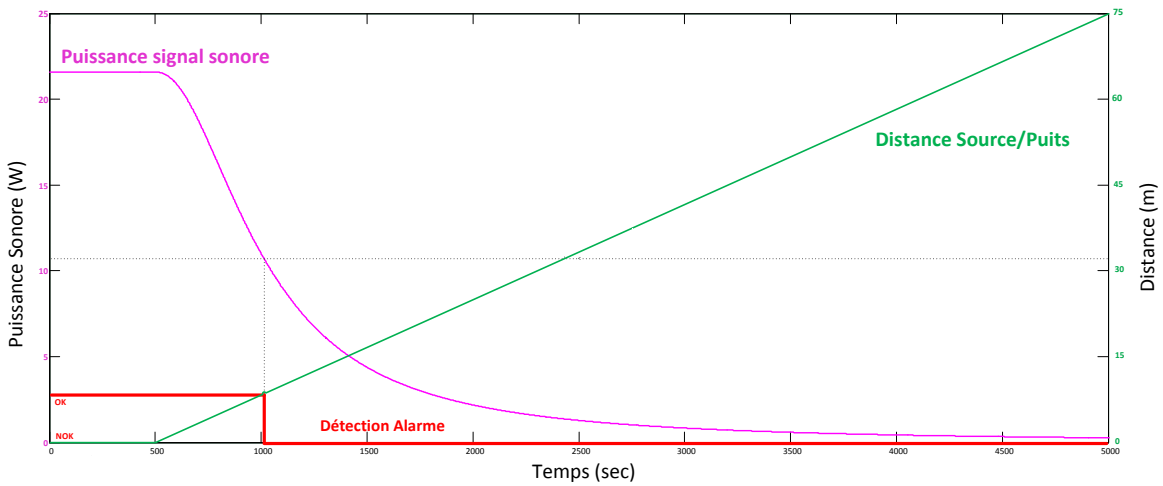


Figure 166 : Puissance sonore (W) en fonction de la distance « Source/Puits »

Le second scénario de test consiste à vérifier que la détection du signal sonore par un bloc humain est conditionnée par le champ auditif humain. En observant la trace d'exécution ci-dessous (Figure 167), nous remarquons que pour une fréquence du signal sonore supérieure à 20 KHz la détection de l'alarme n'est plus assurée. Cela correspond aux résultats attendus étant donné que le signal n'est plus compris dans le champ auditif humain.

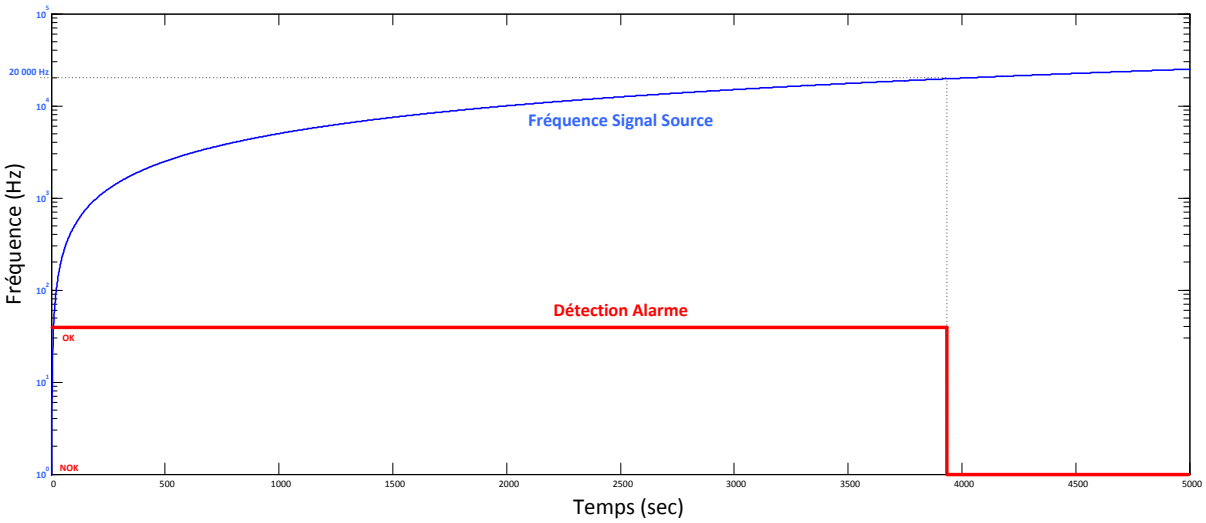


Figure 167 : Détection de l'alarme sonore en fonction de la fréquence du signal

Le dernier scénario de test que nous présentons consiste à vérifier la prise en compte du bruit de fond dans le modèle de l'interaction de détection sonore. Nous observons dans la trace d'exécution (Figure 168) que pour un bruit de fond supérieure à 10 dB au bruit émis par la source sonore, la détection de l'alarme par un bloc humain n'est plus assurée. En ce sens, la propriété d'atténuation du bruit par rapport au bruit ambiant est bien prise en compte au niveau du modèle constitutif de l'interaction.

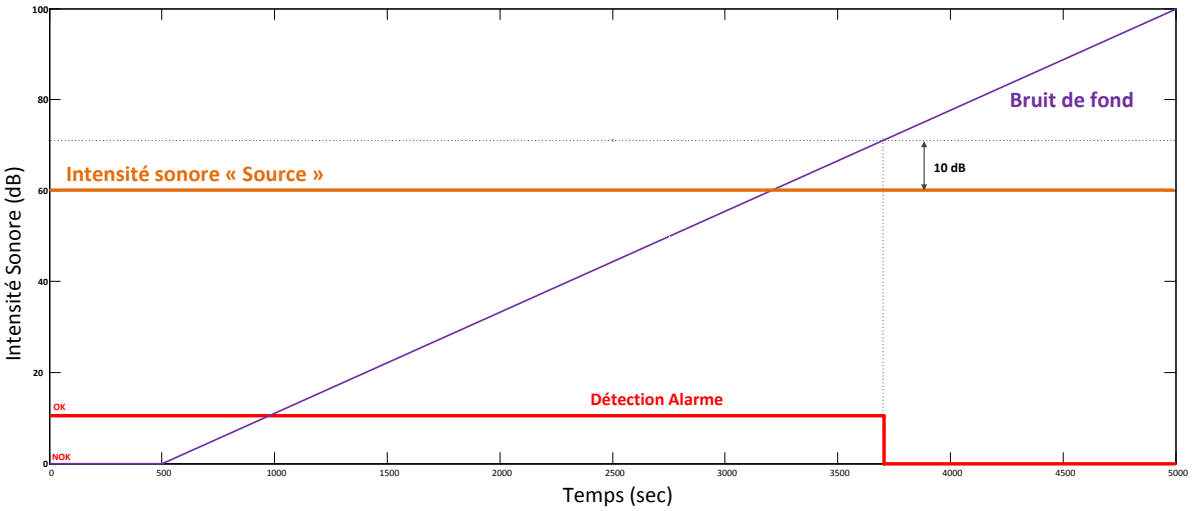


Figure 168 : Détection de l'alarme en fonction du bruit de fond

Ainsi en déroulant un ensemble de scénarios de test, l'**espace-solution** d'ingénierie physiologique est en mesure de vérifier la construction de son modèle (M_{hIM}) à posteriori. Une fois la spécification de l'interaction

de détection sonore (S_{IPhy}) vérifiée, le domaine d'ingénierie physiologique doit prescrire en langage commun SysML sa solution au domaine d'ingénierie système pour validation.

8.2.3.2. Définition d'un modèle exécutable de l'interaction physico-physiologique ciblée

A cette étape du scénario de co-spécification système, l'espace-solution d'ingénierie physiologique doit prescrire à l'espace-problème d'ingénierie système au minimum les interfaces du bloc d'interaction de détection sonore afin que ce dernier puisse l'architecturer avec l'ensemble des modèles spécialistes constitutifs du modèle global afin de valider en co-simulation son comportement global en regard du système visé. En ce sens, l'ingénieur en physiologie doit transposer l'essentiel des informations d'interopération de son modèle Simulink en langage SysML afin de constituer un référentiel partagé de modèles avec l'ensemble des ingénieries spécialistes. En ce sens, l'ingénieur en physiologie applique la même démarche que l'ingénieur d'automatisation (dans le chapitre 6) qui transpose sa spécification du constituant de contrôle en langage Simulink vers le langage SysML en se référant à une table de correspondance entre les éléments de modélisation des deux langages (Chapitre 6) .

Ainsi, l'ingénieur en physiologie restreint la transposition de son modèle (M_{hIM}) d'un point de vue structurel en représentant l'architecture externe de l'interaction de détection sonore dans un diagramme de définition de blocs SysML, ainsi que sa structure interne dans un diagramme interne de blocs SysML. Notons que d'un point de vue comportemental, les équations traduisant la propagation du son d'un bloc technique « source » vers un bloc physiologique « puits » auraient pu être formalisées dans des diagrammes paramétriques SysML (Cf. Chapitre 5). Dans le diagramme de définition de blocs, la structure hiérarchique de l'interaction de détection sonore (« blocks » SysML : « *Interaction_Detection_Sonore* ») est représentée par une relation de composition SysML de l'ensemble des « blocks » SysML constitutifs : « *Emission_Sonore, Interaction_Fonctionnelle, Detection_Sonore* » (Figure 169).

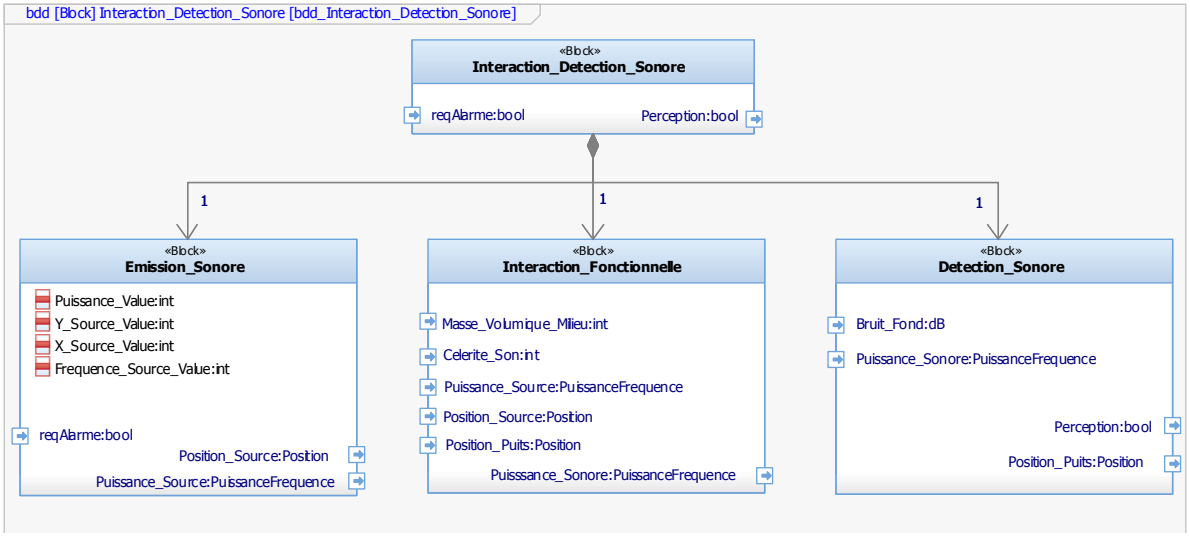


Figure 169 : Diagramme de définition de blocs (bdd) - Architecture externe de l'interaction de détection sonore

La structure interne de l'interaction de détection sonore est quant à elle modélisée dans un diagramme interne de blocs SysML représentant les flux échangés entre les différents constituants de l'interaction (« Part » :

Bloc_Technique_Source, Bloc Source_Puits, Bloc_Humain_Puits) au travers de « connecteurs » SysML reliant des « FlowPorts » indiquant la direction ainsi que le type de flux (Figure 140).

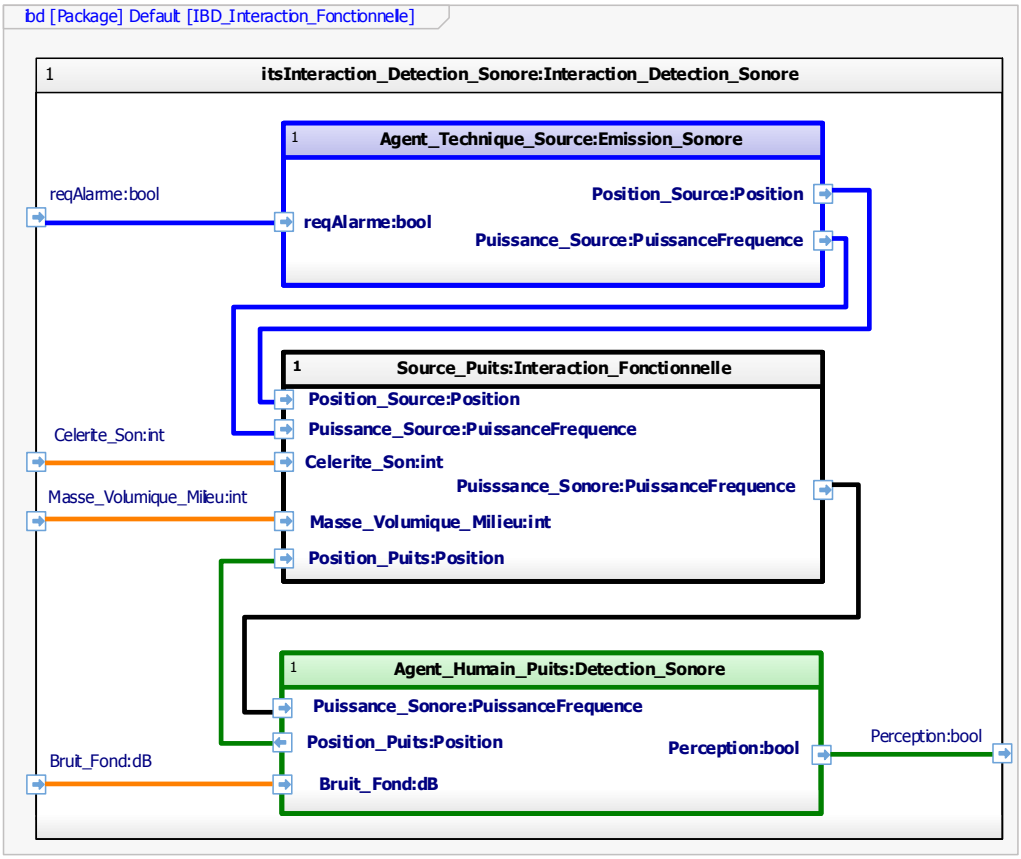


Figure 170 : Diagramme interne de bloc (ibd) - Structure interne de l'interaction de détection sonore

Cela permet à l'**espace-solution** d'ingénierie physiologique de conserver la connaissance (K_{IPhy}) de son modèle (M_{hIM}) à l'intérieur de son domaine d'expertise.

D'un point de vue comportemental, le modèle « d'interaction de détection sonore » spécifié selon un ensemble d'équations est validé par co-exécution de modèles en interaction avec le modèle opérationnel de conduite, ainsi que les modèles constitutifs des parties physique et logique de la plate-forme CISPI.

8.3. Co-intégration système d'un constituant d'interaction physico-physiologique de détection sonore

La validation système d'un élément, modèle ou composant, constitutif d'un modèle global, requiert son intégration dans une situation réelle conformément à la définition formelle d'une spécification ou la reflétant au mieux en pratique par niveaux de maturité technologiques[74].

[74] Ces travaux se situent aux niveaux 4 et 5 d'une échelle TRL

8.3.1. Intégration d'un modèle constitutif de l'interaction physico-physiologique ciblée

Au cours de cette étape du scénario de co-spécification système, l'**espace-problème** d'ingénierie système valide la spécification (S_{IPhy_SysML}) de l'**espace-solution** d'ingénierie physiologique. Cette activité est réalisée par co-exécution de modèles afin de s'assurer que le modèle de l'interaction de détection sonore interopère avec l'ensemble des modèles constitutifs du **modèle global** (*modèle opérationnel, modèle des constituants physiques, de contrôle et de commande*) afin de satisfaire l'ensemble des exigences système ($R_{IS_Phy_SysML}$).

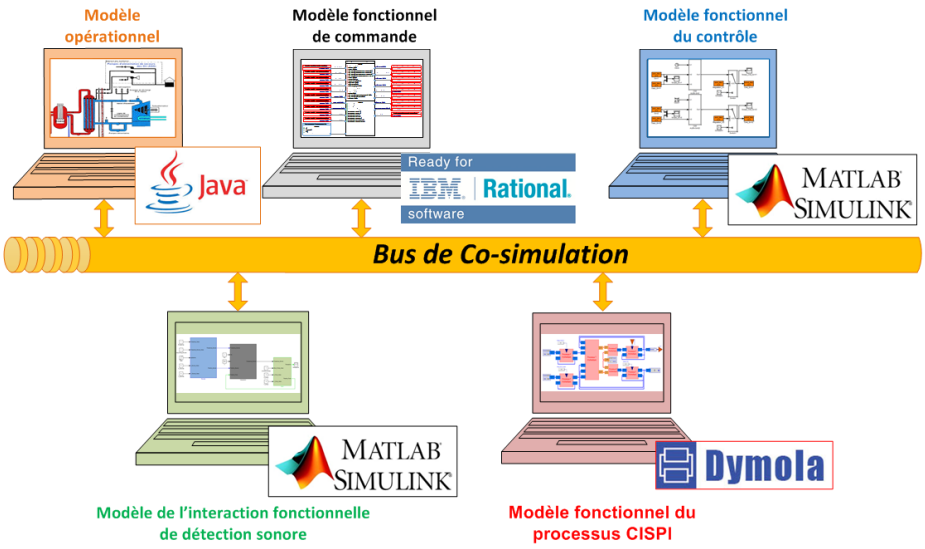


Figure 171 : Environnement de co-simulation système pour la validation de la spécification (S_{IPhy}) d'une interaction de détection sonore

En ce sens, l'espace-problème d'ingénierie système architecture l'ensemble des modèles des ingénieries spécialistes constitutifs de la plate-forme CISPI («Parts » SysML : ***itsInteraction_Detection_Sonore_1***, ***itsSous_Systeme_Logique***, ***itsSous_Systeme_Physique***). Cette spécification exécutable, sous forme d'un diagramme interne de blocs SysML, représente les liens et flux échangés entre les différents constituants de la plate-forme ainsi qu'avec son environnement (« Part » SysML : ***Circuit_Primaire***) (Figure 172) (Cf. Annexe 8-4). Notons que le modèle constitutif de l'interaction de détection sonore nous amène à raffiner le modèle global de la plate-forme CISPI afin de prendre en compte les nouvelles interfaces. En ce sens, nous raffinons :

- Le modèle constitutif du contexte opérationnel afin de prendre en compte les paramètres liés aux bruits de fond (***Bruits_Fond***) de la salle où se situe la plate-forme CISPI du CRAN, ainsi que la masse volumique de l'air (***Masse_Volumique_Milieu***) contenu dans cette salle et par conséquent la vitesse de propagation du son (***Celerite_Son***) dans cet environnement.
- Le modèle constitutif de conduite (contrôle) afin d'élaborer l'ordre (***reqAlarme***) de mise en service (hors service) de l'alarme technique sonore. Cet ordre est calculé par une fonction de contrôle prenant en compte la présence d'eau au niveau du constituant physique
- Le modèle constitutif du procédé afin de renvoyer au constituant de contrôle la donnée physique de la présence d'eau (ou non) (***Presence_Eau***) au niveau du sol de la plate-forme CISPI. Le renvoie de cette donnée requière un observateur humain ou technique au niveau du procédé.

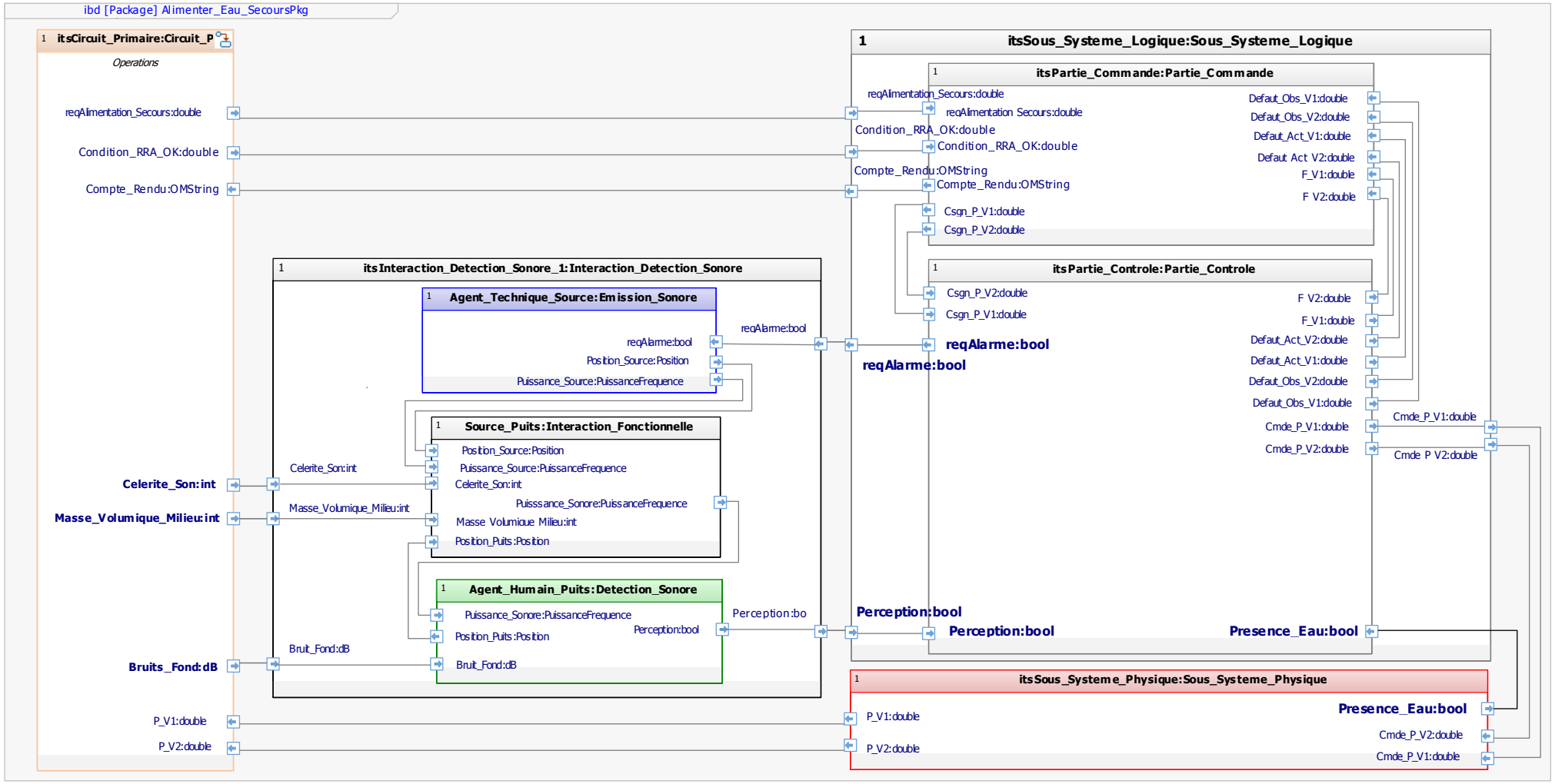


Figure 172 : Définition de l'architecture de co-simulation système (*Bloc d'interaction de détection sonore + bloc logique + bloc physique*) par l'espace-problème d'ingénierie système

Ainsi, en exécutant un ensemble de scénarios en co-simulation système, l'**espace-problème** d'ingénierie système est en mesure d'évaluer si le comportement global du système étudié satisfait l'ensemble des exigences systèmes.

Prenons l'exemple des deux scénarios suivants, dont l'objectif est de valider la puissance de l'alarme technique pour une fréquence donnée (20 Hz), afin que l'agent humain puisse « bien » détecter l'alarme sonore en situation opérationnelle de conduite.

Dans le cas du scénario 1 (Figure 173), on remarque que la fonction de contrôle détecte correctement la présence d'eau. Le temps entre l'apparition de l'événement « Presence_Eau » et l'élaboration de l'ordre « ReqAlarme » correspond à un cycle d'exécution de la fonction de contrôle. En ce sens, l'exigence système de contrôle ($\boldsymbol{R_{IS_Aut_Cont2}}$) est satisfaite. En revanche, nous constatons que dans ce contexte opérationnel, la puissance de l'alarme sonore n'est pas suffisante pour que cette dernière soit détectée « Perception » par l'agent humain situé à cinq mètres de la source. Par conséquent, les exigences systèmes ($\boldsymbol{R_{IS_Phy1.1}}$) et ($\boldsymbol{R_{IS_Phy1.2}}$) ne sont pas satisfaites.

Dans le cas du scénario 2 (Figure 174), on conserve les mêmes conditions (paramètres) que pour le scénario 1, uniquement la puissance de l'alarme technique est augmentée de 0,5 Watts. En exécutant ce scénario par co-exécution de modèles, on constate que cette fois-ci l'agent humain détecte l'alarme sonore. Les exigences systèmes ($\boldsymbol{R_{IS_Phy1.1}}$) et ($\boldsymbol{R_{IS_Phy1.2}}$) sont ainsi satisfaites, l'ingénieur système est alors en mesure de valider la configuration système associée au scénario 2.

Scénario 1
Condition initiale :

Distance Alarme Technique (Source) / Agent Humain (Puits) = 5 mètres	Masse volumique de l'air (Température = 20 °C) = 1,2 kg/m^3
Célérité du son = 331,5 m/s	Bruit de fond = 90 dB
Puissance Alarme = 0.7 W	Fréquence Alarme = 20 Hz

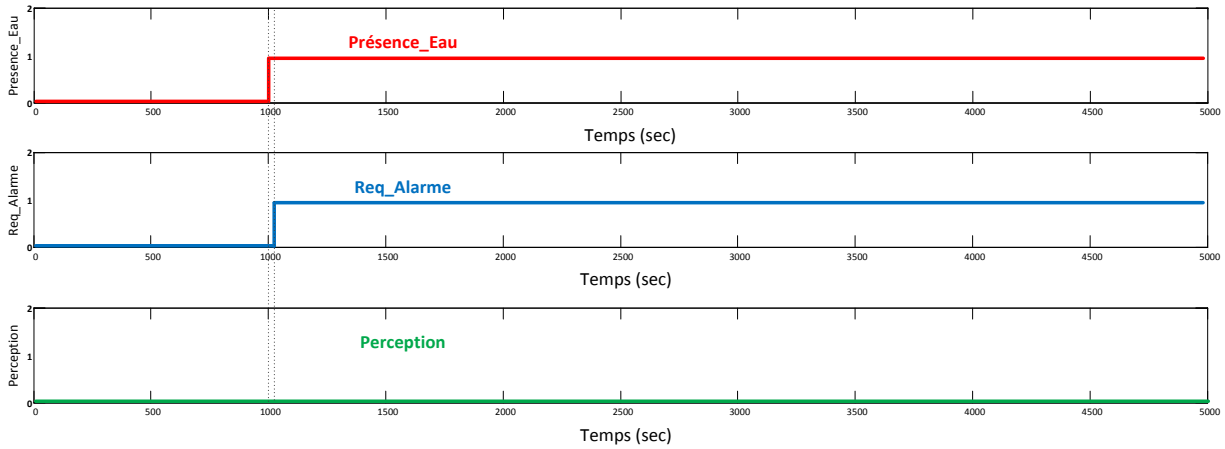


Figure 173 : Trace d'exécution du scénario 1 de validation de l'interaction de détection sonore

Scénario 2

Condition initiale :

Distance Alarme Technique (Source) / Agent Humain (Puits) = 5 mètres	Masse volumique de l'air (Température = 20 °C) = 1,2 kg/m^3
Célérité du son = 331,5 m/s	Bruit de fond = 90 dB
Puissance Alarme = 1.2 W	Fréquence Alarme = 20 Hz

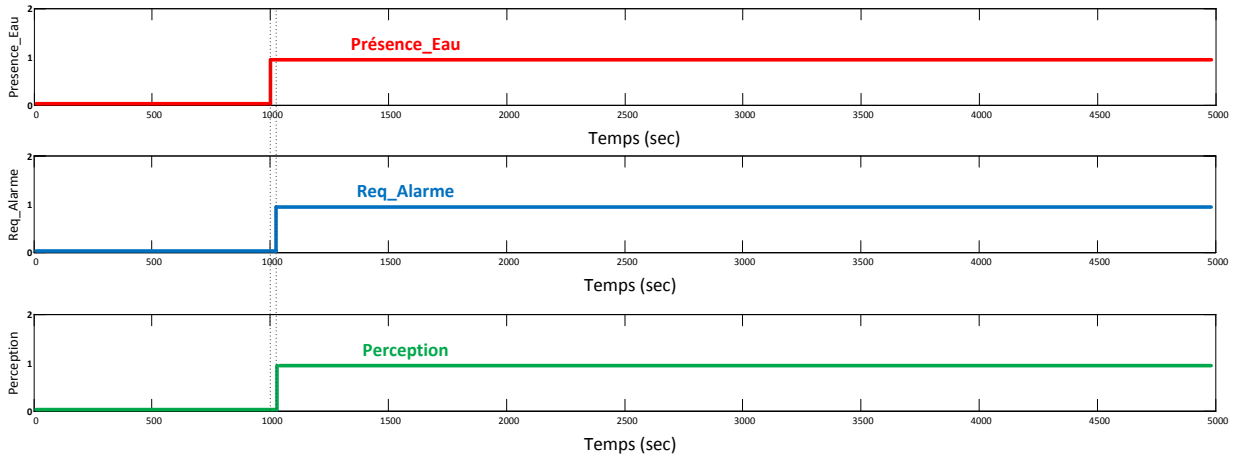


Figure 174 : Trace d'exécution du scénario 2 de validation de l'interaction de détection sonore

8.3.2. Intégration d'un composant constitutif de l'interaction physico-physiologique ciblée

Ces travaux exploratoires, en complément de ceux de (Lieber, 2013), posent cependant la question de la validation système d'un modèle « in-silico »[75] au facteur d'échelle de l'ensemble des facteurs humains dans un contexte donné.

Les travaux complémentaires présentés dans ce paragraphe ont pour objet de montrer que notre environnement de travail pourrait supporter une approche de spécification itérative incluant des plans d'expériences « in-situ » de façon complémentaire, voire alternative, à la modélisation numérique de l'interaction physiologique présentée précédemment. En effet, le modèle constitutif d'interaction de détection sonore est construit de manière théorique selon les fondements de la MTIP et correspond à un premier niveau de spécification pour la compréhension des interactions entre un humain et son environnement. En ce sens, les caractéristiques du constituant humain « Puits » sont décrites selon des caractéristiques générales, ne dépendant pas de :

- Des multi-interactions avec les autres sens : Des études récentes montrent que les frontières sensorielles sont floues et que les cinq sens interagissent les uns avec les autres (Bleicher, 2014)

[75] Expression courante en « Ergonomie » par rapport à « in-vivo » pour qualifier les situations de modélisation de l'Humain

- La variabilité chez les humains : Les opérateurs diffèrent les uns des autres de par leurs caractéristiques physiologiques, psychologiques et cognitives. Par exemple, la qualité de la vue et de l'ouïe dépendent de l'âge.
- Des situations de conduite : il a été montré que les situations incidentelles ou accidentelles sources de stress, peuvent avoir des effets sur l'attention, la mémoire, la performance perceptivo-motrice ou les capacités de jugements et de décision (Staal, 2004) aboutissant in fine à une perturbation de la performance humaine opérationnelle.

Il apparait ainsi que le constituant de détection sonore étudié est fonction de nombreux facteurs[76] (pouvant être dépendants les uns des autres). Le problème du modélisateur est alors d'identifier l'influence de ces facteurs sur la réponse[77] du système. N'ayant pas l'ensemble des connaissances pour relier mathématiquement ces paramètres entre eux, cette étape peut s'effectuer de manière expérimentale en imposant différentes valeurs aux facteurs et en mesurant les réponses obtenues.

Parmi les méthodes expérimentales existantes, nous pouvons citer les méthodes statistiques telles que les plans *d'expériences consistant à sélectionner et ordonner les essais afin d'identifier, à moindres coûts les effets des paramètres sur la réponse du produit*[78]*. Cette méthode comporte trois étapes :*

1. *Postuler un modèle de comportement du système* (Nous pouvons alors nous référer au modèle obtenu selon la MTIP qui permet de caractériser certains paramètres physiologiques
2. *Définir un plan d'expériences, c'est-à-dire une série d'essais permettant d'identifier l'influence des facteurs,*
3. *Faire les essais, identifier les coefficients et conclure.*

Maquette d'interaction physico-physiologique

Afin de mettre en œuvre ce type de plan d'expérience, il a été nécessaire de concevoir une maquette « d'interaction physico-physiologique » permettant d'évaluer l'influence de différents facteurs (*puissance sonore, fréquence sonore, bruits ambiants, âge des opérateurs, position dans l'espace…*) sur la bonne détection de l'alarme sonore. Dans l'objectif de reproduire au plus près la réalité, nous avons conçu cette maquette pour qu'elle puisse s'intégrer à la plate-forme CISPI existante et ainsi être immerger dans l'environnement opérationnel lors des différentes séries d'essais.

En ce sens, cette maquette offre la possibilité de sélectionner plusieurs alarmes (de puissance, et de fréquence différentes) à partir d'une interface graphique (Figure 175).

[76] Facteur : Paramètre que l'on fait varier au cours des essais
[77] Réponse : Grandeur mesurée lors des essais
[78] http://www.si.ens-cachan.fr/accueil_V2.php?id=19&page=affiche_ressource

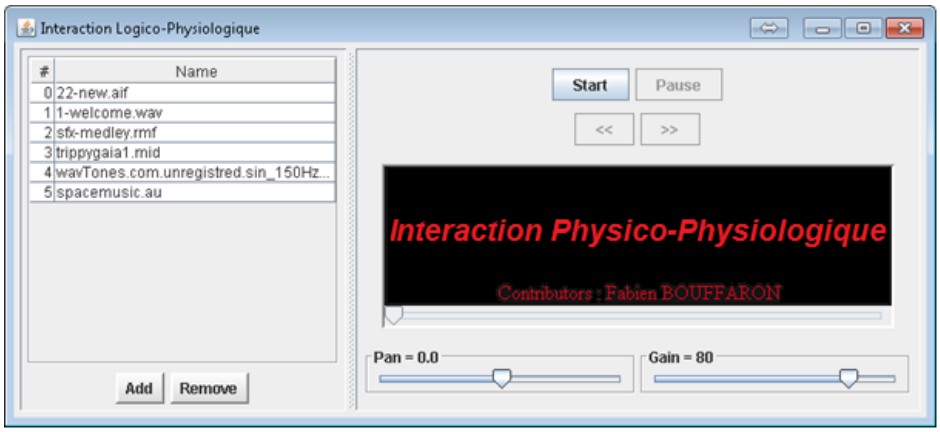


Figure 175 : Interface de la maquette physiologique d'interaction Physico-Physiologique (configuration de l'interaction de détection sonore d'une alarme technique)

Reliée physiquement à la plate-forme CISPI par le réseau de terrain, la maquette se connecte au serveur OPC de la plate-forme pour lire cycliquement une variable automate correspondant à l'ordre (***ReqAlarme***) de mise en service (hors-service) de l'alarme en cas d'une situation incidentelle/ accidentelle (Figure 176).

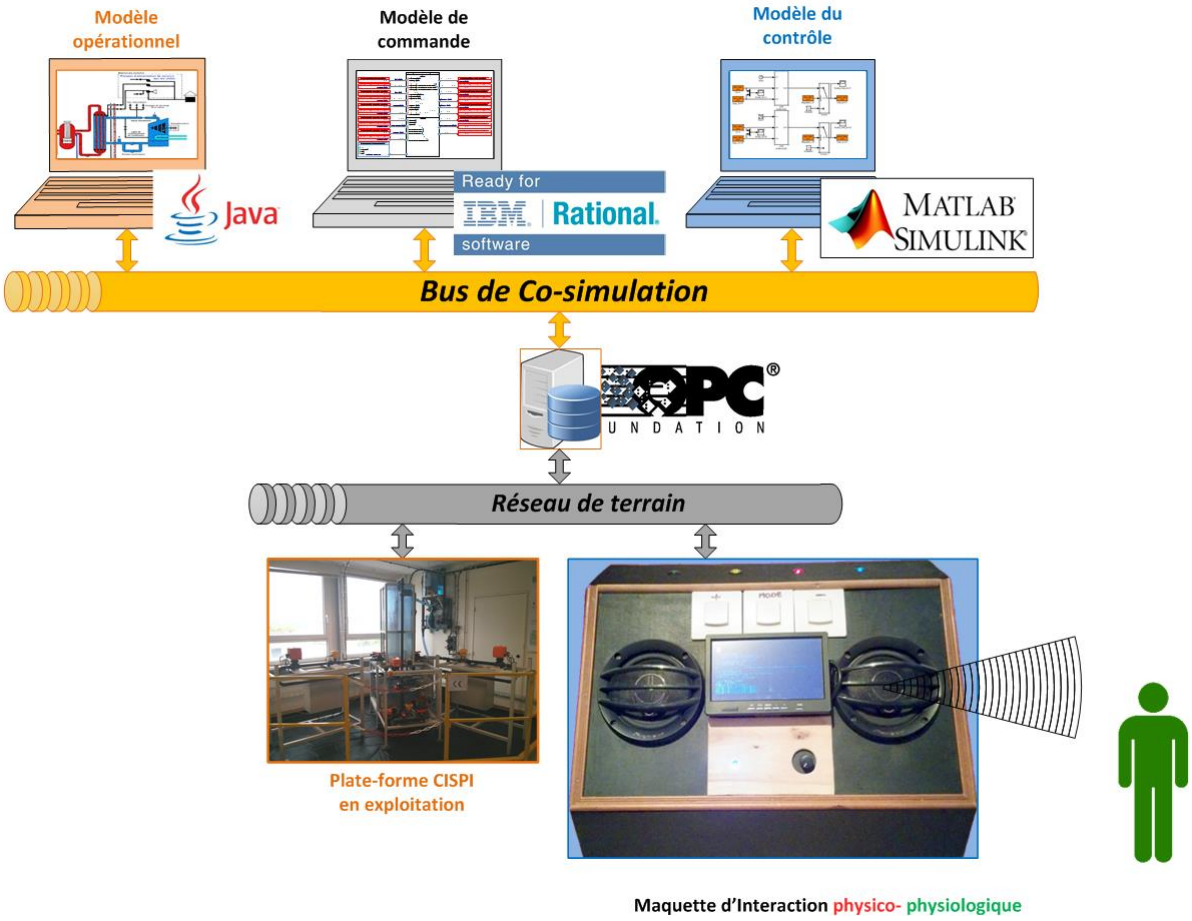


Figure 176 : Maquette physico-physiologique en interface avec la plate-forme CISPI

Le cadre d'expérimentation étant défini, le plan d'expérience à mettre en œuvre consisterait à mesurer selon un panel d'opérateurs le plus représentatif possible à partir des conditions initiales de l'interaction de détection sonore s'il y a perception ou non de l'alarme.

Pour des raisons de temps, nous n'avons pas pu réaliser l'ensemble de ces mesures qui devrait s'appuyer sur les ressources techniques d'un organisme spécialisé, tel que l'Institut National de Recherche et de Sécurité de Nancy,. Il est certain que les résultats dérivés de cette expérience nous auraient conduit à raffiner notre modèle d'interaction de détection sonore afin de spécifier au mieux les contraintes requises pour l'alarme technique permettant ainsi d'assurer la bonne détection du signal sonore. Notons cependant que le modèle « in silico » permet de cibler, en complément de techniques de plans d'expériences fractionnaires (Taguchi, 1987), certains paramètres pertinents d'interaction système afin de mettre en place la bonne stratégie pour minimiser le nombre d'essais à réaliser et, simultanément maximiser le nombre de facteurs à étudier.

	Opérateur 1 Age : xxx Sexe : xxx	Opérateur 2 Age : yyy Sexe : yyy	Opérateur 3 Age : zzz Sexe : zzz
Puissance de l'alarme : 60 dB			
Puissance de l'alarme : 80 dB			
Puissance de l'alarme : 90 dB			
Puissance de l'alarme : 100 dB			
Puissance de l'alarme : 110 dB			
Puissance de l'alarme : 120 dB			
Fréquence de l'alarme : 20 Hz			
Fréquence de l'alarme : 2520 Hz			
Fréquence de l'alarme : 5020 Hz			
Fréquence de l'alarme : 7520 Hz			
Fréquence de l'alarme : 10020 Hz			
Fréquence de l'alarme : 12520 Hz			
Fréquence de l'alarme : 15020 Hz			
Fréquence de l'alarme : 17520 Hz			
Fréquence de l'alarme : 20020 Hz			
Fréquence de l'alarme : 22520 Hz			
Fréquence de l'alarme : 25020 Hz			
Distance Alarme (Source) / Opérateur Puits : 1m			
Distance Alarme (Source) / Opérateur Puits : 2m			
Distance Alarme (Source) / Opérateur Puits : 3m			
Distance Alarme (Source) / Opérateur Puits : 4m			
Distance Alarme (Source) / Opérateur Puits : 5m			
Distance Alarme (Source) / Opérateur Puits : 6m			
Distance Alarme (Source) / Opérateur Puits : 7m			
Distance Alarme (Source) / Opérateur Puits : 8m			
Bruit ambiant : 50 dB			
Bruit ambiant : 60 dB			
Bruit ambiant : 70 dB			
Bruit ambiant : 80 dB			
Bruit ambiant : 90 dB			
Bruit ambiant : 100 dB			
Situation normale de conduite			
Situation incidentelle de conduite			

Tableau 19 : Facteurs d'influence pour la détection d'une alarme sonore (Source) par un agent humain (Puits) : Puissance de l'alarme, Fréquence, Distance, Bruit Ambiant, Situation de conduite, Age, Sexe. L'ensemble des essais à mener correspondent à la combinatoire de ces facteurs, pouvant être regroupés sous la forme d'une matrice d'expérience orthogonale

8.4. Conclusion : Co-spécification exécutable d'un modèle partiel constitutif de la partie « mesurage intelligent » ciblée

L'objectif de ce chapitre était de spécifier un élément, sous forme de modèle puis de composant, qui soit constitutif d'une interaction Homme-Artefact du système CISPI. Il a été choisi de limiter cette co-spécification d'un constituant de mesurage intelligent à une interaction physico-physiologique de détection sonore.

Ce constituant-système de détection d'une alarme sonore a été co-spécifié entre un **espace-solution** d'ingénierie physiologique à partir du cadre de modélisation de (Lieber, 2013) et un **espace-problème** d'ingénierie système en charge d'architecturer un modèle global exécutable. Ce modèle « in-silico » a permis de déterminer les conditions nécessaires mais non suffisantes à la bonne détection du signal sonore « in-situ » par un agent humain de conduite. Ces conditions se traduisent d'un point de vue technique par le dimensionnement physique du constituant technique « source » afin de garantir l'interaction physico-physiologique de détection sonore par un constituant humain « puits ». Afin de s'assurer de la bonne détection de l'alarme sonore par un agent humain en condition opérationnelle de conduite, la validation-système de ce constituant a été réalisée par co-exécution de modèles avec la plate-forme CISPI dans un premier temps. Nous avons cependant montré, dans un deuxième temps, que notre approche pourrait permettre de guider la construction de plans d'expériences, qui sont les techniques courantes de modélisation au facteur d'échelle de l'ensemble des facteurs humains à prendre en compte en situation. Ces résultats sur la modélisation exécutable d'une interaction physico-physiologique contribuent à la définition de critères mesurables de propriétés physiologiques au plus tôt de l'intégration d'un constituant humain en ingénierie centrée système, comme recommandé par (Boy & Narkevicius, 2013) :

> *Traditionally, engineering focused on designing and developing machines that were further tested by human factors and ergonomics specialists. Incorporating HSI into SE resulted in identifying and incorporating human capabilities and limitation in system development.*

Partie 4 : Co-spécification exécutable de constituants d'actionnement et de mesurage du système de conduite CISPI

Conclusion : Co-spécification d'un modèle total constitutif de l'interaction logico-physico-physiologique ciblée

Nous venons d'illustrer dans cette partie la mise en œuvre de notre environnement de co-spécification système pour la définition de deux modèles constitutifs de l'interaction logico-physique entre le sous-système logique et le sous-système physique de la plate-forme CISPI. En ce sens, nous nous sommes focalisés sur deux interactions de nature différentes :

- interaction technico-physique : entre un constituant technique d'actionnement intelligent « Source » et le processus CISPI « Puits »
- interaction physico-physiologique : entre une alarme sonore « Source » et un constituant humain « Puits » de mesure intelligent

Nous défendons que les propriétés de l'élément « Puits » d'une interaction « Source-Puits » conditionne la définition de l'élément « Source » afin de garantir la « bonne » synergie entre ces éléments. Ainsi que ce soit les règles de la physique pour le processus CISPI ou les capacités physiologiques d'un humain, ces caractéristiques contraignent la définition de l'élément « Source » en interaction avec eux.

En ce sens, nous voyons dans ces travaux la possibilité de rompre avec les méthodes traditionnelles d'ingénierie consistant à répartir les fonctions de conduite de manière implicite sur la base de savoir-faire, de retours d'expériences, de coûts d'investissements ou de technologies disponibles (Galara & Hennebicq, 1999). En effet, ces caractéristiques fondées sur des propriétés mesurables permettent d'envisager la mise en place de méthodes d'évaluation basées sur des critères rationnels pour la répartition des fonctions de conduite entre des agents techniques ou humains. Nos travaux sont ainsi en phase avec la vision 2020 de l'ingénierie système par l'INCOSE (INCOSE, 2007) (AFIS, 2012) quant à l'anticipation des performances des systèmes critiques, comme les systèmes homme-machine, et la maîtrise de leur développement au plus tôt du projet, via des modèles numériques, ainsi que leur couplage à travers des techniques de co–simulation.

Conclusion et perspectives générales

Conclusion et perspectives générales :

Nous avons pris prétexte du constat à postériori de dysfonctionnements systémiques (Boardman & Sauser, 2013b) et d'un nouveau paradigme d'Ingénierie Système Basée sur l'exécution de Modèles (Garcia, 2009) pour revisiter la nature « totalisante (holonique)» de la relation d'architecture-système (Claeys, 2013) en regard d'une situation-système visée.

Nous nous sommes plus particulièrement attaché à architecturer deux « **totalités artéfactuelles** » (*constituants de contrôle et de commande*) fonctionnalisant le comportement de conduite du procédé critique ciblé avec deux « **totalités naturelles** » (*Processus hydraulique (matière/énergie), Constituant humain de conduite*) dont on ne peut garantir que partiellement le comportement en réaction à toute décision prescrite de conduite. Cette perspective est en ligne avec celle « **système de systèmes** » qui prévaut au constat ci-dessus et manière plus générale au questionnement récurrent de la communauté d'« Ingénierie Système » depuis une vingtaine d'années, bien que notre situation d'intérêt ne soit pas organisationnelle.

La construction concourante d'une faisabilité virtuelle du système d'intérêt visé concrétise le couplage entre représentations systémiques du réel et interdisciplinaires d'ingénierie. Nos travaux, au facteur d'échelle ciblé en **partie 1**, montre l'intérêt de cette vision pour tendre vers une ingénierie centrée système. Notons que le pivot de cette construction-système reste néanmoins artéfactuel pour asseoir cette faisabilité sur la partie totalement prescriptible en regard de la partie naturelle dont nous voulons tenir compte au plus tôt de certaines contraintes sans distinguer de plus ce qui relèverait d'une totalité physique ou biologique. Des travaux devraient être menés pour comparer cette faisabilité système avec celles qui résulteraient d'autres organisations collaboratives entre parties prenantes « techniques », «physiques », « humaines », …

Nous avons proposé en **partie 2** une heuristique interprétant la nature spécifiante et itérative du processus d'architecture-système selon l'approche des « problem-frames ». La généralisation (Figure 177) que nous pouvons faire en retour de mise en œuvre de cette heuristique montre les dimensions de connaissances collectives et individuelles (inspiré de (Wilber, 1997) du processus collaboratif étudié (Tableau 20).

System of Interest - Systemic Architecting		**System of Interest - Virtual Architecting**
Whole	*Quadrant 1 : collective subjective specification* *problem-oriented :* K_S, $R_S \rightarrow R_H$ *solution-oriented :* $M_H \leftarrow R_S$, K_H	*Quadrant 4 : interdisciplinary subjective specification* *problem-oriented :* K_H, $R_H \rightarrow R_I$ *solution-oriented :* $M_I \leftarrow R_H$, K_I
Coupling	*Quadrant 2: collaborative objective specification* *situation-oriented :* K_S, $M_H \vdash R_S$ *system-oriented :* K_H, $M_h \vdash R_H$ *situation-oriented :* K_s, $M_h \vdash R_s$	*Quadrant 5 : collaborative objective specification* *system-oriented :* K_H, $M_I \vdash R_H$ *constituent-oriented :* K_I, $M_P \vdash R_I$ *system-oriented :* K_h, $M_P \vdash R_h$
Part	*Quadrant 3 : individual subjective specification* *problem-oriented :* K_s, $R_s \rightarrow R_h$ *solution-oriented :* $M_h \leftarrow R_s$, K_h	*Quadrant 6 : disciplinary subjective specification* *problem-oriented :* K_h, $R_h \rightarrow R_P$ *solution-oriented :* $M_P \leftarrow R_h$, K_P

Tableau 20 : Interprétation de l'heuristique de co-spécification système selon le modèle AQAL (All quadrants, All Level). Inspirée de (Wilber, 1997)

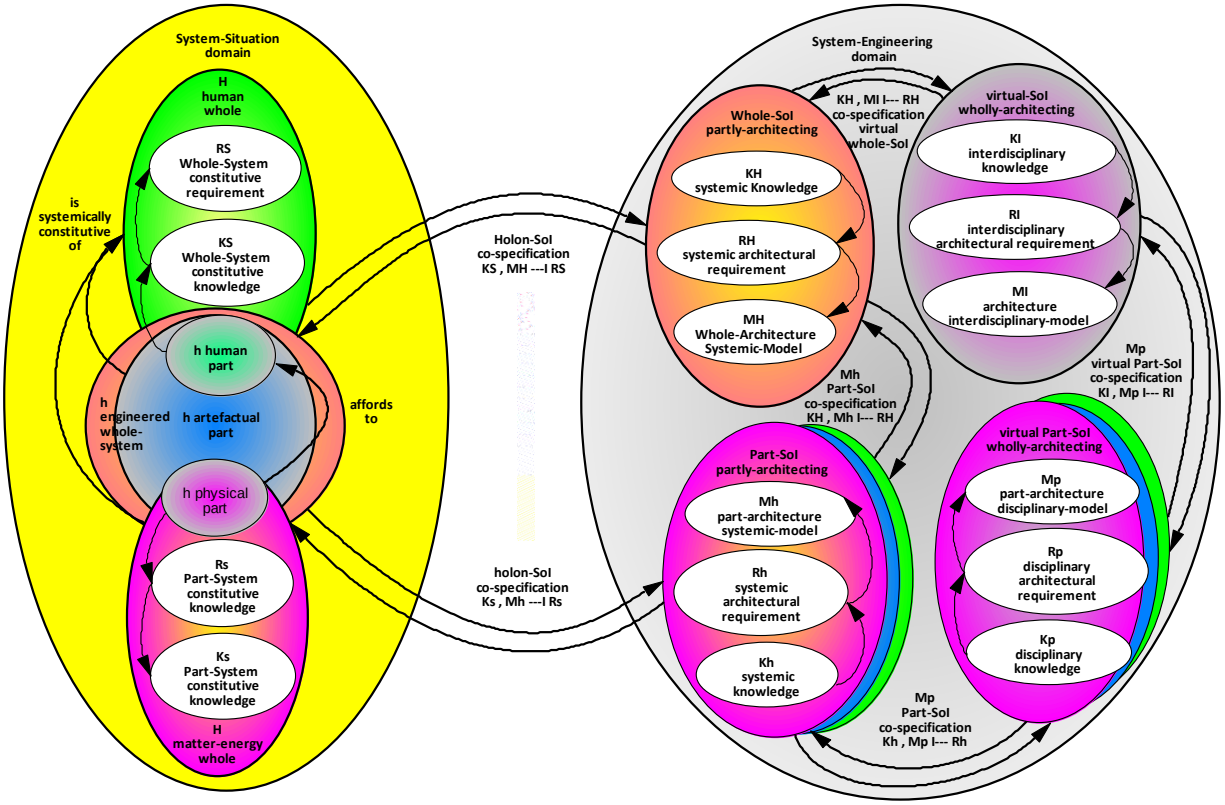


Figure 177 : Généralisation de l'heuristique de co-spécification système étudiée

La co-spécification systémique (Quadrants 1, 2 & 3) fait du « conceptagon » le patron architectural du système d'intérêt à faire pour coupler, vers une même situation, un ensemble :

- De connaissances subjectives (K_S) et (K_s) en une connaissance objective (K_H) de la totalité requise
- D'exigences optatives (R_S) et (R_s) en un ensemble d'exigences architecturales (R_H) du système d'intérêt requis
- De modèles constitutifs (M_h) en un modèle de compréhension (M_H) du système d'intérêt requis

Notons que la visée architecturale (Quadrant 2) de cette spécification systémique diffère ainsi des pratiques courantes d'abstraction pour « penser-système » (Figure 178) en objectivant au plus tôt aussi bien les parties (Quadrant 3) au regard de modèles de faisabilité disciplinaires (Quadrant 6), que la totalité visée (Quadrant 1) au regard d'un modèles de faisabilité interdisciplinaire (Quadrant 4).

La modélisation multi-physique en chapitre 5 illustre ce « feedback » afin de faciliter la notification de chaque expertise spécialiste pour représenter totalement une réalité de la matière-énergie que ne permet pas une modélisation processive plus abstraite ou discrétisée par objet. Précisons aussi que des règles pragmatiques de modélisation systémique complètent ce processus pour, par exemple, repérer l'«objet circulant» qui donne originellement vie à la mission du système d'intérêt ou pour définir la frontière entre ce qui est contrôlable intérieurement de ce qui ne l'est pas extérieurement mais qui influence son comportement total.

Figure 178 : The System Thinker's Tree (Smith & Daniel-Allegro, 2015)

La faisabilité sous forme virtuelle (Quadrants 4, 5 & 6) de cette co-spécification systémique cible une concrétisation concourante à la compréhension récursive du modèle du système d'intérêt (M_H). Elle s'appuie sur un patron architectural « CMMS » afin d'intégrer un ensemble :

- de connaissances disciplinaires (K_P) en une connaissance interdisciplinaire (K_I) visant objectivement (K_H)
- d'exigences réalisationnelles (R_P) en un ensemble d'exigence architecturale de faisabilité (R_h) du système d'intérêt visé
- de modèles disciplinaires exécutables (M_h) en un modèle interdisciplinaire exécutable (M_I) du système d'intérêt visé

L'ensemble des relations (K_{source}, $R_{source} \rightarrow R_{puits}$) et ($M_{puits} \leftarrow K_{puits}$, R_{source}) traduit la nature descriptive puis prescriptive de la relation de spécification entre espace-problème requérant et espace-solution délivrant menant à une solution vérifiée (M_{puits}). L'ensemble des relations (K_{source}, $M_{puits} \vdash R_{source}$) traduit la validation contractuelle de cette solution menant par raffinements successifs à la validation opérationnelle d'un système ingéniéré ($K_{situation}$, $S \vdash R_{opérationnelle}$) de façon plus générale que dans notre expérimentation.

Il est important de remarquer que la connaissance perçue par un espace-problème pour requérir (R_{puits}) un espace-solution diffère de celle (K_{puits}) que cet espace met en œuvre pour prescrire une solution (M_{puits}). Pour faciliter cette expression de chaque connaissance experte, nos travaux ont ainsi contribué, d'un point de vue

technique, à éprouver l'intérêt des outils collaboratifs d'échanges de modèles et de connaissances, l'usage des différents diagrammes du langage de modélisation système SysML ainsi que des outils co-simulation système pour supporter notre heuristique de co-spécification système. Le tableau 2 illustre le déploiement de l'ensemble de ces moyens pour la mise en œuvre des scénarios des **parties 3 et 4.**

	System of Interest - Systemic Architecting	System of Interest - Virtual Architecting
Whole	*Problem oriented :* • *Diagrammes d'exigences SysML / Diagrammes de cas d'utilisations SysML* *Solution oriented :* • *Plate-forme CISPI du CRAN (Chapitre 1)*	*Problem oriented :* • *Diagrammes d'exigences SysML / Diagrammes de cas d'utilisations SysML* *Solution oriented :* • *Diagrammes Internes de Blocs (SysML) représentant les architectures interdisciplinaires de co-simulation système entre les différents modèles d'ingénierie disciplinaire (Partie 2 et Partie 3)*
Coupling	*Situation & System oriented :* • *Couplage du bus de terrain de la plate-forme CISPI avec le bus de co-simulation système via le serveur OPC (Chapitre 7)* • *Mise en situation opérationnelle (CISPI) de l'opérateur de conduite pour l'évaluation des paramètres influençant la détection sonore (Chapitre 8)*	*System & Component oriented :* • *Bus de co-simulation (Cosimate) permettant d'orchestrer l'échange de données entre les différents modèles disciplinaires des ingénieries spécialistes (Partie 2 et Partie 3)*
Part	*Problem oriented :* • *Diagrammes d'exigences SysML / Diagrammes de cas d'utilisations SysML* *Solution oriented :* • *Processus hydraulique de CISPI (Vannes, Débitmètre, Cuves….) (Chapitre 7)* • *Opérateurs de conduite CISPI (salle de commande, terrain) (Chapitre 8)* •	*Problem oriented :* • *Diagrammes d'exigences SysML / Diagrammes de cas d'utilisations SysML* *Solution oriented :* • *Modèle du procédé CISPI (ISO 10628) ➔ Translation Diagrammes d'activités SysML (Chapitre 5)* • *Modèle du processus CISPI (Modelica) ➔ Translation Diagrammes Internes de Blocs SysML + Diagrammes Paramétriques SysML (Chapitre 5)* • *Modèle de Contrôle (Simulink) ➔ Translation Diagrammes Internes de blocs SysML / Diagrammes paramétrique SysML (Chapitre 6)* • *Modèle de Commande (Simulink) ➔ Translation Diagrammes Internes de blocs SysML / Diagrammes d'états SysML (Chapitre 6)* • *Modèle Physiologique (Simulink) ➔ Translation Diagrammes internes de blocs SysML / Diagrammes paramétrique SysML (Chapitre 8)*

Tableau 21 : Mise en œuvre de l'heuristique de co-spécification système

D'un point de vue pédagogique, ces travaux ont permis de décliner une heuristique de co-spécification système qui constitue maintenant la base d'une démarche pour l'enseignement par projet à l'ISBM (G. Morel, et al., 2014).

Cependant malgré cet intérêt formatif, nos travaux n'en restent pas moins exploratoires et nécessitent un passage à une échelle pour une preuve de maturité supérieure à celle que nous nous sommes efforcés de mettre en œuvre lors de la définition de la plate-forme CISPI du CRAN. Cette évolution se concrétise notamment à travers la définition d'un environnement de co-spécification et de co-simulation plus mature

prenant en compte le partage et l'exécution de modèles de manière complétement intégré. Cela va dans le sens du standard FMI (Functional Mock-up Interfaces) qui ouvre des perspectives pour l'échange de modèles et la co-simulation.

De plus, à un facteur d'échelle industriel, cet environnement de co-spécification exécutable basée sur des modèles soulève des questions relatives à la co-simulation et notamment à la configuration du bus de co-simulation ainsi que des différents outils de simulation, afin de garantir le comportement global du SYSTEME en simulation. En ce sens, la création d'un nouveau métier d'architecte simulation (Retho, 2015) en interaction avec l'ingénieur système nous apparait nécessaire pour faire face à cette complexité. En complémentarité des travaux de (Lieber, 2013) qui prônent un architecte centré humain en pendant de l'architecte centré technique, nos travaux montrent l'intérêt de généraliser cette vision à ce qui relève du naturel (humain, matière/énergie, écologie) de façon générale.

Enfin, d'un point de vue plus général, la mise en œuvre de notre heuristique de co-spécification système requiert un changement des organisations en entreprises afin d'évoluer d'une ingénierie centrée documents vers une ingénierie SYSTEME centrée sur des modèles exécutables. Bien que cette évolution soit en marche, il reste encore un palier important à franchir pour déployer ces techniques de co-simulation d'un point de vue industriel, requérant pour chacun des projets une analyse entre les investissements pour simuler les modèles et les apports vis-à-vis des contraintes de coûts, de temps et de qualité du produit à fournir.

Pour finir, nos nombreuses interactions avec différents domaines d'ingénieries spécialistes (physique, automatisation et physiologique) nous ont permis d'obtenir des résultats prometteurs d'un point de vue de la modélisation disciplinaire selon une perspective SYSTEME, requérant cependant d'être explorés plus en détails :

- La bibliothèque Modelica développée dans le chapitre 5 pour la modélisation du procédé hydraulique de la plateforme CISPI à un niveau d'abstraction fonctionnelle nécessite d'être généralisée pour être utiliser pour des systèmes multi-physiques (Thermique, Electrique, Mécanique …..).
- La dynamique du paramètre F estimé en ligne par la CSM (Commande Sans Modèles) mérite d'être étudié plus en détail afin d'en extraire des informations plus précises pour l'étude de dérive et la détection de défaut du système dans la perspective de relier les sous-systèmes de conduite, de gestion technique et de maintenance composant l'architecture CMMS-IAMS retenue.
- La modélisation des interactions physico-physiologiques en puissance, dans le chapitre 8, nécessite d'être étendue à l'ensemble des capteurs sensoriels chez l'humain (odorat, vue toucher, gout) afin de se référer à un ensemble de propriétés mesurables et homogènes quel que soit le capteur (qu'il soit technique ou humain) lors de l'évaluation des performances de ces derniers pour la répartition des fonctions de conduite d'actionnement et de mesures entre des agents techniques ou humains.

Bibliographie

Abrial, J., & Hoare, A. (2005). *The B-book: assigning programs to meanings*. Cambridge University Press.

AFIS - GT Ingénierie Système. (2005). Processus de l'Ingénierie Système - Qu'est ce qu'un processus ? . In: AFIS (Ed.).

Amalberti, R. (1996). *La conduite de systèmes à risques: le travail à l'hôpital*. Presses universitaires de France.

Andreasson, J. (2003). VehicleDynamics library. In: *Proceedings of the 3rd International Modelica Conference, Linköping*. Citeseer.

ANSI-EIA-632. (1999). Processes for engineering a system. In. Electronic Industries Alliance, Governement Electronics And Information Technolo Association Engineering Departmen, EIA Standard.

Appell, B., & Chabert, J. (1998). Contrôle-commande des réacteurs et des usines: architecture générale. *Techniques de l'ingénieur. Génie nucléaire* (BN3411), BN3411. 3411-BN3411. 3422.

Appell, B., & Chambon, Y. (1998). Procédures de conduite et interface homme-machine. *Techniques de l'ingénieur. Génie nucléaire* (BN3421), 1-12.

Artman, H., & Garbis, C. (1998). Team communication and coordination as distributed cognition. In: *9th Conference of Cognitive Ergonomics* (pp. 151-156), pp. 151-156.

Auzelle, J. (2009). *Proposition d'un cadre de modélisation multi-échelles d'un système d'information en entreprise centré sur le produit.* Université Henri Poincaré-Nancy I.

Baïna, S. (2006). *Interoperabilité dirigée par les modèles : Une Approche Orientée Produit pour l'interopérabilité des systèmes d'entreprise.* Université Henri Poincaré-Nancy I.

Bajic, E. (2005). Ambient services modelling framework for intelligent products. *Smart Object Systems, UbiComp*, 83-90.

Bear, M., Paradiso, M., Connors, B., & Neuroscience, M. (1996). Exploring the brain. *Neuroscience: Williams & Wilkins*.

Belhimeur, A. (1989). *Contribution à l'étude d'une méthode de conception des automatismes des systèmes de conduite des processus industriels.* Lille 1.

Bellifemine, F., Poggi, A., & Rimassa, G. (1999). JADE–A FIPA-compliant agent framework. In: *Proceedings of PAAM* (Vol. 99, pp. 33). London, pp. 33.

Bennani, O., Chauvet, G., Chauvet, P., Dupont, J., & Jouen, F. (2009). A hierarchical modeling approach of hippocampus local circuit. *Journal of integrative neuroscience,* 8 (01), 49-76.

Bennani, O., Chauvet, P., & Dupont, J. (2010). The Mathematical Theory of Integrative Physiology: Application to hippocampus for drug discovery. In: *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on* (pp. 1406-1416). IEEE, pp. 1406-1416.

Berthoz, A. (1997). *Sens du mouvement (Le)*. Odile Jacob.

Berthoz, A. (2009). *Simplexité (La)*. Odile Jacob.

Berthoz, A. (2014). Simplexity: Biologically Inspired Principles For Complex Systems ? Paper presented at the CSD&M 2014 (Complex Systems Design & Management), Paris.

Bertsch, C., & Schulmeister, E. (2014). The Functional Mockup Interface-seen from an industrial perspective. In: *10 th International Modelica Conference, Lund, Sweden*.

Birraux, C., Bataille, C., & Sido, B. (2011). Rapport de la mission parlementaire sur la sécurité nucléaire, la place de la filière et son avenir. In: *Vol. N°3614 (Assemblée) et N°701 (Sénat)*. Assemblée nationale, Sénat.

Bjelkemyr, M., & Lindberg, B. (2007). The Effects of Limits to Human Abilities on System of Systems Properties. In: *The 7th International Conference on Complex Systems October*.

Bjørner, D. (2008). From Domain to Requirements. *Concurrency, Graphs and Models*, 278-300.

Bjørner, D. (2009). From Domains to Requirements. On a Triptych of Software Development.

Bjørner, D., & Henson, M. (2007). *Logics of specification languages*. Springer Science & Business Media.

Bleicher, A. (2014). Le mélange ses sens. *Cerveau & Psycho,* 61.

Blochwitz, T., Otter, M., Åkesson, J., Arnold, M., Clauss, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., & Neumerkel, D. (2012). Functional mockup interface 2.0 : The standard for tool independent exchange of simulation models. In: *9th International Modelica Conference*.

Boardman, J., & Sauser, B. (2006). Systomics : In Search of the System Genome.

Boardman, J., & Sauser, B. (2008). *Systems thinking: Coping with 21st century problems*. CRC Press.

Boardman, J., & Sauser, B. (2013a). *Systemic thinking : building maps for worlds of systems*. John Wiley & Sons.

Boardman, J., & Sauser, B. (2013b). *Systemic thinking: building maps for worlds of systems*. John Wiley & Sons.

Boardman, J., Sauser, B., John, L., & Edson, R. (2009). The conceptagon: A framework for systems thinking and systems practice. In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* (pp. 3299-3304). IEEE, pp. 3299-3304.

Boardman, J., Sauser, B., & Verma, D. (2009). In Search of Systems "DNA. *Journal of Computers,* 4 (10), 1043-1052.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer,* 21 (5), 61-72.

Boes, J. (2014). *Apprentissage du contrôle de systèmes complexes par l'auto-organisation coopérative d'un système multi-agent : application à la calibration de moteurs à combustion.* Université de Toulouse, Université Toulouse III-Paul Sabatier.

Bibliographie

Bouffaron, F. (2014). Co-Spécification exécutable basée sur des modèles : application à la conduite d'un procédé industriel In: *IBM Symposium Système 2014*, Paris.

Bouffaron, F., Dupont, J., Mayer, F., & Morel, G. (2014). Integrative construct for Model-Based Human-System Integration: a case study. Paper presented at the World Ifac Congress 2014, Cape Town - South Africa.

Bouffaron, F., Gouyon, D., Dobre, D., & Morel, G. (2012). Revisiting the interoperation relationships between Systems Engineering collaborative processes. Paper presented at the INCOM 2012, 14th IFAC Symposium on Information Control Problems in Manufacturing, Bucharest, Romania.

Bouffaron, F., Marange, P., & Morel, G. (2014). Checking models based on an iterative co-specification process of a critical system. Paper presented at the INDIN (Internation Conference On Industrial Informatics) Porto Alegre, Brazil.

Boy, G. (2005). Human-centered design: The AUTOS pyramid. *EURISCO International Newsletter,* 4.

Boy, G. (2011). *The handbook of human-machine interaction: a human-centered design approach*. Ashgate Publishing.

Boy, G. A., & Narkevicius, J. M. (2014). Unifying human centered design and systems engineering for human systems integration. In: *Complex Systems Design & Management* (pp. 151-162). Springer, pp. 151-162.

Carot, S. (1994). *La cristallisation en chimie fine : optimisation des paramètres de régulation de température et application à l'élaboration de principes actifs de médicaments.*

Cassandras, C., & Braatz, R. (2011). People In Control. *IEEE Control Systems Magazine,* 31 (6), 24.

Cassandras, C., & Lafortune, S. (1999). *Introduction to discrete event systems*. Kluwer Academic Publishers, Dordrecht.

Cellier, F., & Nebot, À. (2005). The Modelica bond graph library. In: *the 4th International Modelica Conference*.

Chauvet, G. (1993). Hierarchical functional organization of formal biological systems: a dynamical approach. III. The concept of non-locality leads to a field theory describing the dynamics at each level of organization of the (D-FBS) sub-system. *Philosophical Transactions of the Royal Society of London B: Biological Sciences,* 339 (1290), 463-481.

Chauvet, G. (1995). *La vie dans la matière: le rôle de l'espace en biologie*. Flammarion [Paris].

Chauvet, G. (2002). On the mathematical integration of the nervous tissue based on the S-propagator formalism I: Theory. *Journal of integrative neuroscience,* 1 (01), 31-68.

Claeys, D. (2013). *Architecture et complexité: Un modèle systémique du processus de (co) conception qui vise l'architecture.* Université de Louvain.

Clanché, F., Gouyon, D., Dobre, D., Pétin, J. F., & Morel, G. (2010). Plate-forme pour la conduite interactive et sûre.

Cloutier, R. (2006). *Applicability of patterns to architecting complex systems.* Stevens Institute of Technology.

Cloutier, R., & Bone, M. (2010). Compilation of SysML RFI–Final Report. *Stevens Institute of Technology*.

Cloutier, R., Sauser, B., Bone, M., & Taylor, A. (2015). Transitioning Systems Thinking to Model-Based Systems Engineering: Systemigrams to SysML Models. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on,* 45 (4), 662-674.

Coppolani, P. (2004). *La chaudière des réacteurs à eau sous pression*. EDP sciences.

Cott, B., & Macchietto, S. (1989). Temperature control of exothermic batch reactors using generic model control. *Industrial & engineering chemistry research,* 28 (8), 1177-1184.

Cressent, R., Idasiak, V., & Kratz, F. (2011). Intégration des analyses de sûreté de fonctionnement dans le processus de conception système. In: *4èmes Journées Doctorales/Journées Nationales MACS* (pp. Intégration des analyses de sûreté de fonctionnement dans le processus de conception système), pp. Intégration des analyses de sûreté de fonctionnement dans le processus de conception système.

Czarnecki, K. (1998). *Generative programming: Principles and techniques of software engineering based on automated configuration and fragment-based component models.* Unpublished PhD thesis, Technical University Of Ilmenau.

De Miras, J., Riachy, S., Fliess, M., Join, C., & Bonnet, S. (2012). Vers une commande sans modèle d'un palier magnétique. In: *7e Conférence Internationale Francophone d'Automatique (CIFA)* (pp. CDROM), pp. CDROM.

Després, F. (1991). *Automatisation des systèmes de production - du besoin à l'utilisation* (KIRK edition ed.).

Devic, C. (2014). CONNEXION COntröle Commande Nucléaire Numérique pour l'EXport et la rénovatION - De la schématique proressive à la centrale numériqe au service de la productivité des études. *Génie Logiciel,* N° 109, 2-11.

Devic, C., & Morilhat, P. (2013). CONNEXION COntröle Commande Nucléaire Numérique pour l'EXport et la rénovatION - Coupler génie logiciel et ingénierie système : source d'innovations. *Génie Logiciel,* N° 104, 2-11.

Dobre, D. (2010). *Contribution à la modélisation d'un système interactif d'aide à la conduite d'un procédé industriel.* Unpublished PhD Thesis, Université Henri Poincaré, Nancy I, Nancy

Duco, J. (2004). Accidents nucléaires. Three Mile Island (Etats-Unis). *Techniques de l'ingénieur. Génie nucléaire* (BN3883), BN3883-3881.

Ducroq, A. (1996). Concept of semantial and phenomenon of emergence. *Proceedings of ECHO symposium.*

Dulmet, M., & Lhote, F. (2001). Analyse et caractérisation des couplages entre processus dans les systèmes de production. *Journal européen des systèmes automatisés,* 35 (10), 1193-1214.

Dupont, J., Lieber, R., Morel, G., Méry, D., & Bouffaron, F. (2012). Spécification d'un Processus Technico-Physiologique de Perception de Fermeture et Verrouillage d'un capot moteur en situation de maintenance aéronautique. In. Airbus - CRAN - LORIA.

Endsley, M. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society,* 37 (1), 32-64.

Esteban, P., Pascal, J., & Esteve, D. (2009). Une méthodologie de Conception Produit basée sur la norme EIA-632. *Congrès de Génie Industriel,(CIGI'2009)*.

Estefan, J. (2007). Survey of model-based systems engineering (MBSE) methodologies. *Incose MBSE Focus Group,* 25, 8.

Bibliographie

Faisandier, A. (2014). *Notions de système et d'ingénierie de système* (Vol. 1). Sinergy'Com.

Feliot, C. (1997). *Modélisation de systèmes complexes: intégration et formalisation de modèles.*

Fiorèse, S., & Meinadier, J. (2012). *Découvrir et comprendre l'Ingénierie Système* Cépaduès.

Fliess, M., Abouaissa, H., Iordanova, V., & Join, C. (2012). La commande<< sans modèle>> du trafic autoroutier. *FLASH X--La revue scientifique de l'Ecole polytechnique* (14), 25-28.

Fliess, M., & Join, C. (2013). Model-free control. *International Journal of Control,* 86 (12), 2228-2252.

Forsberg, K., & Mooz, H. (1995). Application of the "Vee" to incremental and evolutionary development. *Systems Engineering in the Global Market Place*, 801-808.

Frachet, J. (1987). *Une introduction au génie automatique: faisabilité d'une chaîne d'outlis CAO pour la conception et l'exploitation des machines automatiques industrielles.* Université Henri Poincaré, Nancy I, France.

Friedenthal, S., Moore, A., & Steiner, R. (2008). *A Practical Guide to SysML: Systems Model Language*. Morgan Kaufmann.

Fritzson, P., Aronsson, P., Bunus, P., Engelson, V., Saldamli, L., Johansson, H., & Karstöm, A. (2002). The open source Modelica project. In: *The 2th International Modelica Conference* (pp. 18-19), pp. 18-19.

Fusaoka, A., Seki, H., & Takahashi, K. (1983). A description and reasoning of plant controllers in temporal logic. In: *International Joint Conference on Artificial Intelligence* (Vol. 8-12/09, pp. 405-408), Karlsruhe, pp. 405-408.

Gaaloul, S. (2012). *Interopérabilité sur les standards Modelica et composant logiciel pour la simulation énergétique des sytèmes de bâtiment.* Université de Grenoble.

Galara, D. (1986). Elaboration d'informations crédibles pour les systèmes de contrôle-commande industriels : les modules fonctionnels d'automatismes. In. EDF/DER, R. I. Chatou.

Galara, D. (2006). Roadmap to master the complexity of process operation to help operators improve safety, productivity and reduce environmental impact. *Annual Reviews in Control,* 30 (2), 215-222.

Galara, D. (2011). Vers un langage de représentation des schémas mentaux des exploitants de systèmes de production complexes et à risques. *Génie Logiciel,* 96.

Galara, D., & Hennebicq, J. (1999). Process control engineering trends. *Annual Reviews in Control,* 23, 1-11.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: elements of reusable object-oriented software*. Pearson Education.

Garcia, J. (2009). Executable and Integrative Whole-System Modeling via the Application of OpEMCSS and Holons for Model-based Systems Engineering. *INSIGHT,* 12 (4), 21-23.

Garro, A., & Tundis, A. (2014). RAMSAS4Modelica: a Simulation-driven Method for System Dependability Analysis centered on the Modelica language and related tools. In: *Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative* (pp. 29). Society for Computer Simulation International, pp. 29.

Gazzaniga, M., Ivry, R., & Mangun, G. (2000). *Neurosciences cognitives: la biologie de l'esprit*. De Boeck Supérieur.

Gibson, J. (1977). The theory of affordances. *Hilldale, USA*.

Gibson, J. (2014). *The Ecological Approach to Visual Perception: Classic Edition*. Psychology Press.

Goldstein, E. (2009). *Encyclopedia of perception*. Sage Publications.

Gotel, O., & Finkelstein, C. (1994). An analysis of the requirements traceability problem. In: (pp. 94-101), pp. 94-101.

Gouyon, D., Bouffaron, F., & Morel, G. (2014). Specifying Some Key SE Training Artifacts. In: *Complex Systems Design & Management* (pp. 207-218). Springer, pp. 207-218.

Gunter, C., Gunter, E., Jackson, M., & Zave, P. (2000). A reference model for requirements and specifications. *Software, IEEE,* 17 (3), 37-43.

Halbwachs, N., Caspi, P., Raymond, P., & Pilaud, D. (1991). The synchronous data flow programming language LUSTRE. *Proceedings of the IEEE,* 79 (9), 1305-1320.

Hall, J., Jackson, M., Laney, R., Nuseibeh, B., & Rapanotti, L. (2002). Relating software requirements and architectures using problem frames. In: *the IEEE Joint International Conference on Requirements Engineering* (pp. 137-144), Essen, Germany, pp. 137-144.

Hall, J., & Rapanotti, L. (2005). Problem frames for sociotechnical systems. *Requirements Engineering for Sociotechnical Systems*, 318-339.

Hamdan, M., & Gao, Z. (2000). A novel PID controller for pneumatic proportional valves with hysteresis. In: *Industry Applications Conference, 2000. Conference Record of the 2000 IEEE* (Vol. 2, pp. 1198-1201). IEEE, pp. 1198-1201.

Hartson, R. (2003). Cognitive, physical, sensory, and functional affordances in interaction design. *Behaviour & Information Technology,* 22 (5), 315-338.

Herzog, E. (2004). *An approach to Systems Engineering tool data representation and exchange.* Unpublished PhD thesis, Linköping University.

Hollnagel, E. (2006). Resilience: the challenge of the unstable. *Resilience Enginnering-Concepts and Precepts*, 9-17.

Hollnagel, E., & Woods, D. (2005). *Joint cognitive systems: Foundations of cognitive systems engineering*. CRC Press.

Hopcroft, J. (1979). *Introduction to automata theory, languages, and computation*. Pearson Education India.

Houmb, S., Islam, S., Knauss, E., J\ürjens, J., & Schneider, K. (2010). Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec. *Requirements engineering,* 15 (1), 63-93.

IEEE 1220. (2005). IEEE Standard for Application and Management of the Systems Engineering Process. In. IEEE Computer Society.

INCOSE. (2007). Systems Engineering Vision 2020. In. INCOSE.

INCOSE. (2010). *Systems Engineering Handbook : a guide for system life cycle processes and activities (v 3.2)* (Vol. 3.2).

Bibliographie

ISO 10628. (2014). Schémas de procédé pour l'industrie chimique et pétrochimique - Partie 1 : Spécification des schémas du procédé. In. ISO (International Standardization Orgnaism).

ISO/IEC 15288. (2008). *ISO/IEC 15288 : Systems and software enginerring - System life cycle processes*. International Organisation for Standardization.

Iung, B. (1992). *Contribution à une intelligence distribuée dans les equipements de niveau zero des processus industriels complexes.*

Jackson, M. (1995). *Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices* (Vol. 8). ACM Press Wokingham.

Jackson, M. (1997). The meaning of requirements. *Annals of Software Engineering,* 3 (1), 5-21.

Jackson, M. (2001). *Problem frames: analysing and structuring software development problems*. Addison-Wesley.

Jackson, M., & Zave, P. (1995). Deriving specifications from requirements: an example.

Jacquemin, D. (2013). Les acccidents de fusion du coeur des réacteurs nucléaires de puissance. In: *Collection sciences et techniques*. IRSN.

Javaux, D., & De Keyser, V. (1997). Complexity and its certification in aeronautics. In: *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on* (Vol. 3, pp. 2120-2124). IEEE, pp. 2120-2124.

Jin, Z. (2006). Revisiting the meaning of requirements. *Journal of computer science and technology,* 21 (1), 32-40.

Johnson, T., Kerzhner, A., Paredis, C., & Burkhart, R. (2012). Integrating models and simulations of continuous dynamics into SysML. *Journal of Computing and Information Science in Engineering,* 12 (1), 011002.

Join, C. (2012). *Une approche algébrique pour la pratique de l'estimation, du diagnostic, de la commande et de la finance.* Université de Lorraine.

Join, C., Fliess, M. (2014). Développements industriels autour du concept de commande des systèmes sans modèle à priori. *Présentation Forum Académie/Entreprise de la Fédération Charles Hermite*

Join, C., Masse, J., & Fliess, M. (2008). Etude préliminaire d'une commande sans modèle pour papillon de moteur..... A model-free control for an engine throttle: a preliminary study. *Journal européen des systèmes automatisés,* 42 (2-3), 337--354.

Join, C., Robert, G., & Fliess, M. (2010). Model-free based water level control for hydroelectric power plants. In: *IFAC conference on control methodologies and tecnologies for energy efficiency, CMTEE*.

Kerkar, N., & Paulin, P. (2012). *Exploitation des coeurs REP*. EDP Sciences.

Kintzig, C., Poulain, G., Privat, G., & Favennec, P. (2002). *Objets communicants*. Hermes Science Publications.

Kleiner, S., & Kramer, C. (2013). Model based design with systems engineering based on RFLP using V6. In: *Smart Product Engineering* (pp. 93-102). Springer, pp. 93-102.

Koestler, A. (1967). The ghost in the machine. 1967. *London: Hutchinson*.

Krob, D. (2009). Eléments d'architecture des systèmes complexes. *Gestion de la complexité et de l'information dans les grands systèmes critiques*, 179-207.

Krob, D. (2012). Eléments de systémique-Architecture de systémes. *Complexité-Simplexité, Editions Odile Jacob*.

Kroes, P., Franssen, M., Poel, I., & Ottens, M. (2006). Treating socio-technical systems as engineering systems: some conceptual problems. *Systems research and behavioral science,* 23 (6), 803-814.

Kuras, M. (2006). A Multi Scale Definition of a System. *MT Report, ed.(Bedford, MA: The MITRE Corp.), Report MTR 06B0000060, August, www. mitre. org*.

Lamboley, P. (2001). *Proposition d'une méthode formelle d'automatisation de systèmes de production à l'aide de la méthode B.* Unpublished PhD thesis, Nancy University.

Lawson, H. (2010). *A journey through the systems landscape*. College Publications.

Lawson, H., Wade, J., & Hofkirchner, W. (2005-2014). Systems Science, Systems Thinking, Systems Engineering and Software Engineering. In: *Systems series published books, http://www.collegepublications.co.uk/systems/syt/*.

Le Gallou, F., & Bouchon-Meunier, B. (1992). *Systémique : théorie et applications*. Lavoisier-Tec & Doc.

Le Moigne, J. (1990). *La modélisation des systèmes complexes* (Vol. 2). Dunod Paris.

Le Moigne, J. (1994). *La théorie du système général: théorie de la modélisation*. jeanlouis le moigne-ae mcx.

Le Put, A., & Meinadier, J. (2009). Référentiel Ingénierie Système AFIS. Paper presented at the 5 ème conférence annuelle d'ingénierie système, Paris.

Leger, J. (1999). *Contribution méthodologique à la maintenance prévisionnelle des systèmes industriels de production: proposition d'un cadre formel de modélisation.*

Lhoste, P. (1985). *Exploitation des systèmes automatisés, EX. A. O. : proposition d'une approche méthodologique et d'outils d'assistance.* Henri Poincaré.

Lhoste, P. (1994). Contribution au Génie Automatique : Concepts, Modèles, Méthodes et Outils. In.

Lieber, R. (2013). *Spécification d'exigences physico-physiologiques en ingénierie d'un système support de maintenance aéronautique.* Université de Lorraine.

Lieber, R., Dupont, J., Bouffaron, F., & Morel, G. (2013). Improving physical-physiological interaction requirements for maintenance enabling systems specification. In: *12th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*.

Liu, Y. (2002). *Contribution à l'automatisation des systèmes intégrés et à l'intelligence distribuée de Contrôle, Maintenance et Gestion Technique (ICMMS) pour les centrales hydroélectriques.*, Université Henri Poincaré (Nancy) - Université de HUST (Pekin).

Lo, M. (2013). *Contribution à l'évaluation d'architectures en Ingénierie Système: application en conception de systèmes mécatroniques.* Université Montpellier II-Sciences et Techniques du Languedoc.

Mankins, J. (1995). Technology readiness levels. *White Paper, April,* 6.

Mateják, M., Kulhánek, T., Šilar, J., Privitzer, P., Ježek, F., & Kofránek, J. (2014). Physiolibrary-modelica library for physiology. In: *10th International Modelica Conference* (pp. 499-505). Linköping University Electronic Press Lund, Sweden, pp. 499-505.

Mayer, F. (1995). *Contribution au Génie Productique: application à l'Ingénierie Pédagogique en Atelier Inter-établissements de Productique Lorrain.*

Medjaher, K. (2005). *Contribution de l'outil bond graph pour la conception de systèmes de supervision des processus industriels.* Lille 1.

Michel, L., Join, C., Fliess, M., Sicard, P., & Chériti, A. (2010). Model-free control of dc/dc converters. In: *Control and Modeling for Power Electronics (COMPEL), 2010 IEEE 12th Workshop on* (pp. 1-8). IEEE, pp. 1-8.

Micouin, P. (2011). Exigences en ingénierie systèmes basée modèles. In: *9 ième Congrès International de Génie Industriel*, Saint Sauveur, Quebec, Canada.

Millot, P. (2014). *Designing Human-machine Cooperation Systems*. John Wiley & Sons.

Millot, P., Debernard, S., & Vanderhaegen, F. (2011). Authority and cooperation between humans and machines. *Handbook for Human-Machine Interaction. Ashgate Publishing Ltd, Wey Court East, Union Road, Farnham, Surrey, GU9 7PT, England*.

Modelica. (2008). The Modelica Standard Library. *Online, URL: http://www. modelica. org/libraries/Modelica*.

Modelica. (2012). Modelica - A Unified Object-Oriented Language for Physical Systems Modeling: Language Specification Version 3.3. In.

Morel, G. (1999). Contribution à la simulation des parties opératives interopérables pour la valiation d'automatismes distribués. In. Université Henri Poincaré.

Morel, G., Bouffaron, F., Nartz, O., Mayer, F., Marangé, P., Levrat, E., & Panetto, H. (2014). Vers un apprentissage itératif à l'ingénierie système basée sur des modèles. In: *10ème Conférence Francophone de Modélisation, Optimisation et Simulation, MOSIM'14*.

Morel, G., Dupont, J., Lieber, R., Bouffaron, F., Mery, D., Mayer, F., & Marty, J. (2013). Spécification d'exigences physico-physiologiques d'interaction homme-machine en ingénierie système. *Génie Logiciel* (104), 29-39.

Morel, G., & Grabot, B. (2003). Editorial of special issue. *Engineering applications of artificial intelligence,* 16 (4), 271-275.

Morel, G., Panetto, H., Zaremba, M., & Mayer, F. (2003). Manufacturing enterprise control and management system engineering: paradigms and open issues. *Annual Reviews in Control,* 27 (2), 199-209.

Morel, G., Valckenaers, P., Faure, J., Pereira, C., & Diedrich, C. (2007). Manufacturing plant control challenges and issues. *Control Engineering Practice,* 15 (11), 1321-1331.

Morel, J. (1992). *Surveillance vibratoire et maintenance prédictive*. Ed. Techniques Ingénieur.

Morin, E. (1980). *La méthode, tome 2. La Vie de la Vie* (Vol. 4).

Müller, G., & Möser, M. (2012). *Handbook of engineering acoustics*. Springer Science & Business Media.

NASA. (2007). *NASA Systems Engineering Handbook (NASA/Sp-2007-6105 Rev1)*. US National Aeronautics & Space Administration.

Nasri, M., Nezamabadi-Pour, H., & Maghfoori, M. (2007). A PSO-based optimum design of PID controller for a linear brushless DC motor. *World Academy of Science, Engineering and Technology,* 26 (40), 211-215.

Neunreuther, E. (1998). *Contribution à la modélisation des Systémes Intégrés de Production à Intelligence Distribuée: application à la distribution du contrôle et de la gestion technique sur les équipements de terrain.*

Neyns, V. (2011). *Les modalités du contrôle cognitif en situation dynamique: anticipation et gestion des dérives. Le cas de l'anesthésie.* Université Toulouse le Mirail-Toulouse II.

NF EN 60848. (2002). Langage de spécification GRAFCET pour diagrammes fonctionnels en séquence. In.

NF S32 001. (1975). Signal sonore d'évacuation d'urgence. In. AFNOR.

Norman, D. (1986). Cognitive engineering. *User centered system design: New perspectives on human-computer interaction,* 3161.

Nugent, P., & Collar, E. (2015). The Hidden Perils of Addressing Complexity with Formal Process - A Philosophical and Empirical Analysis. In: Boulanger, F., Krob, D., Morel, G. & Roussel, J.-C. (Eds.), *Complex Systems Design & Management* (pp. 119-131). Springer International Publishing, pp. 119-131.

OMG. *Systems Modeling Language (OMG SysML™), V1. 0*.

OMG. (2010). *OMG Systems Modeling Language (OMG SysML) (v 1.2)*.

OMG. (2012a). *OMG Systems Modeling Language (OMG SysML) (v 1.3)*.

OMG. (2012b). SysML-Modelica transformation specification - v1.0. In. OMG.

Otter, M., Årzén, K., & Dressler, I. (2005). StateGraph–a Modelica library for hierarchical state machines. In: *the 4th international Modelica conference* (pp. 569-578), pp. 569-578.

Palaci, F. (2009). Les agents de terrain du nucléaire : comprendre leur activité pour contribuer à la conception d'un outil d'aide informatique. In: *Mémoire de Master Professionnel d'ergonomie*. Conservatoire National des Arts et Métiers Paris.

Panetto, H. (1991). *Une contribution au génie automatique: le prototypage des machines et systèmes automatisés de production.* Université Henri Poincaré-Nancy I.

Paynter, H. (1961). *Analysis and design of engineering systems*. MIT press.

Penalva, J. (1997). *La modélisation par les systèmes en situation complexes.*

Peterson, J. (1981). Petri net theory and the modeling of systems.

Pétin, J. (1995). *Contribution méthodologique à l'Actionnement et la Mesure Intelligents: Application au projet ESPRIT III-PRIAM nR 6188.* Université Henri Poincaré, Nancy I, Nancy.

Pétin, J. (2007). *Méthodes et modèles pour un processus sûr d'automatisation.* Université Henri Poincaré-Nancy I.

Pétin, J., Iung, B., & Morel, G. (1998). Distributed intelligent actuation and measurement (IAM) system within an integrated shop-floor organisation. *Computers in Industry,* 37 (3), 197-211.

Pétin, J., Morel, G., & Panetto, H. (2006). Formal specification method for systems automation. *European journal of control,* 12 (2), 115-130.

Pouvreau, D. (2013). *Une histoire de la" systémologie générale" de Ludwig von Bertalanffy-Généalogie, genèse, actualisation et postérité d'un projet herméneutique.* Ecole des Hautes Etudes en Sciences Sociales (EHESS).

Prouillac, M., Lerat, & Janior. (1981). Prise en compte de la fiabilité humaine dans la conception des centrales nucléaires. *Revue générale de l'électricité,,* Numéro 5 - La fiabilité humaine, 434-440.

PTA. (1987). PTA (Poste de Travail Pour l'Automatisation) - Synthèse MRES, RNUR, MICHELIN, ADEPA, PSA, SGN-ISMCM, LACN, LAG, LAMM.

Pyster, A., Olwell, D., Hutchison, N., Enck, S., Anthony, D., Squires, H., & Squires, A. (2012). Guide to the Systems Engineering Body of Knowledge (SEBoK). Version 1.0.1. *Hoboken, NJ: The Trustees of the Stevens Institute of Technology ©2012. Available at: http://www.sebokwiki.org/index.php/article_title*.

Qamar, A., During, C., & Wikander, J. (2009). Designing mechatronic systems, a model-based perspective, an attempt to achieve sysml-matlab/simulink model integration. In: *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on* (pp. 1306-1311). IEEE, pp. 1306-1311.

Rahman, M., Nor, N., & Mizukawa, M. (2012). Evaluation for SysML-based design and analysis models using PCE. In: *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on* (pp. 339-344). IEEE, pp. 339-344.

Ramos, A., Ferreira, J., & Barceló, J. (2012). Model-based systems engineering: An emerging approach for modern systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* 42 (1), 101-111.

Ranky, P. (1986). *Computer integrated manufacturing*. Prentice-Hall, Inc.

Rao, M., & Wang, Q. (1993). Computer integrated process systems in continuous manufacturing industries. *Computer Integrated Manufacturing Systems,* 6 (4), 260-272.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction. An Approach to Cognitive Engineering* (Vol. 12).

Renier, R., & Chenouard, R. (2011). De SysML à Modelica aide à la formalisation de modèles de simulation en conception préliminaire. In: *12ème Colloque National AIP PRIMECA*.

Retho, F. (2015). *Méthodologie collaborative d'aide à la construction de produits virtuels pour la conception d'aéronefs à propulsion électrique.* Supelec.

Richard, J. (1990). *Les activités mentales: comprendre, raisonner, trouver des solutions*. A. Colin.

Rochet, S. (2007). Formalisation des Processus de l'Ingénierie Système: Proposition d'une méthode d'adaptation des processus génériques à différents contextes d'application.

Roques, P. (2011). *SysML par l'exemple-Un langage de modélisation pour systèmes complexes*. Editions Eyrolles.

Royce, W. (1970). Managing the development of large software systems. In: *IEEE WESCON* (Vol. 26, pp. 328-388). Los Angeles, pp. 328-388.

Salmon, P., Stanton, N., Walker, G., Baber, C., Jenkins, D., McMaster, R., & Young, M. (2008). What really is going on? Review of situation awareness models for individuals and teams. *Theoretical Issues in Ergonomics Science,* 9 (4), 297-323.

Salmon, P., Stanton, N., Walker, G., & Jenkins, D. (2009). Distributed situation awareness. *Theory Measurement and Application to Teamwork. Farnham: Ashgate*.

Sannier, N., & Baudry, B. (2011). Défis pour la variabilité et la traçabilité des exigences en ingénierie système. In: *INFORSID 2011*.

Sauser, B., Ramirez-Marquez, J., Verma, D., & Gove, R. (2006). Determining system interoperability using an integration readiness level. *Stevens Institute of Technology,* 21.

Schmidt, R. (1999). *Physiologie*. De Boeck Supérieur.

Simão, J., & Stadzisz, P. (2009). Inference based on notifications: a holonic metamodel applied to control issues. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* 39 (1), 238-250.

Sindico, A., Di Natale, M., & Panci, G. (2011). Integrating SysML with Simulink using Open-source Model Transformations. In: *SIMULTECH* (pp. 45-56), pp. 45-56.

Smith, G., & Daniel-Allegro, B. (2015). Mind mapping systems thinker's attitude facing a sepsis problem. Paper presented at the ISSS2015 (9th Meeting of the ISSS) overning the Antropocene Berlin.

Staal, M. (2004). Stress, cognition, and human performance: A literature review and conceptual framework. *NaSA technical memorandum,* 212824, 9.

Suied, C. (2007). *De l'urgence perçue au temps de réaction: application aux alarmes sonores.* Paris 6.

Taguchi, G. (1987). System of experimental design; engineering methods to optimize quality and minimize costs. In.

Tarride, B. (2013). *Physique, fonctionnement et sûreté des REP: Maîtrise des situations accidentelles du système réacteur*. EDP sciences.

Tixador, J. (1989). *Une contribution au genie automatique: la specification executable des machines et systemes automatises de production.* Nancy 1.

Valckenaers, P., Van Brussel, H., Bongaerts, L., & Wyns, J. (1997). Holonic manufacturing systems. *Integrated Computer-Aided Engineering,* 4 (3), 191-201.

Van Lamsweerde, A. (2000). Formal specification: a roadmap. In: (pp. 147-159). ACM, pp. 147-159.

Vicente, K. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. CRC Press.

Vicente, K., Mumaw, R., & Roth, E. (2004). Operator monitoring in a complex dynamic work environment: A qualitative cognitive model based on field observations. *Theoretical Issues in Ergonomics Science,* 5 (5), 359-384.

Viel, M., Cheriaux, F., & Galara, D. (2012). Expérimentation de techniques d'ingénierie système pour la formalisation de données de fonctionnement et de contrôle-commande. *Génie Logiciel,* 100.

Von Bertalanffy, L. (1968). *General System Theory: Foundations, Development* (Revised edition (december 1968) ed.). Braziller.

Waldner, J.-B., & Duffin, W. (1992). *CIM, principles of computer-integrated manufacturing*. Wiley Chichester.

Weilkiens, T. (2007). *Systems engineering with SysML/UML: modeling, analysis, design*. Morgan Kaufmann.

Wetter, M. (2010). Modelica library for building heating, ventilation and air-conditioning systems.

Wilber, K. (1997). An integral theory of consciousness. *Journal of Consciousness Studies,* 4 (1), 71-92.

Wong, C., McFarlane, D., & Agarwal, V. (2002). The intelligent product driven supply chain. In: *Systems, Man and Cybernetics, 2002 IEEE International Conference On* (Vol. 4, pp. 6 pp. vol. 4). IEEE, pp. 6 pp. vol. 4.

Workgroup OSLC. (2010). OSLC core specification version 2.0. *Open Services for Lifecycle Collaboration, Tech. Rep*.

Yufei, S., & Kazuo, F. (2005). An inference method of team situation awareness based on mutual awareness. *Cogn. Technol. Work,* 7 (4), 272-287.

Zachman, J. (1987). A framework for information systems architecture. *IBM systems journal,* 26 (3), 276-292.

Zarifian, P. (1990). Hiérarchisation, Réseaux et Systèmes de décision dans les grandes entreprises industrielles. *Réseaux,* 8 (41), 93-103.

Annexes

Annexe 1-1 : Etats standards du palier 1300 MWe

Domaines d'exploitation	Etats standard	Inventaire en réfrigérant primaire	Pression (bar abs.)	Température moyenne (°C)	Concentration en bore (ppm)	Puissance neutronique
Réacteur Complétement Déchargé (RCD)	Tout combustible dans BK	-	-	-	-	-
Arrêt Pour Rechargement (APR)	Arrêt à froid pour rechargement	729 m^3 (P4) 657 m^3 (P'4) au-dessus du PJC	Atmos.	$10 \leq T \leq 60$	≥ 2385	0
Arrêt Pour Intervention (API)	Arrêt à froid pour intervention primaire suffisamment ouvert (Le trou d'homme pressuriseur est entièrement ouvert)	≥ NB PTB RRA	Atmos.	$10 \leq T \leq 60$	≥ 2385	0
	Arrêt à froid pour intervention primaire entrouvert	≥ NB PTB RRA	Atmos.	$10 \leq T \leq 60$	≥ 2385	0
	Arrêt à froid normal primaire fermé et dépressurisé	≥ NB PTB RRA	$P \leq 5$	$10 \leq T \leq 60$	≥ 2385	0
Arrêt Normal sur RRA (AN/RRA)	Arrêt à froid normal	Primaire Plein	$5 \leq P \leq 31$	$10 \leq T \leq 90$	≥ CB_{AF}	0
	Arrêt intermédiaire monophasique	Primaire Plein	$25 \leq P \leq 31$	$90 \leq T \leq 180$	≥ CB_{AF}	0
	Arrêt intermédiaire diphasique aux conditions RRA connecté	Primaire Plein Pressuriseur diphasique	$25 \leq P \leq 31$	$120 \leq T \leq 180$	≥ CB_{AF}	0
Arrêt Normal sur GV (AN/GV)	Arrêt intermédiaire diphasique aux conditions RRA isolé	Primaire Plein Pressuriseur diphasique	$27 \leq P \leq 31$	$160 \leq T \leq 180$	≥ CB_{AF}	0
	Arrêt intermédiaire diphasique sur GV	Primaire Plein Pressuriseur diphasique	$27 \leq P \leq 31$ Ou $160 \leq T \leq P_{12}$		≥ CB_{AF}	0
	Arrêt à chaud	Primaire Plein Pressuriseur diphasique	$P_{11} \leq P \leq 155$ Et $P_{12} \leq T \leq 297{,}2^{+3/-2}$		≥ CB_{AC}	0
Réacteur en Production (RP)	Recherche de la criticité	Primaire Plein Pressuriseur diphasique	≈ 155	$297{,}2^{+3/-2}$	Recherche CB critique	≈ 0
	Attente à chaud	Primaire Plein Pressuriseur diphasique	≈ 155	$297{,}2^{+3/-2}$	CB critique	≤ 2% Pn
	Puissance	Primaire Plein Pressuriseur diphasique	≈ 155	$297{,}2^{+3/-2}$	CB critique	2% Pn ≤ P ≤ 100 % Pn

Tableau 22 : Correspondance entre domaines d'exploitations et états standard du palier 1300MWe (Kerkar & Paulin, 2012)

Annexe 3-1 : Modélisation fonctionnelle selon le paradigme systémique

Selon l'interprétation du paradigme systémique faites par (Galara, 1995)(Figure 179) le modèle de compréhension du système comme un TOUT correspond au modèle fonctionnel.

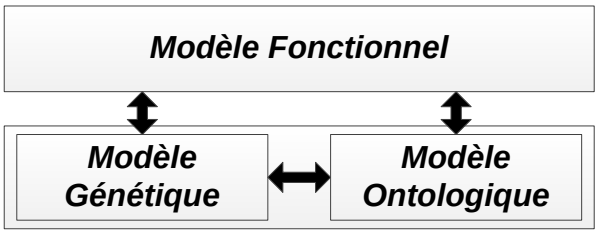


Figure 179 : Interprétation du paradigme systémique par (Galara, 1995)

- **Le modèle fonctionnel** d'un système est une description externe de tous ses comportements attendus dans son environnement, pour représenter son fonctionnement indépendant de toutes les contraintes d'implantation. Notons cependant, que cette modélisation fonctionnelle met en jeu tous les acteurs de modélisation qu'ils soient fonctionnel génétique ou ontologique. D'un point de vue de la co-spécification cela correspond pour nous à intégrer l'ensemble des ingénieries spécialistes impliquées dans le projet pour décrire le fonctionnent attendu du système d'intérêt.

- **Le modèle ontologique** d'un système est une description interne de tous ses comportements technologiques pour représenter son fonctionnement en tenant compte des contraintes de conception. Ce modèle est une description physique et normative en termes de matériels et de logiciels supportant la réalisation du système d'intérêt.

- **Le modèle génétique** d'un système est une description à la fois externe et interne de son fonctionnement distribué qui a pour objectif de garantir son évolution en reliant l'approche descendante de modélisation fonctionnelle indépendante de toute technologie avec l'approche ascendante de modélisation ontologique dépendante de la technologie existante. Le modèle génétique effectue l'interface entre le modèle fonctionnel et le modèle ontologique.

Annexe 4-1 : Spécification d'un AMI avec les outils PRIAM et SPEX

Logical Unit	Function Name	Meaning	Class	Algorithm
	Mon_cp	Monitoring the current position	Basic	

Interface description

Function mon_cp Interface Editor

File Help

AUTHOR : CRANVEACN_GSIP
PROJECT : EIAMUO
DATE : 30 Aug 94

READERS :

Label	Modified	Direction	Mode	CO	CN	NO	MN	SL	Mandatory	Service	Access
phy_sgn_pos		Consume	Normal	True	True	True	True	True	True	State Control	Data
act_state		Produced		True	True	True	True	True	True	State Control	Cyclic
curr_pos		Produced		True	True	True	True	True	True	State Control	Cyclic
al_p_state		Produced		False	False	False	False	False	False	State Control	Event

OK Cancel

Figure 180 : Spécification des interfaces d'une fonction AMI avec l'outil PRIAM

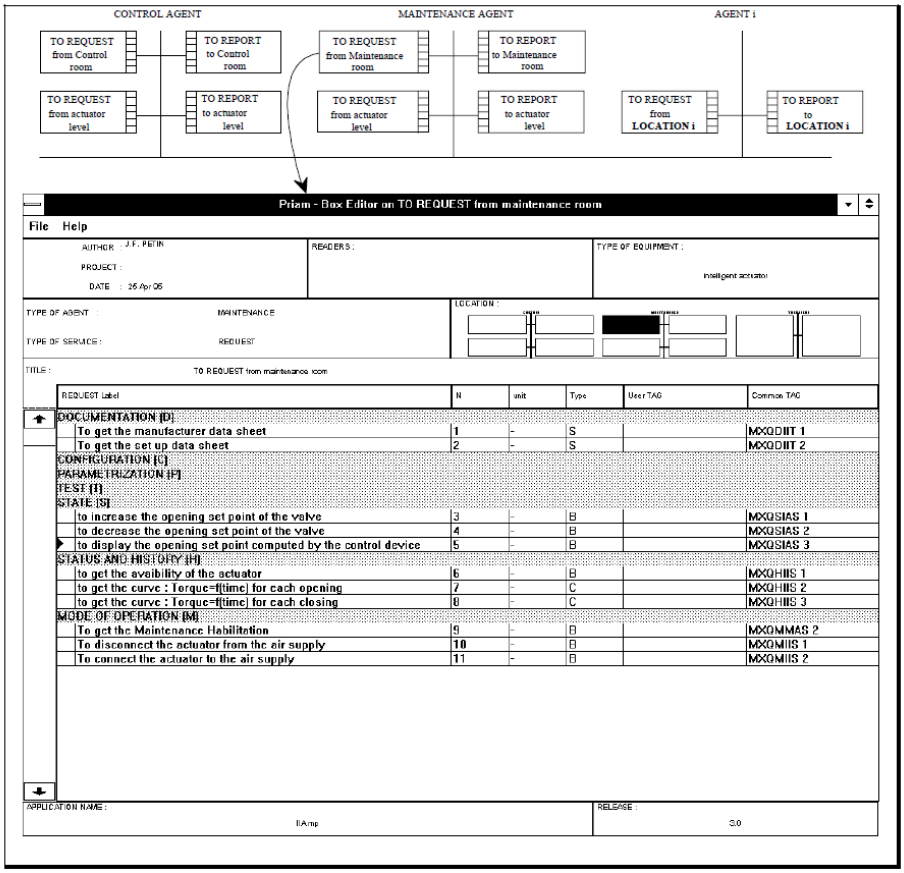


Figure 181 : Exemple de listes de requêtes et de comptes rendus (PRIAM)

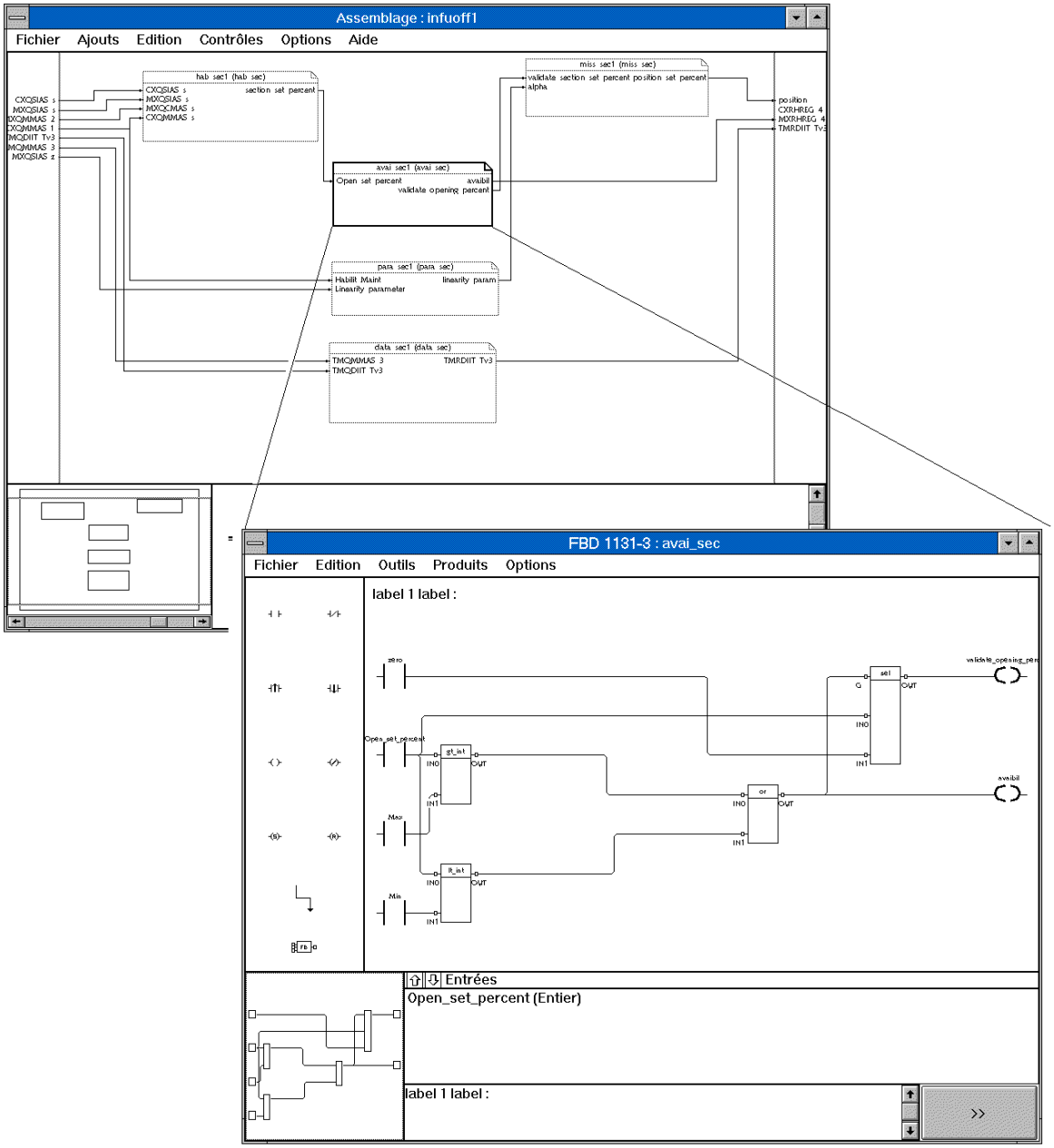


Figure 182 : Spécification comportementale d'un AMI en langage FBD (Bloc Fonctionnel) avec l'outil SPEX

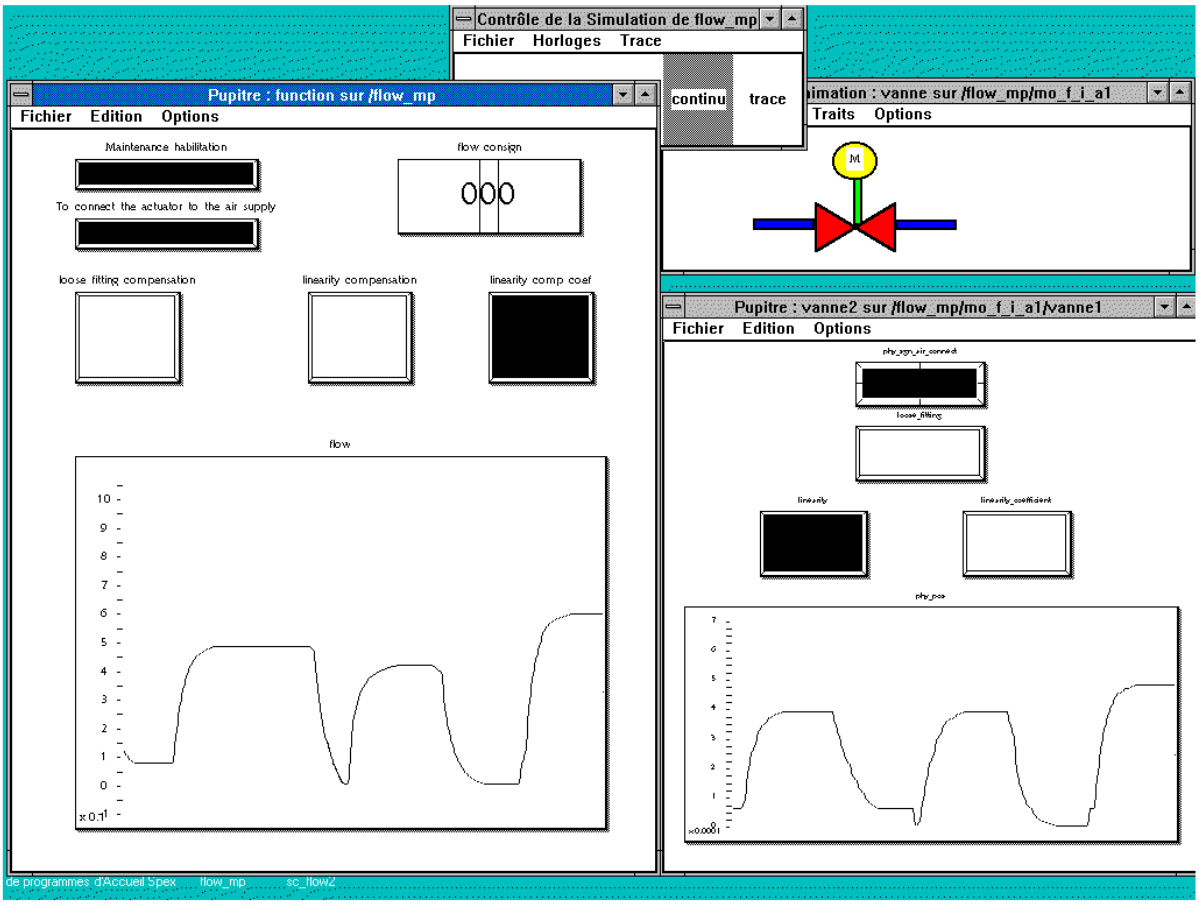


Figure 183 : Simulation du modèle comportemental réalisée sur l'outil SPEX

Annexe 4-2 : Structuration d'un espace collaboratif de co-spécification

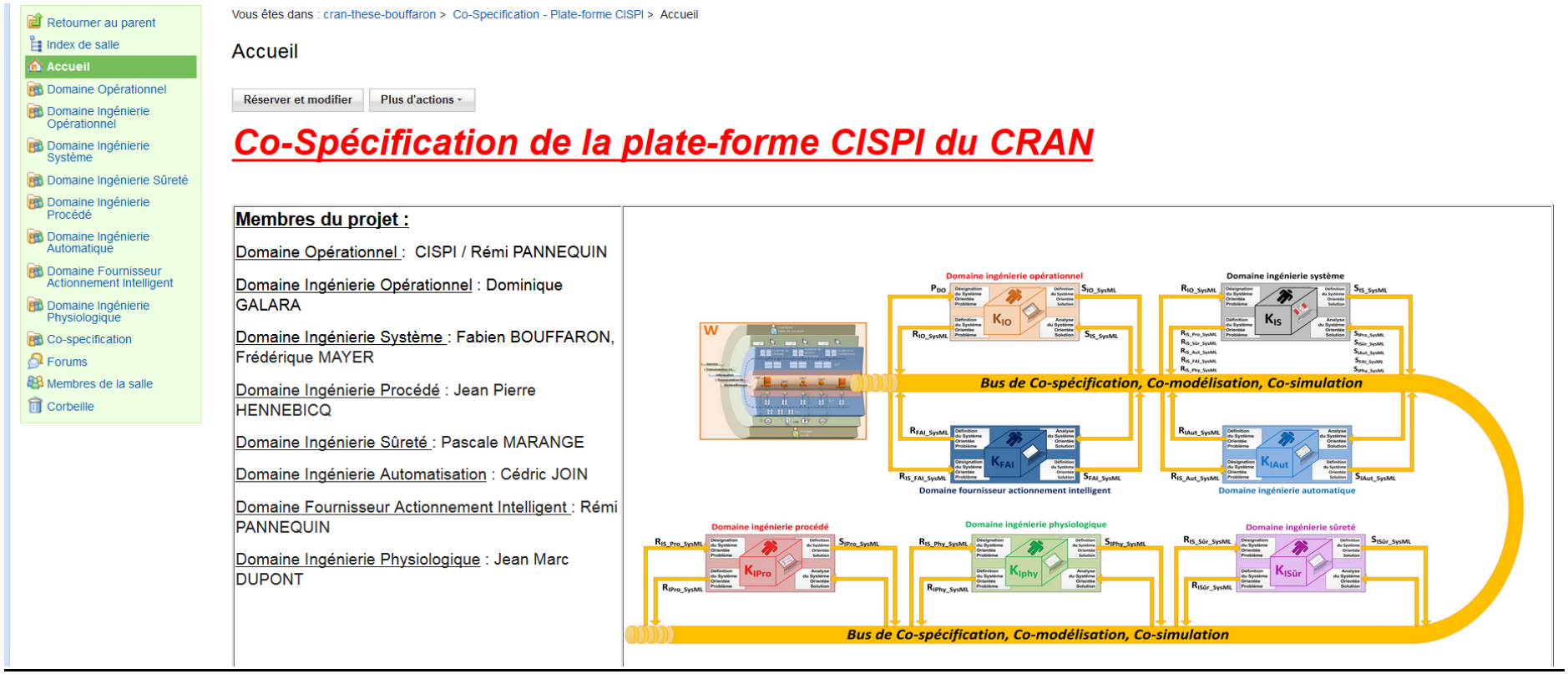


Figure 184 : Structuration de l'espace "Quickplace" de co-spécification de la plate-forme CISPI

Figure 185 : Structuration d'un domaine d'ingénierie en espaces problème et solution et connaissances

Lotus. Quickr | Mes espaces

Mes espaces

Co-specification

- Retourner au parent
- Accueil
- Index de salle
- Spécification - Domaine Opérationnelle - ES Ingénierie Opérationnel
- Spécification - ES Ingénierie Opérationnel - EP Ingénierie Opérationnel
- Spécification - EP Ingénierie Opérationnel - ES Ingénierie Système
- Spécification - ES Ingénierie Système - EP Ingénierie Système
- Spécification - EP Ingénierie Système - ES Ingénierie Procédé
- Spécification - EP Ingénierie Système - ES Ingénierie Sûreté
- Spécification - EP Ingénierie Système - ES Ingénierie Automatique
- Spécification - EP Ingénierie Système - ES Fournisseur Actionnement Intelligent
- Spécification - EP Ingénierie Système - ES Ingénierie Physiologique
- Forums
- Membres de la salle
- Corbeille

Vous êtes dans : cran-these-bouffaron > ... > [Salle parente] > Co-specification > Spécification - EP Ingénierie Système - ES Ingénierie Procédé

Spécification - EP Ingénierie Système - ES Ingénierie Procédé

Créé le aujourd'hui à 00:31 par bouffaro5 | Dernière modification le aujourd'hui à 00:34 par bouffaro5 | Public

Télécharger | Nouveau | Plus d'actions

1-2 sur 2 — Page 1

Type	Nom
	Spécification Descriptive (Problème)
	Spécification Prescriptive (Solution)

Afficher : 10 | 25 | 50 | 100 éléments — Atteindre la page 1 sur 1

S'abonner à ce dossier

Figure 186 : Structuration de la salle "Co-Spécification" en dossier de spécification entre espaces problème et solution

Annexe 5-1 : Définition des exigences des parties prenantes intéressées (R_{IO_SysML}) – (Espace-problème d'ingénierie opérationnelle)

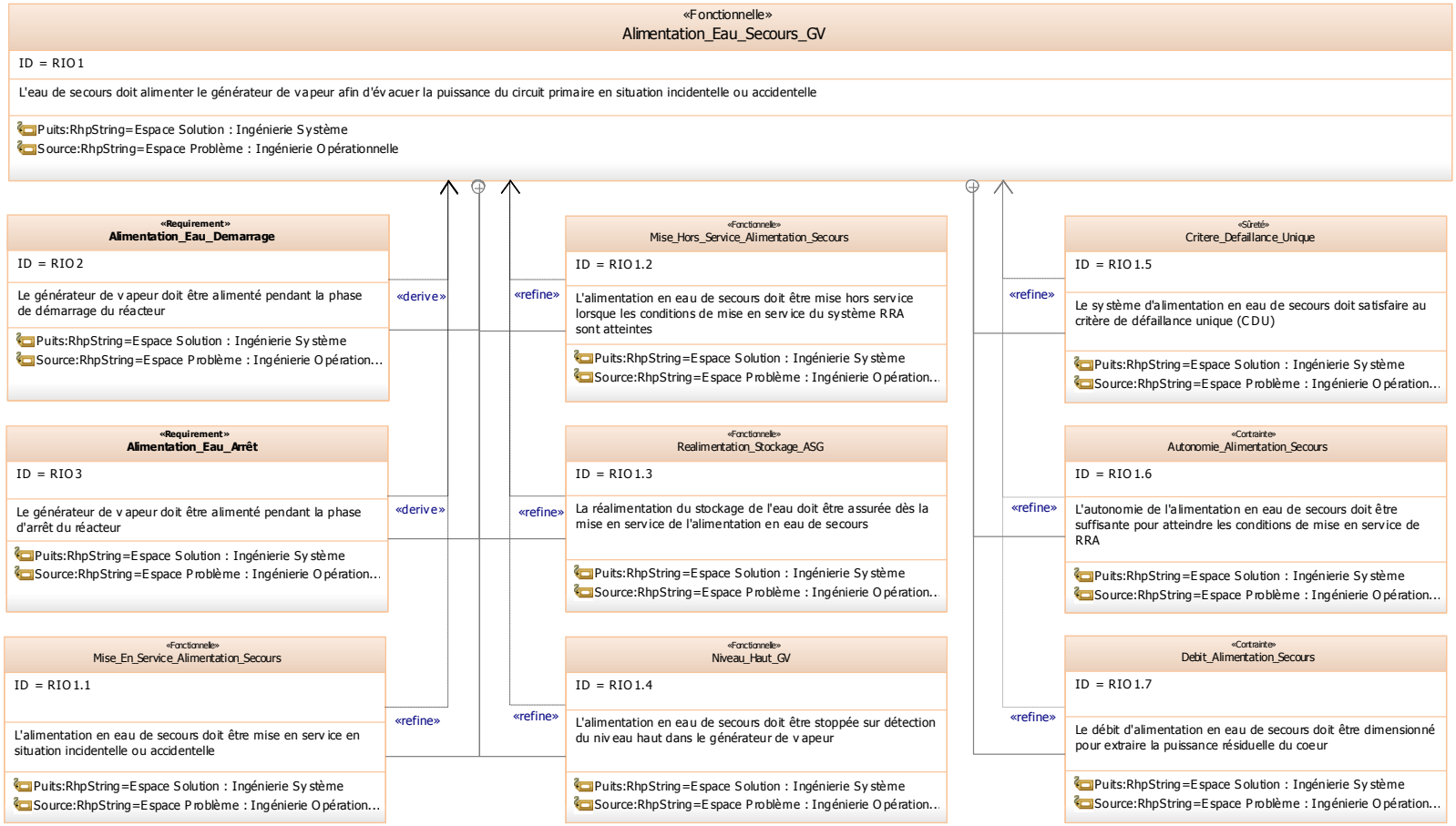


Figure 187 : Diagramme d'exigence SysML - Exigences décrites par l'*espace problème* d'ingénierie opérationnelle (R_{IO_SysML})

Annexe 5-2 : Analyse des exigences parties prenantes (R_{IO_SysML}) – (Espace-solution d'ingénierie système)

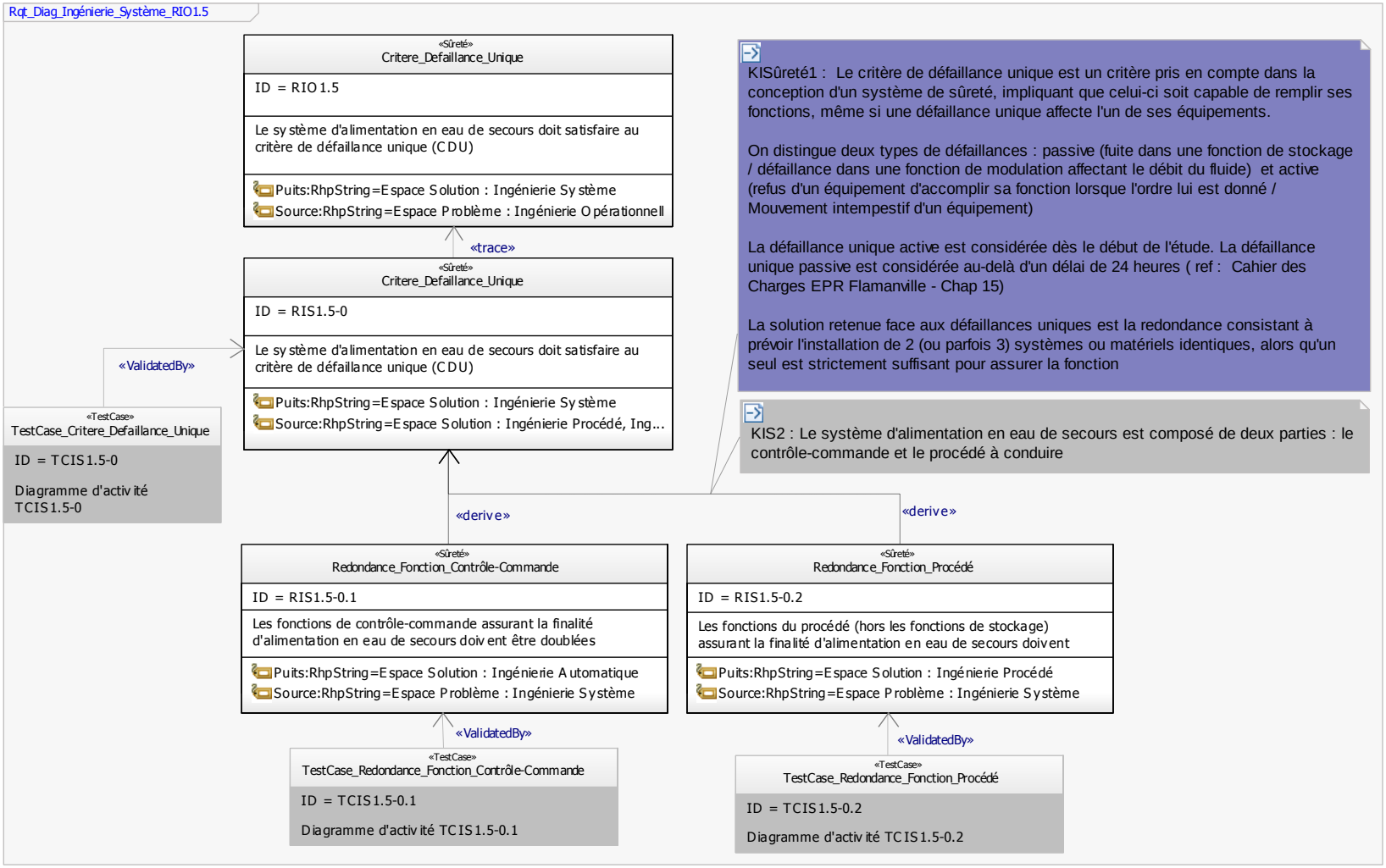


Figure 188 : Diagramme d'exigences SysML - Raffinement de l'exigence opérationnelle (*RIO 1.5*) par l'*espace-solution* d'ingénierie système

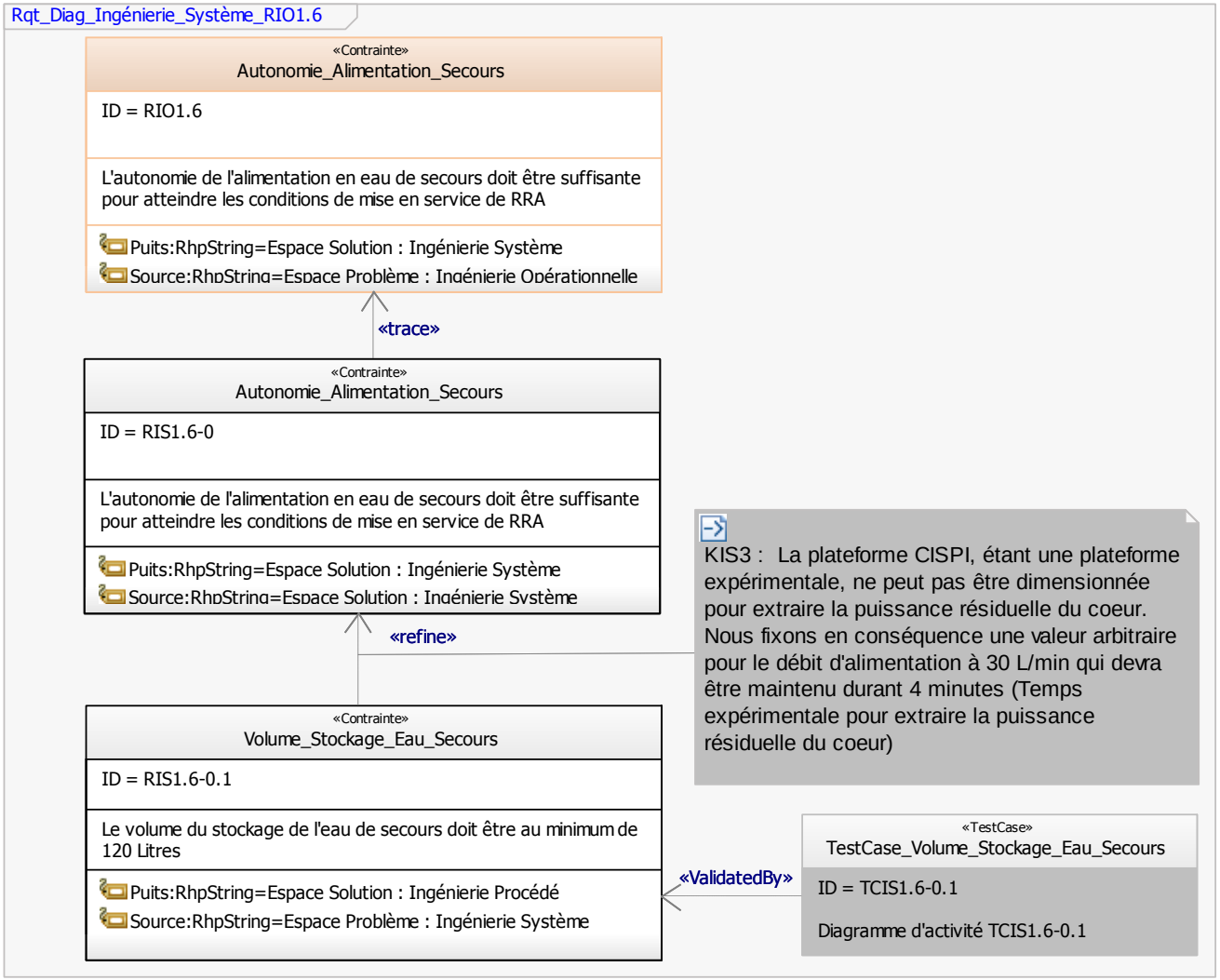


Figure 189 : Diagramme d'exigences SysML - Raffinement de l'exigence opérationnelle (*RIO 1.6*) par l'*espace-solution* d'ingénierie système

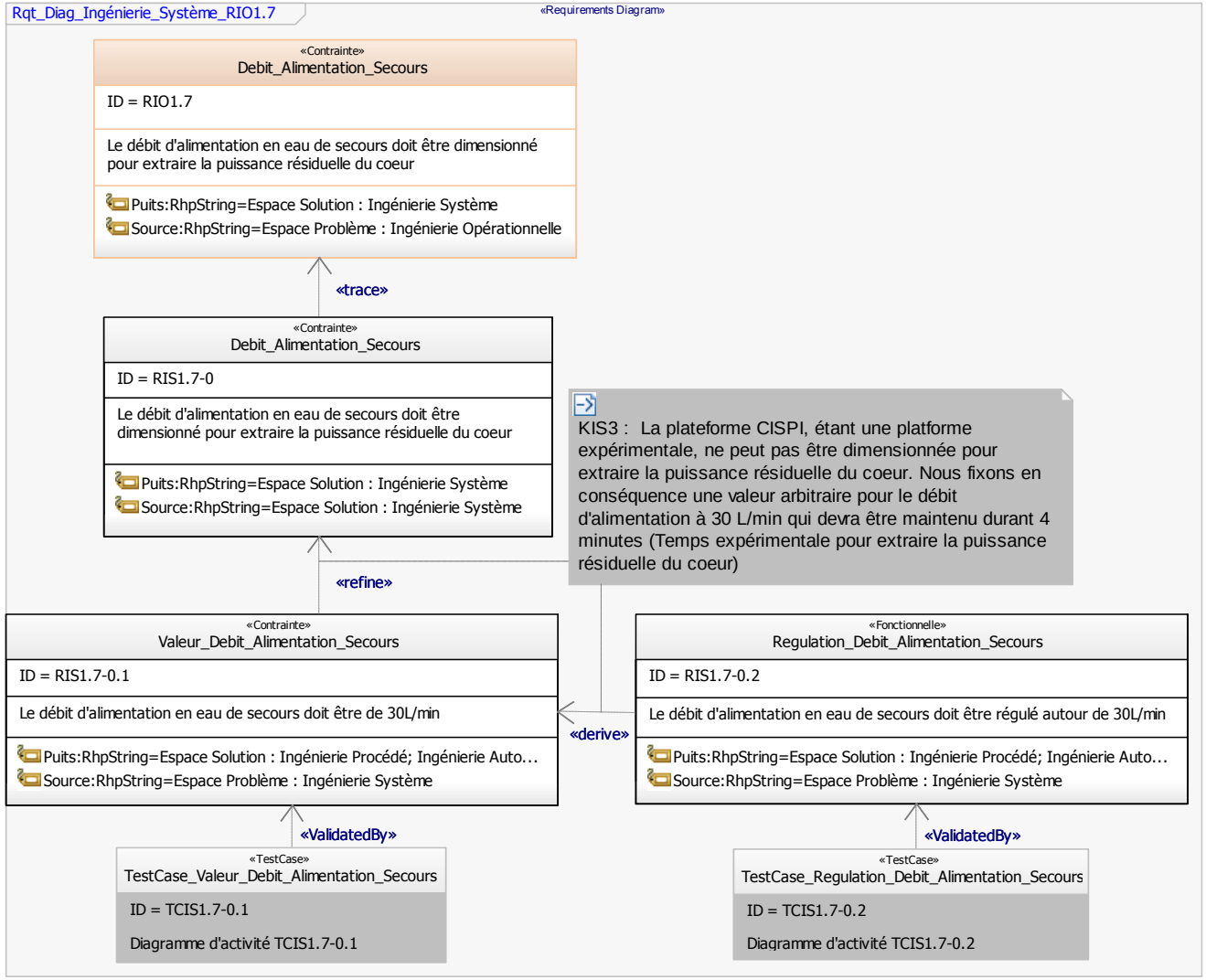


Figure 190 : Diagramme d'exigences SysML - Raffinement de l'exigence opérationnelle (*RIO 1.7*) par l'*espace-solution* d'ingénierie système

Annexe 5-3 : Exigences et problèmes systèmes ($R_{IS_Pro_SysML}$) décrits à l'espace-solution d'ingénierie du procédé

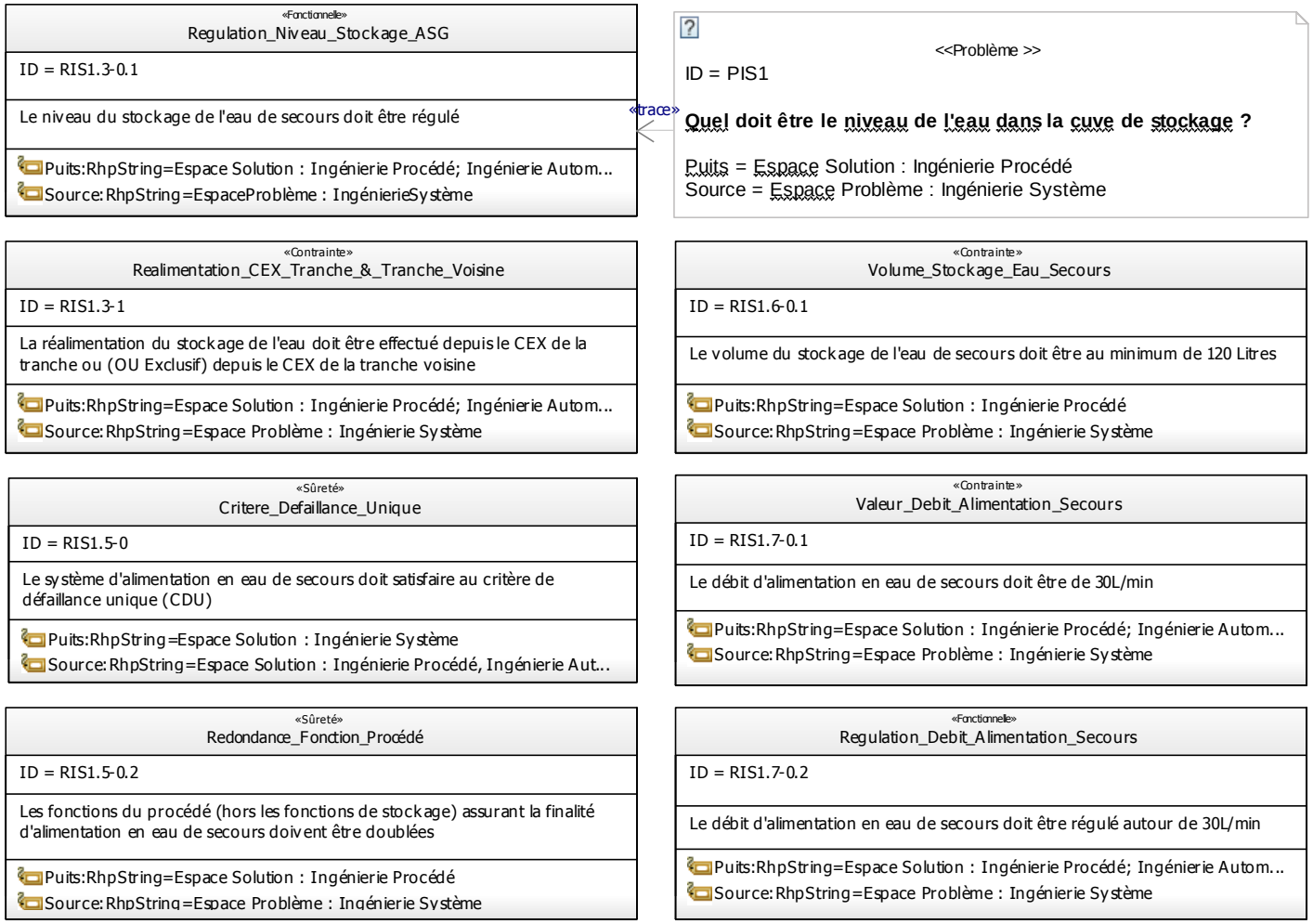


Figure 191 : Diagramme d'exigences SysML - Exigences et problèmes systèmes ($R_{IS_Pro_SysML}$) relatives au procédé

Annexe 5-4 : Modèle de l'interaction$_{\text{Procédé-Environnement}}$

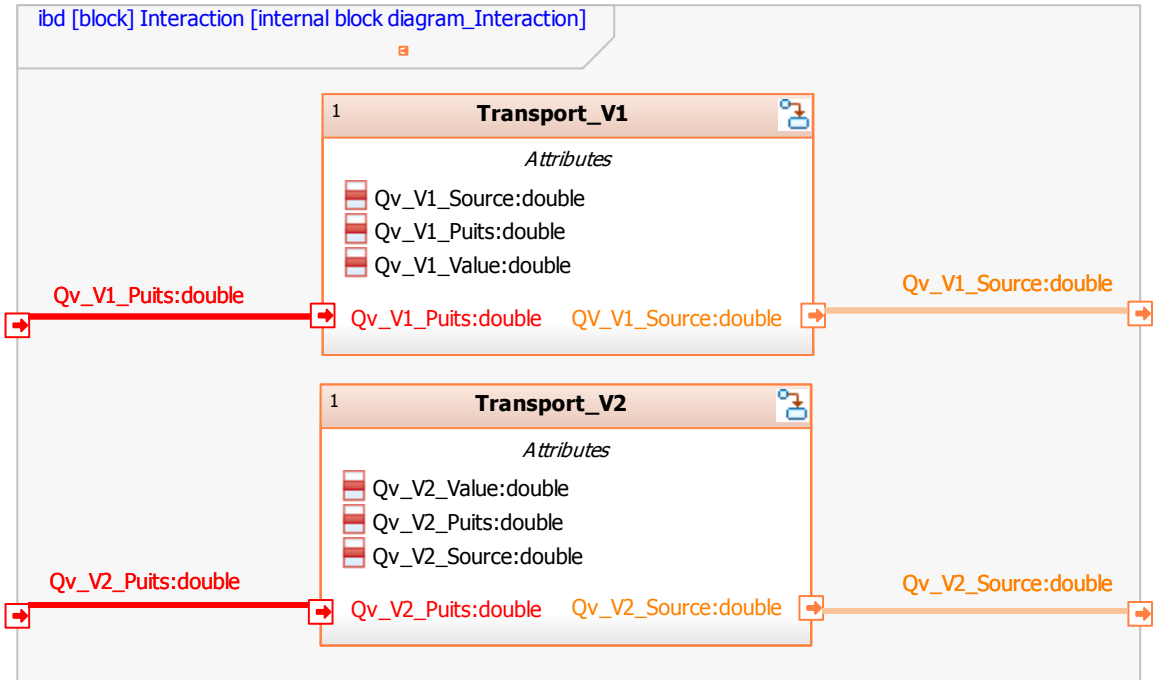


Figure 192 : Spécification des fonctions de transport par les voies V1 et V2 entre la plate-forme CISPI et son environnement

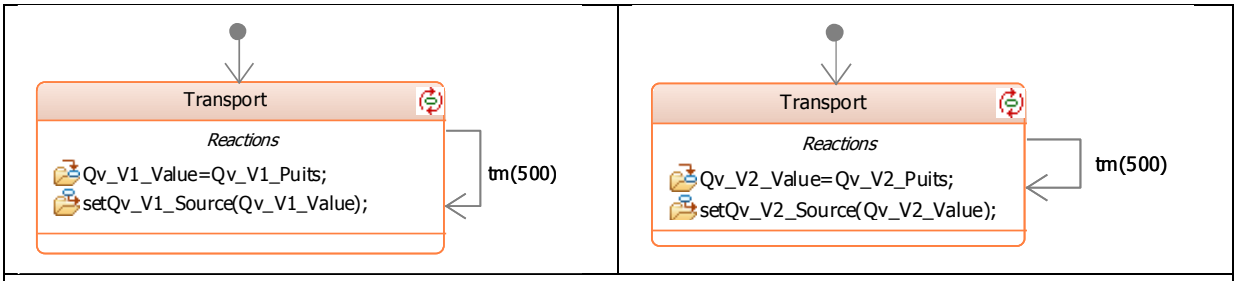


Figure 193 : Spécification de la dynamique des fonctions de transport par les voies V1 et V2 selon une fonction identité (*l'entrée est égale à la sortie*)

Annexe 5-5 : Spécification du procédé de la plate-forme CISPI (S_{IPro})

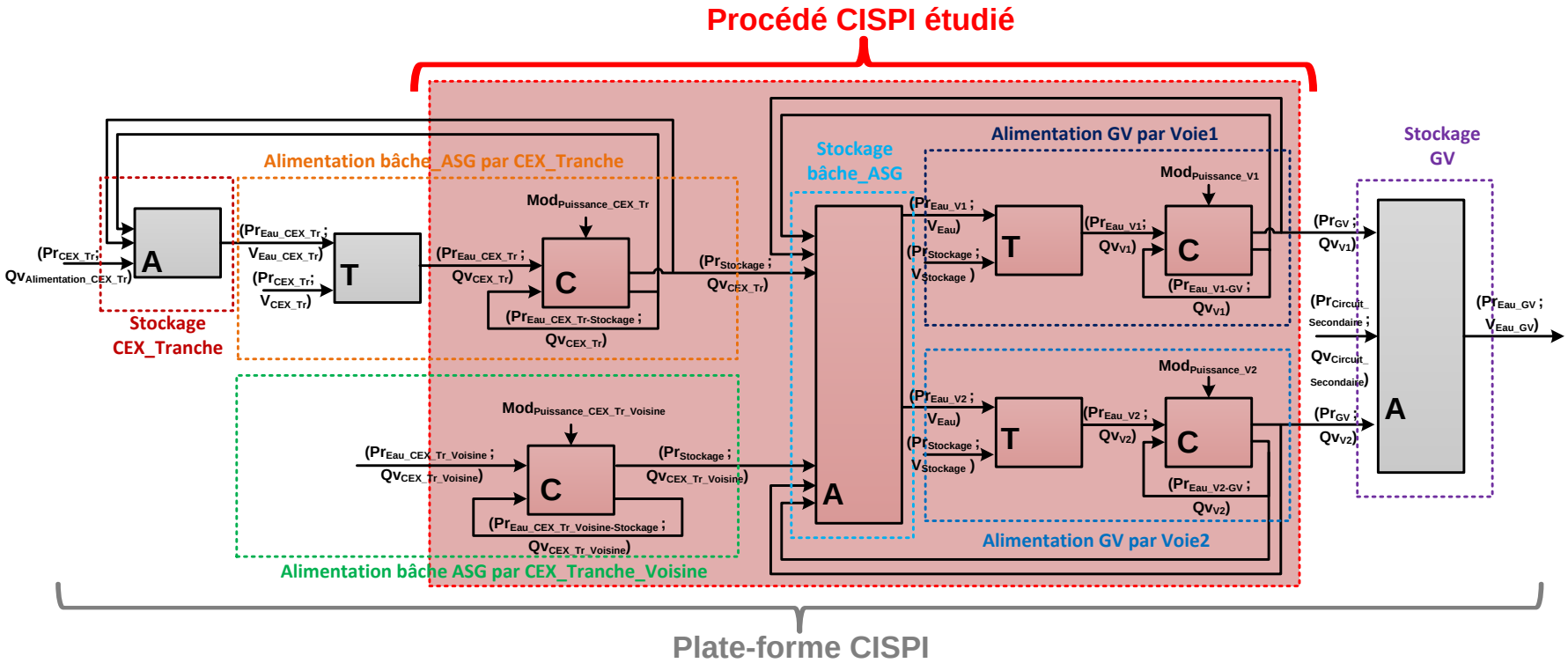
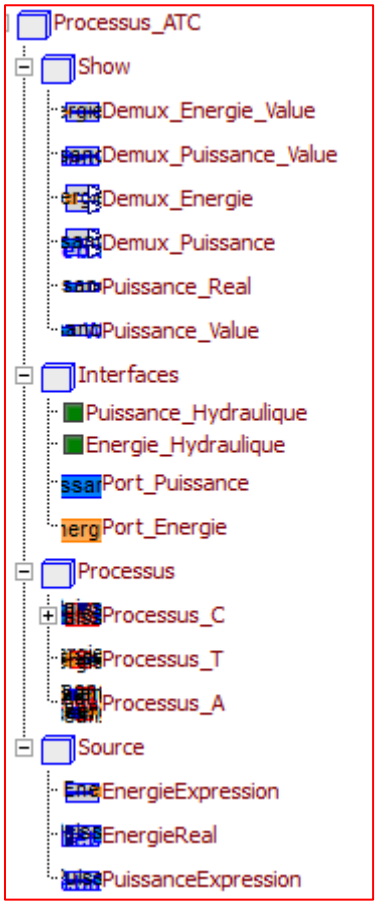


Figure 194 : Spécification du procédé CISPI à un niveau fonctionnel (Point de vue Structurel)

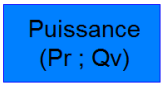
Pr_{CEX_Tr}	Pression à la surface de l'eau dans la cuve CEX_Tranche – (Cuve Ouverte) $Pr_{CEX_Tranche}$ = Pression atmosphérique
$Qv_{_CEX_Tr}$	Débit volumique d'alimentation de la cuve CEX_Tranche
$Pr_{Eau_CEX_Tr}$	Pression de l'eau en sortie de la cuve CEX_Tranche
$V_{eau_CEX_Tr}$	Volume de l'eau à l'intérieur de la cuve CEX_Tranche
V_{CEX_Tr}	Volume du fluide à la surface de l'eau - (Cuve Ouverte) $V_{CEX_Tranche}$ = Volume de l'air = ∞
$Mod_{Puissance_CEX_Tr}$	Commande du débit d'alimentation de la voie CEX_Tranche
$Pr_{Eau_CEX_Tr_Voisine}$	Pression de l'eau en sortie de la cuve CEX_Tranche_Voisine
$Mod_{Puissance_CEX_Tr_Voisine}$	Commande du débit d'alimentation de la voie CEX_Tranche_Voisine
Qv_{CEX_Tr}	Débit volumique d'alimentation de la bâche ASG par la voie CEX_Tranche
$Pr_{Stockage}$	Pression à la surface de l'eau dans la bâche d'ASG – (Cuve Ouverte) $Pr_{Stockage}$= Pression atmosphérique
$Qv_{CEX_Tr_Voisine}$	Débit volumique d'alimentation de la bâche ASG par la voie CEX_Tranche_Voisine
Pr_{Eau_V1}	Pression de l'eau en sortie (Voie1) de la bâche ASG
V_{Eau}	Volume de l'eau à l'intérieur de la bâche ASG
$V_{Stockage}$	Volume du fluide à la surface de l'eau - (Cuve Ouverte) $V_{Stockage}$= Volume de l'air = ∞
Qv_{V1}	Débit volumique de l'eau à travers la voie 1
$Mod_{Puissance_v1}$	Commande du débit d'alimentation de la voie 1
Pr_{GV}	Pression à la surface de l'eau dans le GV– (Cuve Ouverte) Pr_{GV} = Pression atmosphérique
Pr_{Eau_V2}	Pression de l'eau en sortie (Voie2) de la bâche ASG
Qv_{V2}	Débit volumique de l'eau à travers la voie 2
$Mod_{Puissance_v2}$	Commande du débit d'alimentation de la voie 2
$Pr_{Circuit_Secondaire}$	Pression dans le circuit secondaire
$Qv_{Circuit_Secondaire}$	Débit volumique de l'eau dans le circuit secondaire
Pr_{Eau_GV}	Pression de l'eau en sortie du GV
V_{Eau_GV}	Volume d'eau à l'intérieur du GV

Table 1 : Légende spécification du procédé CISPI (S_{IPro})

Annexe 5-6 : Bibliothèque Modelica – Processus Accumulation, Transport, Conversion



Port :

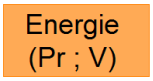
- 

Processus_TEF.Interfaces.Port_Puissance

Modélise un port de type « puissance » représenté par un couple : {pression (en Pa) ; débit volumique (en $m^3.s^{-1}$)}.

Modelica Definition:

```
connector Port_Puissance
extends Processus_TEF.Interfaces.Puissance_Hydraulique;
end Port_Puissance;
```

- 

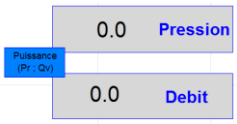
Processus_TEF.Interfaces.Port_Energie

Modélise un port de type « énergie » représenté par un couple : {pression (en Pa) ; volume (en m^3)}.

Modelica Definition:

```
connector Port_Energie
extends Processus_TEF.Interfaces.Energie_Hydraulique;
end Port_Energie;
```

Interaction :

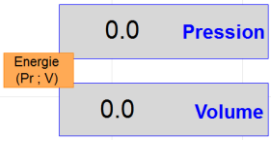
- 

Processus_TEF.Show.Puissance_Value

Affiche les valeurs des variables composant un couple de type « puissance» : {pression (en Pa) ; débit volumique (en $m^3.s^{-1}$)}

Modelica Definition:

```
block Puissance_Value
Real Qv;
Real Pr;
Interfaces.Port_Puissance port_Puissance
equation
Qv = port_Energie.Qv;
Pr = port_Energie.Pr;
end Puissance_Value;
```

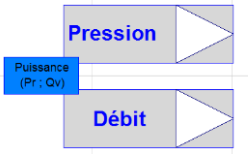
- 

Processus_TEF.Show.Energie_Value

Affiche les valeurs des variables composant un couple de type « énergie » : {pression (en Pa) ; volume (en m^3)}

Modelica Definition:

```
block Energie_Value
Real V;
Real Pr;
Interfaces.Port_Energie port_Energie
equation
 V = port_Puissance.V;
 Pr = port_Puissance.Pr;
end Energie_Value;
```

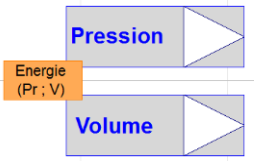
- 

Processus_TEF.Show.Puissance_Real

Divise un couple de type «puissance » en deux connecteurs de type réel : {pression; débit volumique}

Modelica Definition:

```
block Puissance_Real
Interfaces.Port_Puissance port_Puissance
Modelica.Blocks.Interfaces.RealOutput Qv
Modelica.Blocks.Interfaces.RealOutput Pr
equation
 Qv = port_Puissance.Qv;
 Pr = port_Puissance.Pr;
end Puissance_Real;
```

- 

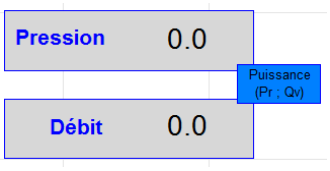
Processus_TEF.Show.Energie_Real

Divise un couple de type «énergie » en deux connecteurs de type réel : {pression ; volume}

Modelica Definition:

```
block Energie_Real
Interfaces.Port_Energie port_Energie
Modelica.Blocks.Interfaces.RealOutput V
Modelica.Blocks.Interfaces.RealOutput Pr
equation
 V = port_Energie.V;
 Pr = port_Energie.Pr;
end Energie_Real;
```

Source :

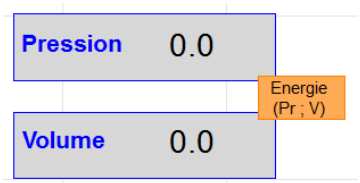
- 

Processus_TEF.Source.Puissance_Expression

Définit les valeurs des variables composant un couple de type «puissance » : {pression (en Pa) ; débit volumique (en $m^3.s^{-1}$)} sous la forme de deux réels.

Modelica Definition:

```
block Puissance_Expression
parameter Real Pr;
parameter Real Qv;
Interfaces.Port_Puissance port_Puissance
equation
port_Puissance.Pr = Pr;
port_Puissance.Qv = Qv;
end Puissance_Expression;
```

- 

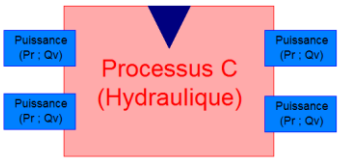
Processus_TEF.Source.Energie_Expression

Définit les valeurs des variables composant un couple de type «énergie » : {pression (en Pa) ; volume (en m^3)} sous la forme de deux réels.

Modelica Definition:

```
block Energie_Expression
parameter Real Pr;
parameter Real V;
Interfaces.Port_Energie port_Energie
equation
port_Energie.Pr = Pr;
port_Energie.V = V;
end Energie_Expression;
```

Processus :

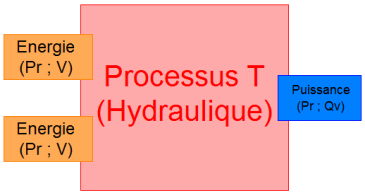
- 

Processus_TEF.Processus.Processus_C

Un processus de dissipation de type F, dû à la restriction d'une canalisation par exemple, produit une puissance en sortie et des pertes de charges que l'on reboucle à l'entrée du processus pour traduire d'un point de vue énergétique une contre réaction.

Modelica Definition:

```
model Processus_C
parameter Modelica.SIunits.Pressure Pression_Sortie = 101325 "Pression_Cuve_Sortie";
parameter Real k = 0.05 "Constante ";
parameter Real Diametre_Forme = 0.038 "Diametre Forme";

Real Pression_Voie;
Real Section_Voie;

Interfaces.Port_ Puissance port_Entree1
Interfaces.Port_ Puissance port_Entree2
Interfaces.Port_ Puissance port_Sortie2
Interfaces.Port_ Puissance port_Sortie1
Modelica.Blocks.Interfaces.RealInput Mod_Puissance
equation
Section_Forme = (Diametre_Forme)^2 /4 * Modelica.Constants.pi;
Pression_Voie = if opening > 0 then (k*995.58/2* (port_Entree1.Qv/Section_Forme)^2)/2*Mod_Puissance+ Pression_Sortie else Pr
ession_Sortie;
port_Sortie1.Qv= if opening >0 then port_Entree1.Qv else 0;
port_Sortie1.Pr= Pression_Sortie;
port_Sortie2.Qv= port_Entree1.Qv;
port_Sortie2.Pr= Pression_Voie - Pression_Sortie;

end Processus_C;
```

- 

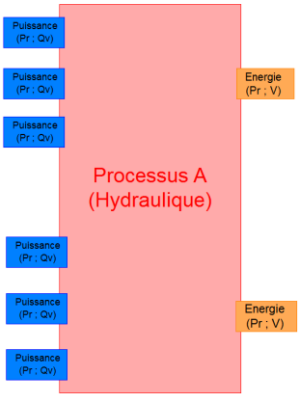
Processus_TEF.Processus.Processus_T

Un processus d'échange de type T, produit un flux de puissance. Cette circulation est due à une différence de niveau d'énergie entre l'énergie en entrée (Pr, V) et l'énergie en sortie (Pr ; V).

Modelica Definition:

```
model Processus_T
parameter Modelica. SIunits.Area Section_Cuve ;
parameter Modelica.SIunits.Area Section_Sortie ;
Interfaces.Port_Puissance port_Sortie
Interfaces.Port_ Energie port_Entree1
Interfaces.Port_ Energie port_Entree2
equation
```

```
port_Sortie.Pr=port_Entree1.Pr;
port_Sortie.Qv= Section_Sortie*sqrt(abs(((2*(port_Entree2.Pr -port_Entree1.Pr)) /995.58) + 2*9.80665 * (port_Entree1.V /
Section_Cuve)));
end Processus_T;
```



- **Processus_TEF.Processus.Processus_A**

Ce processus de stockage de type T, produit deux énergies hydrauliques en sortie (deux sorties au niveau du stockage). Cette énergie résulte du bilan de l'accumulation des puissances hydrauliques en entrée et en sortie.

Modelica Definition:

```
model Processus_A
parameter Modelica.SIunits.Area Section_Cuve = 1;
parameter Real h0 = 3.0 "Hauteur Initiale";
parameter Modelica.SIunits.Pressure Pression_Cuve = 101325  "Pression à la surface de l'eau)";
Real h "Hauteur Cuve";
Modelica.SIunits.Pressure Pression_Cuve_0 "Pression en sortie de cuve à débit nul";
Real Integrale_DE "Integrale_Debit_Entree";
Real Integrale_DS "Integrale_Debit_Sortie";
Interfaces.Port_ Puissance port_Entree3
Interfaces.Port_ Puissance port_Entree4
Interfaces.Port_Energie port_Sortie1
Interfaces.Port_ Puissance port_Entree5
Interfaces.Port_ Puissance port_Entree6
Interfaces.Port_ Puissance port_Entree2
Interfaces.Port_ Puissance port_Entree1
Interfaces.Port_ Energie port_Sortie2
 equation
der(Integrale_DE) = port_Entree3.Qv + port_Entree6.Qv;
der(Integrale_DS) = port_Entree1.Qv + port_Entree4.Qv;
h = h0 + 1/Section_Cuve *(Integrale_DE - Integrale_DS);
Pression_Cuve_0 = (h*995.58*9.80665)+ Pression_Cuve_0;
port_Sortie1.Pr = if port_Entree1.Qv >0 then port_Entree1.Pr+port_Entree2.Pr else Pression_Cuve_0;
port_Sortie1.V = Section_Cuve * h;
port_Sortie2.Pr = if port_Entree4.Qv >0 then  port_Entree4.Pr+port_Entree5.Pr else Pression_Cuve_0;
port_Sortie2.V = Section_Cuve * h;
end Processus_A;
```

Annexe 5-7 : Table de correspondance SysML-Modelica

Modelica	SysML
Model (Class) *(constitue la brique de base pour la modélisation de la structure d'un procédé. Il peut représenter un procédé, un processus, un composant… Il contient des équations qui décrivent les relations entre les ports de connexion (Connector))*	**Block** (*constitue la brique de base pour la modélisation de la structure d'un système. Il peut représenter un système complet, un sous-système ou un composant élémentaire à différents niveaux d'abstractions). Les blocs peuvent posséder également un comportement*)
(Object) *(Instance de modèle Modelica dimensionné pour un usage)*	**Parts** (*instance de bloc*)
Variable (Constant/Parameter/Continuous/Discrete) décrit des données appartenant au « model »)	**Value** (*décrit des caractéristiques du bloc. Les valeurs sont des variables stockant des informations d'état du bloc*)
Type (*décrit le domaine de la valeur, dimension et unité (optionnel))*	**ValueType** (*décrit le domaine de la valeur, dimension et unité (optionnel))*
Connector (*Souvent appelés Ports, spécifie les interfaces externes du modèle pour les interactions*)	**FlowPort** (*définit un point d'interaction entre les ports. Les ports de type FlowPort autorise la circulation de flux physiques (matières, énergies, données) entre les blocs*)
Connector	**Flow Specification** *(spécifie le type d'un flow port non atomique, incluant plusieurs propriétés de flux (Flow Properties))*
Connector	**Flow Properties** (*Définit la propriété d'un flux selon un nom, un type et une direction*)
Equation « Connect » (*L'équation de connexion est une équation spéciale qui spécifie une interaction entre des connecteurs Modelica de même type*)	**Connector** (*relie deux parties entre elles, leur fournissant l'opportunité d'agir à travers des ports*)
Equation (*spécifie le comportement du « model » selon une modélisation acausale (pas de direction du flux de données))* + ***Variable***	**ConstraintBlock** (*conditionne l'énoncé d'une contrainte de sorte qu'il peut être appliqué d'une manière réutilisable pour contraindre les propriétés d'autres blocs. Il définit généralement un ou plusieurs paramètres de contraintes qui sont liées aux propriétés d'autres blocs*)
Equation	**ConstraintProperty** (*Instance d'un ConstraintBlock pour une utilisation localisée*)
Equation	**Constraint** (*représente une expression mathématique contraignant les propriétés physique d'un système*)
Variable	**ConstraintParameters** (*Paramètres d'une contrainte*)

Tableau 23 : Tableau de correspondance SysML / Modelica

Annexe 5-8 : Modélisation de la structure des processus énergétiques (diagramme interne de blocs SysML)

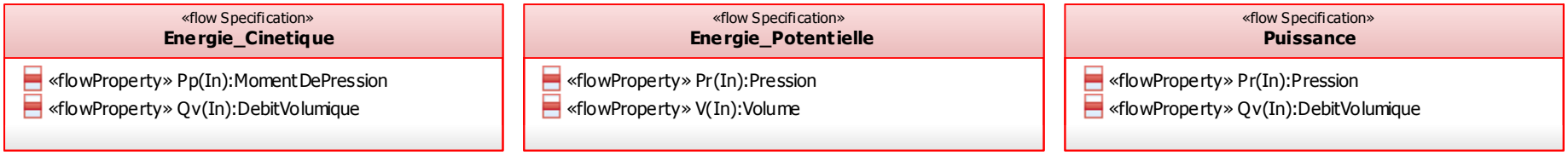


Figure 195 : Spécification de flux non atomique (Flow Specification)

Annexe 5-9 : Modélisation de la dynamique des processus énergétiques (diagrammes paramétriques SysML)

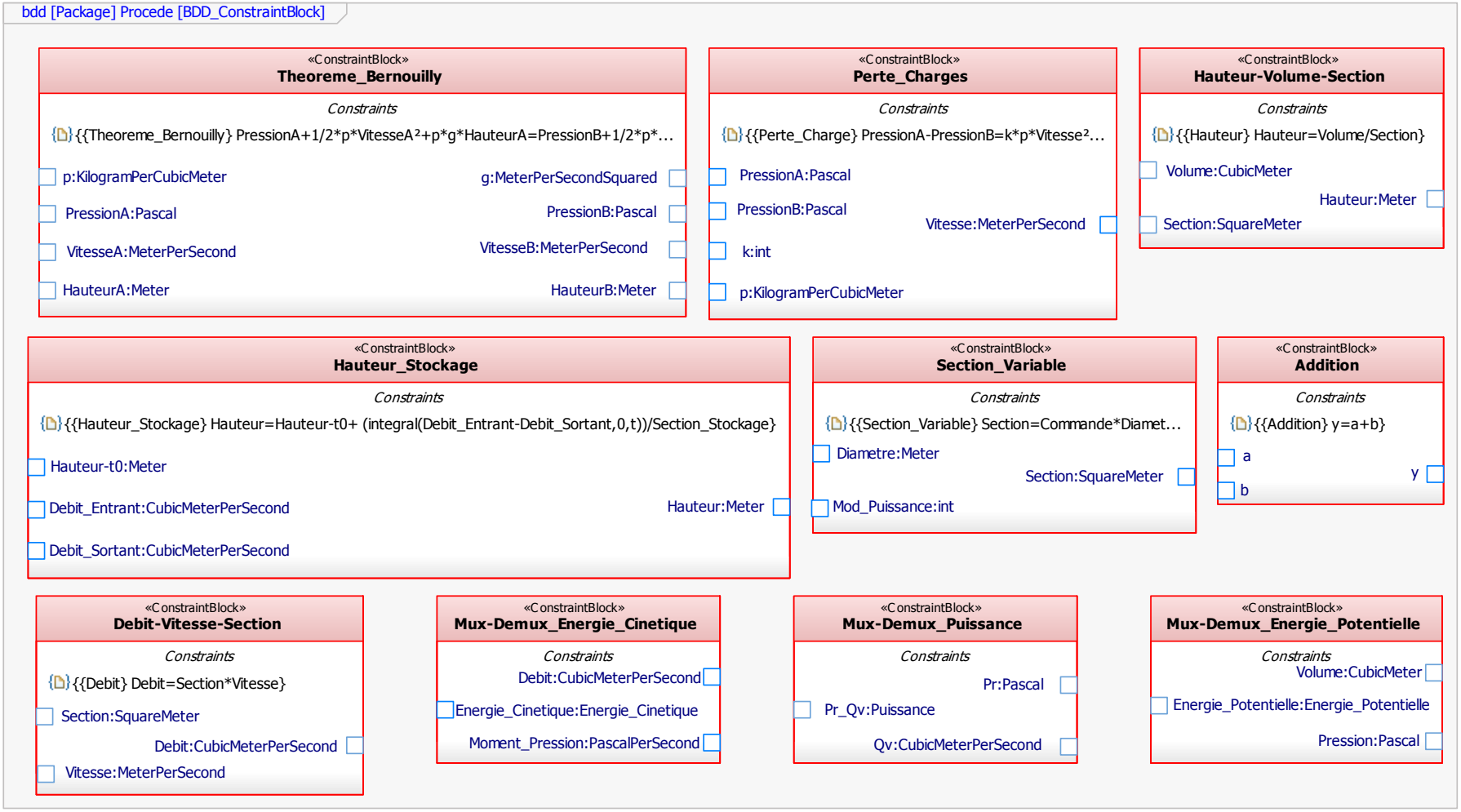


Figure 196 : Diagramme de définition de blocs (bdd) SysML - "Constraint blocks " utilisés pour la définition du comportement des processus A,T,C

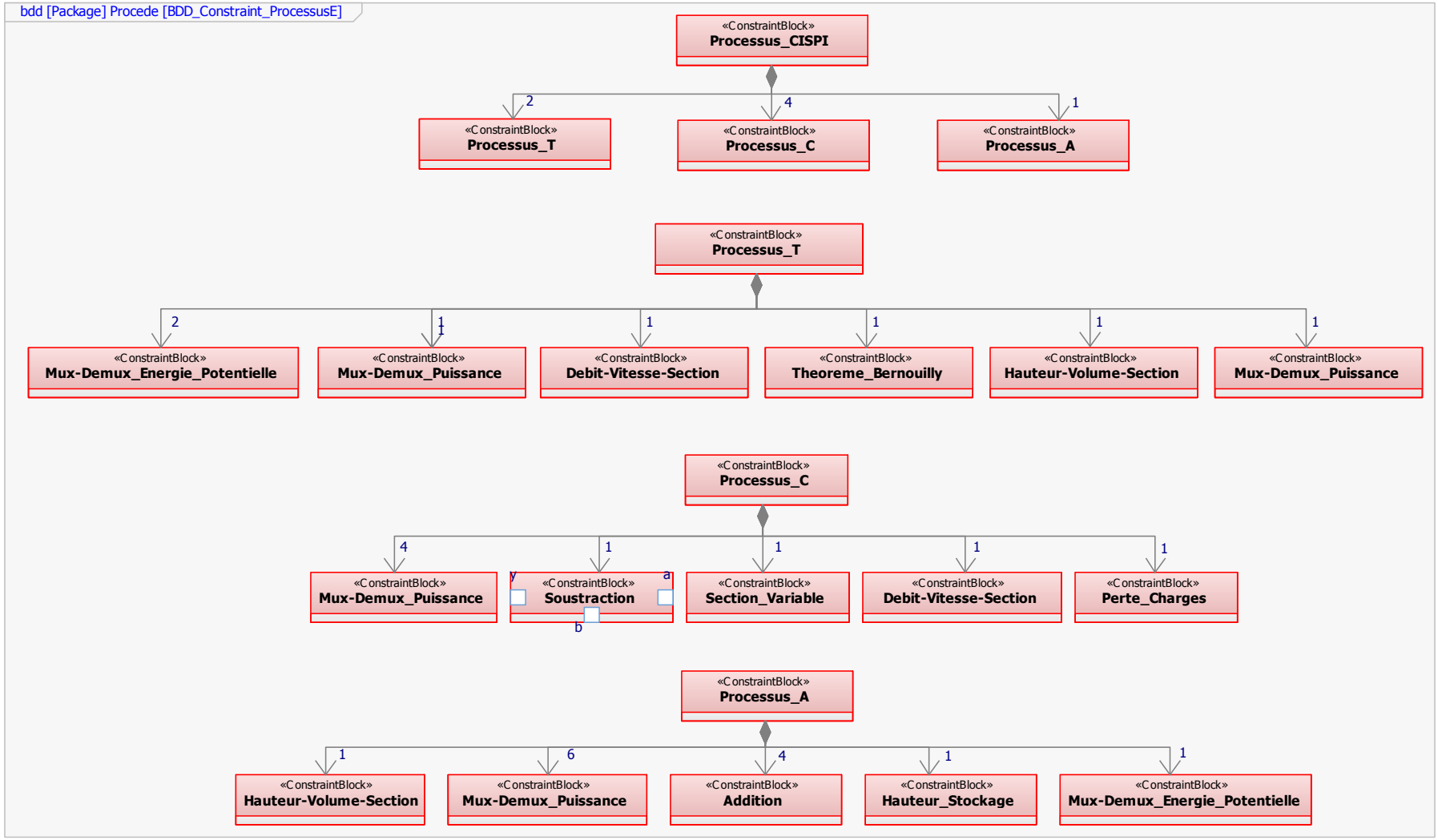


Figure 197 : Diagramme de définition de blocs (bdd) SysML - Composition des processus Accumulation, Transport, Conversion du processus CISPI

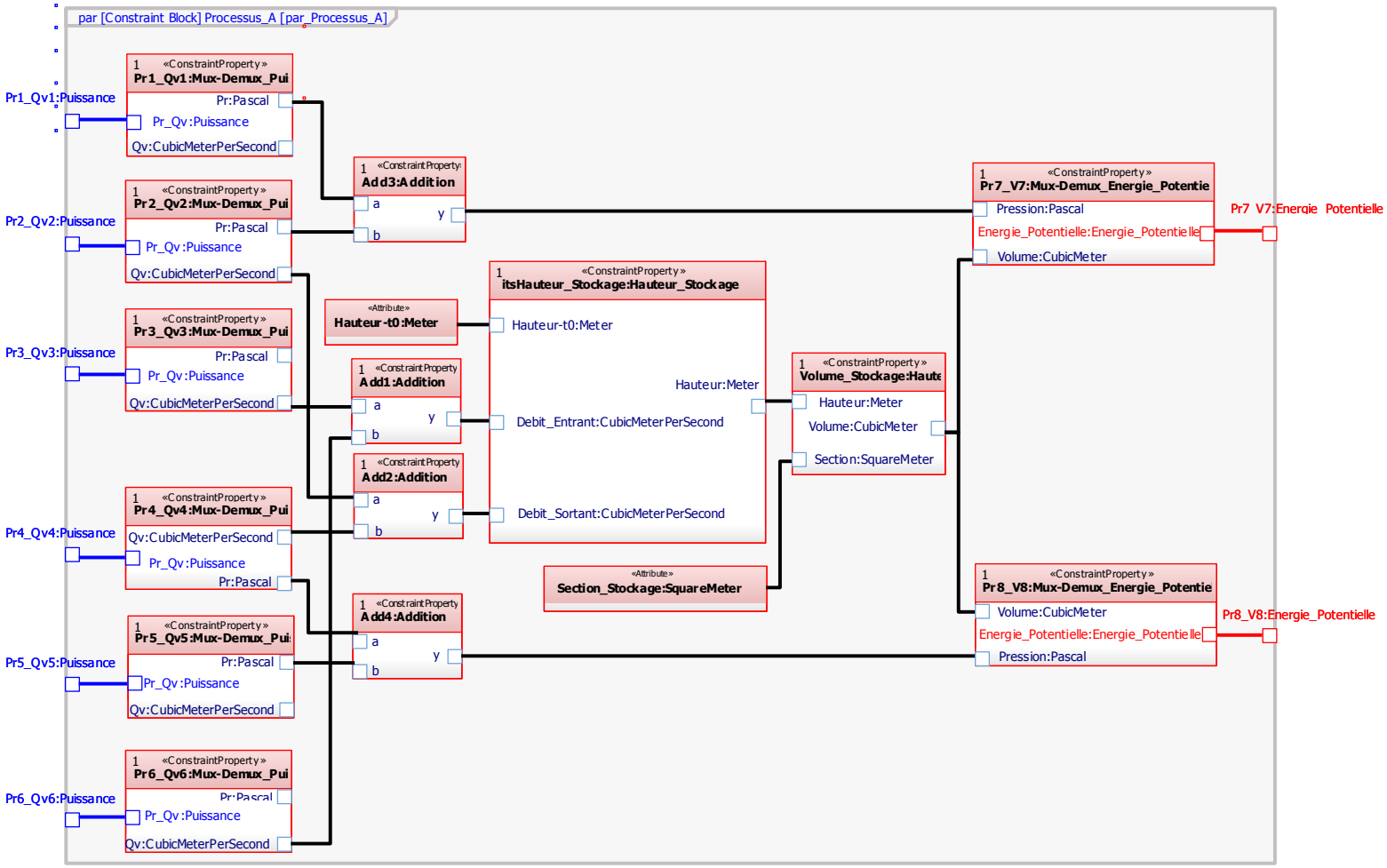


Figure 198 : Diagramme paramétrique (par) SysML – Dynamique processus A

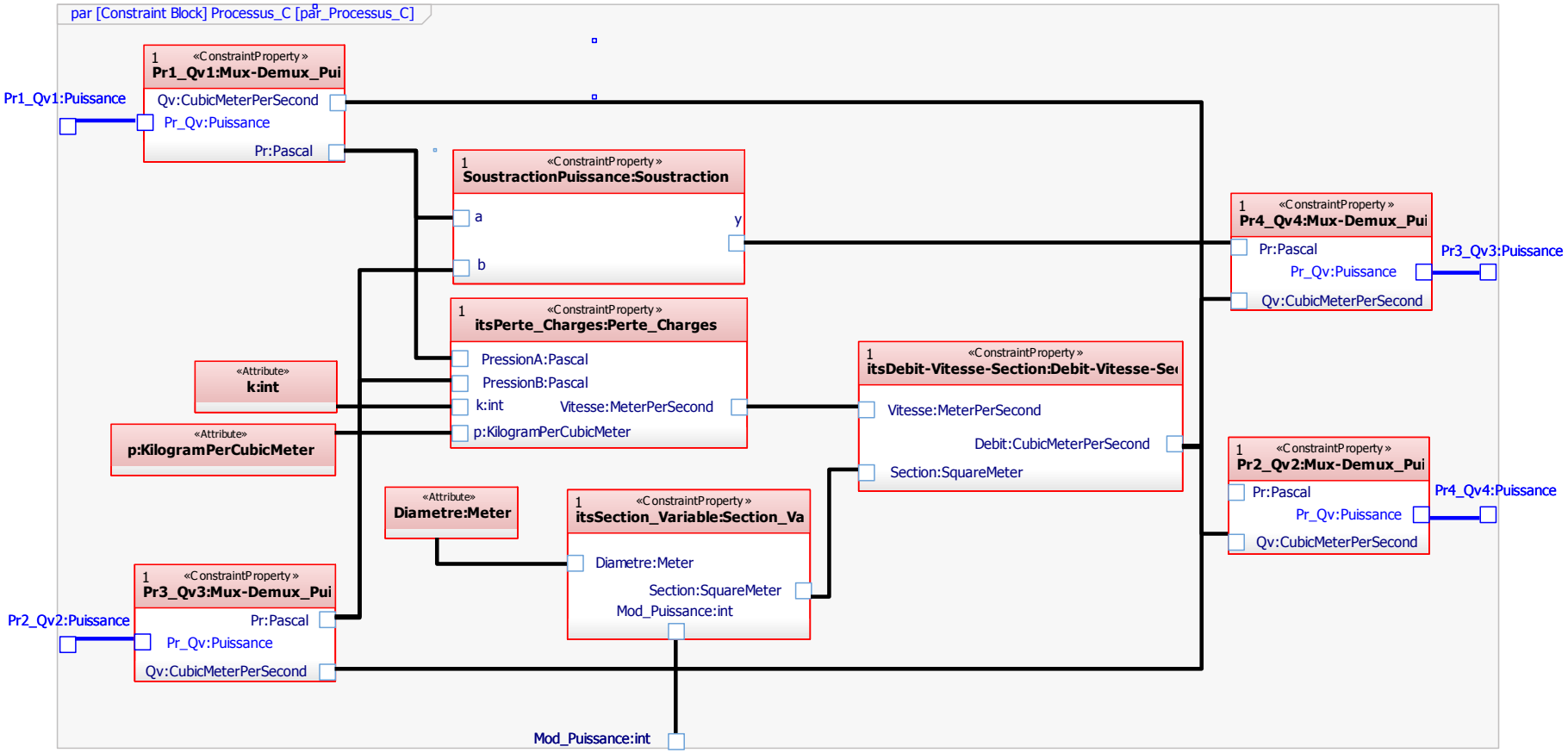


Figure 199 : Diagramme paramétrique (par) SysML – Dynamique processus C

Annexe 5-10 : Environnement de co-simulation pour la validation du modèle de la partie physique de la plate-forme CISPI

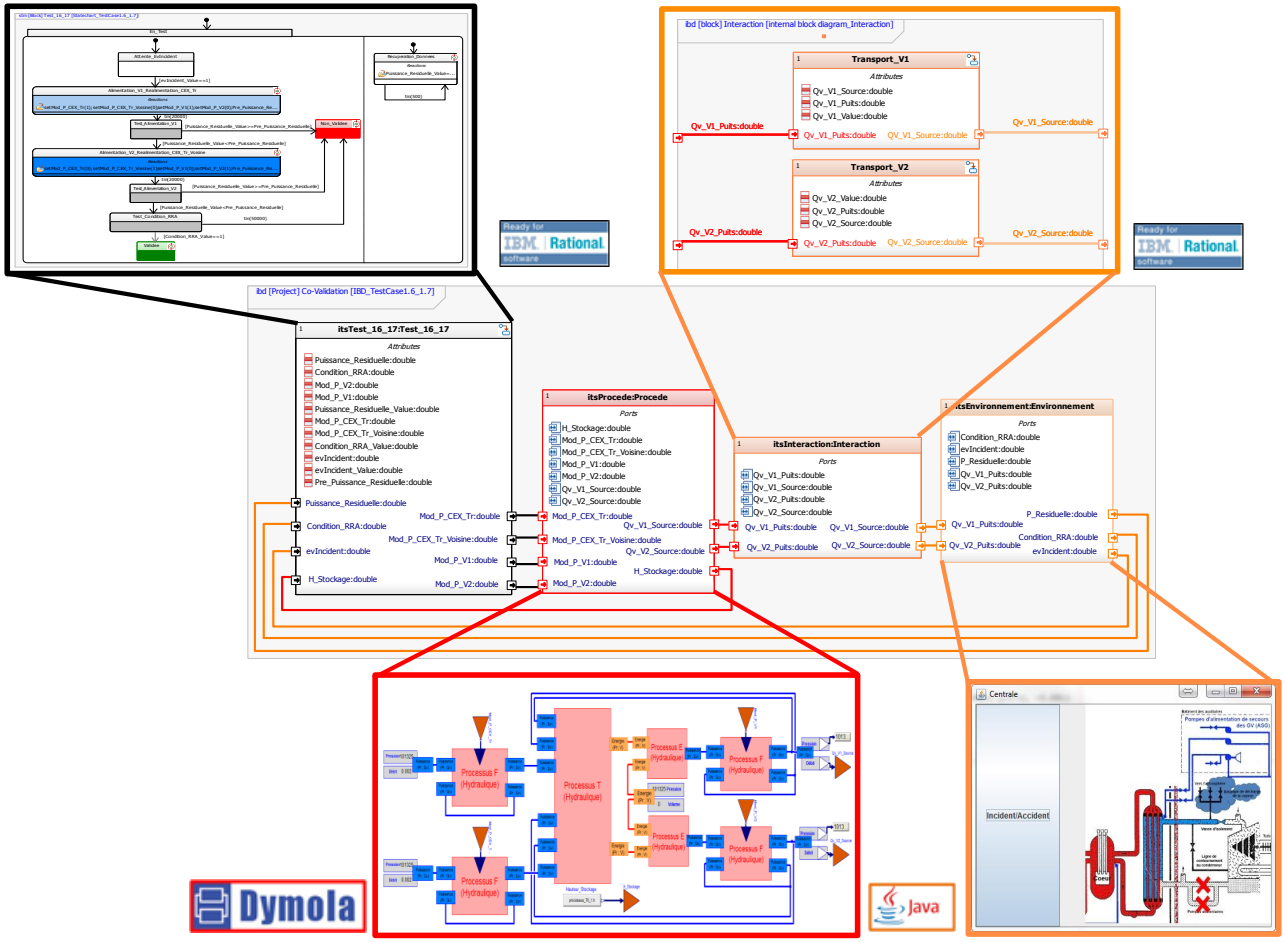


Figure 200 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle de la partie physique de la plate-forme CISPI

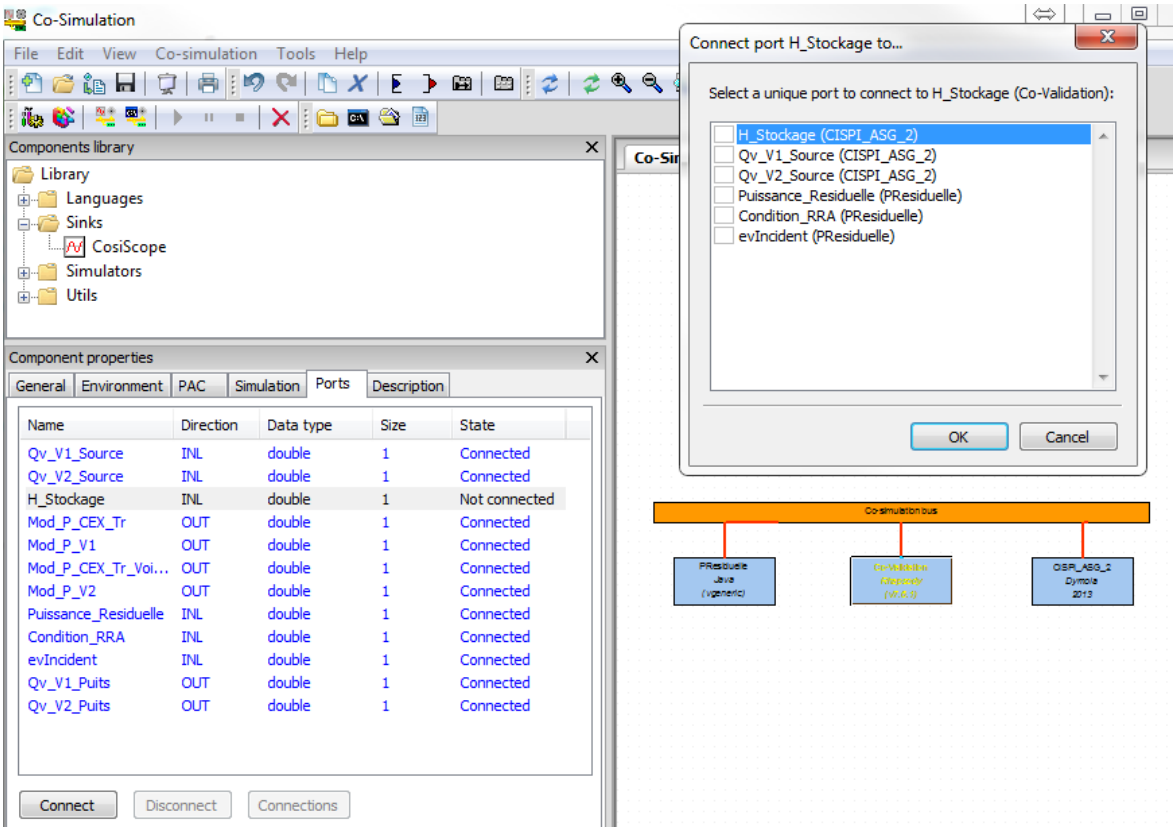


Figure 201 : Configuration de l'environnement de co-simulation pour la validation du modèle de la partie physique de la plate-forme CISPI : définition des interactions entre les modèles spécialistes au travers des ports de type "Cosimate" selon le diagramme interne de blocs (Figure 200).

Annexe 6–1 : Environnement de co-simulation pour la validation du modèle de contrôle de la partie logique de la plate-forme CISPI

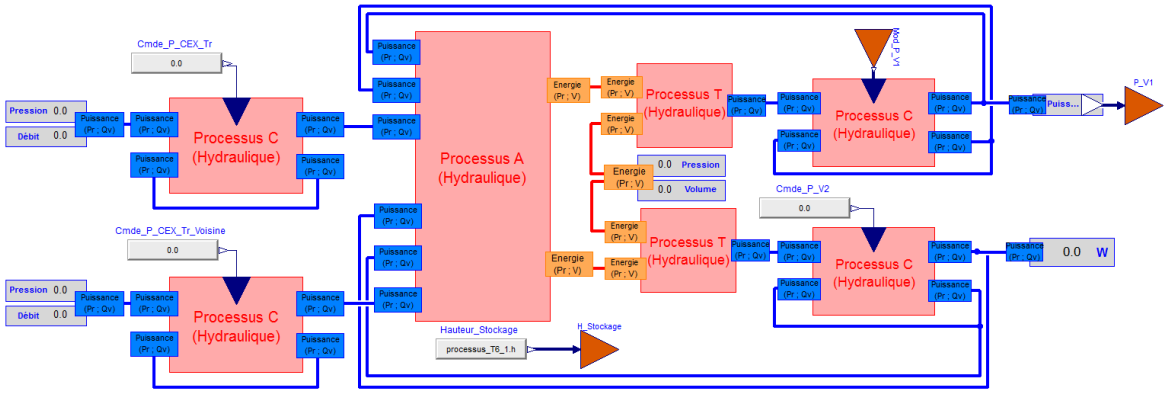


Figure 202 : Modèle du processus de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle de contrôle de la partie-logique de la plate-forme CISPI

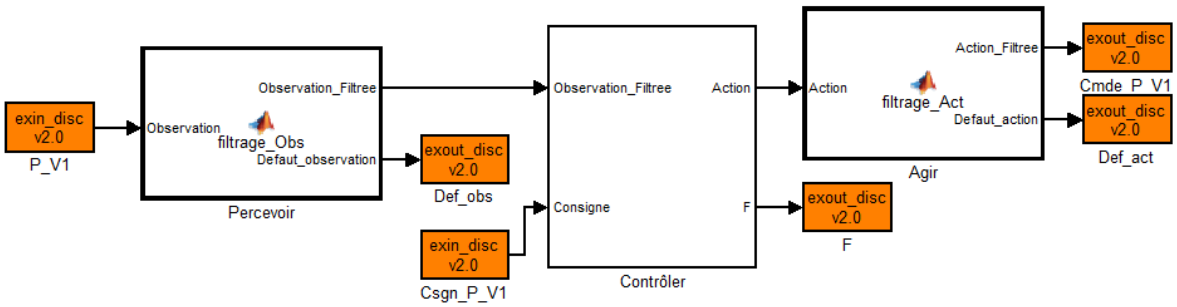


Figure 203 : Modèle de la partie contrôle de la plate-forme CISPI avec des ports de type "Cosimate"

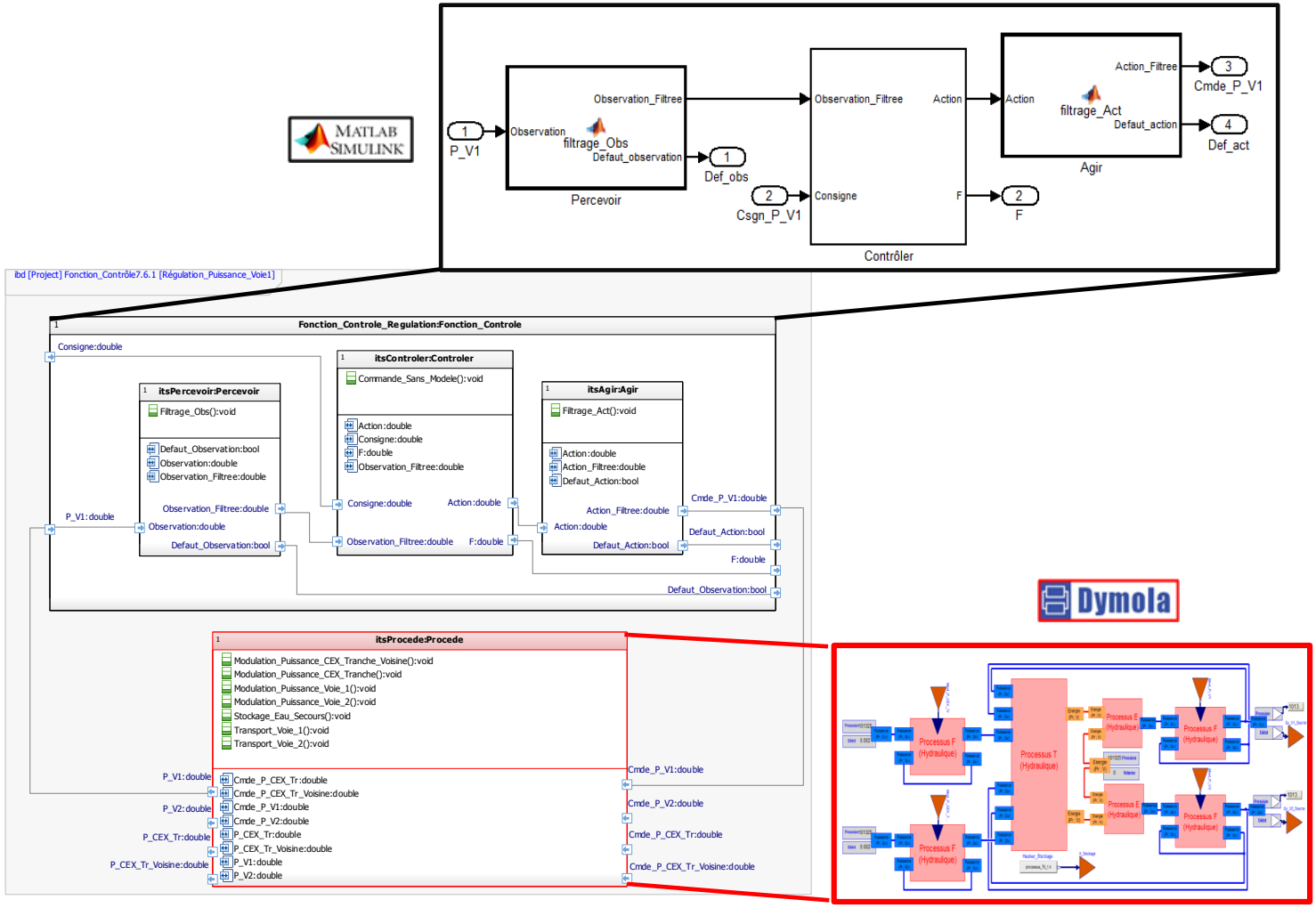


Figure 204 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle de contrôle de la partie logique de la plate-forme CISPI

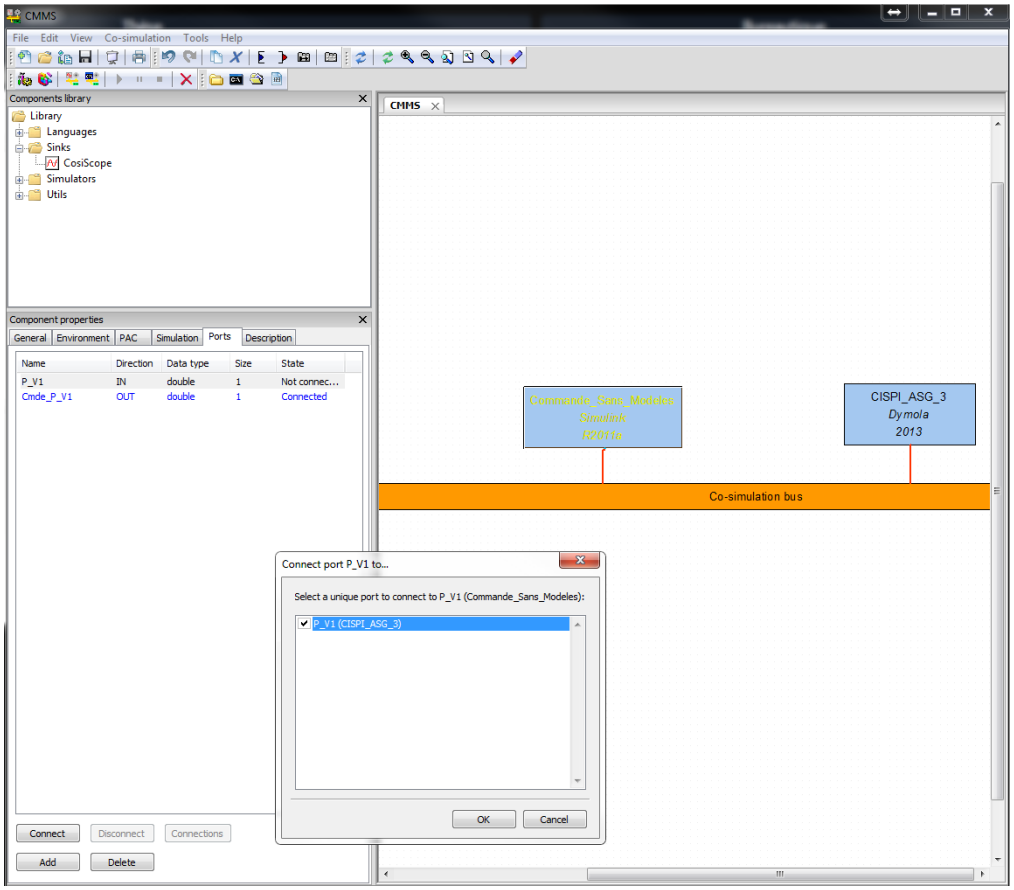


Figure 205 : Configuration de l'environnement de co-simulation pour la validation du modèle de contrôle de la partie logique de la plate-forme CISPI : définition des interactions entre les modèles spécialistes au travers des ports de type "Cosimate" selon le diagramme interne de blocs (Figure 204)

Annexe 6-2 : Modélisation du sous-système de commande

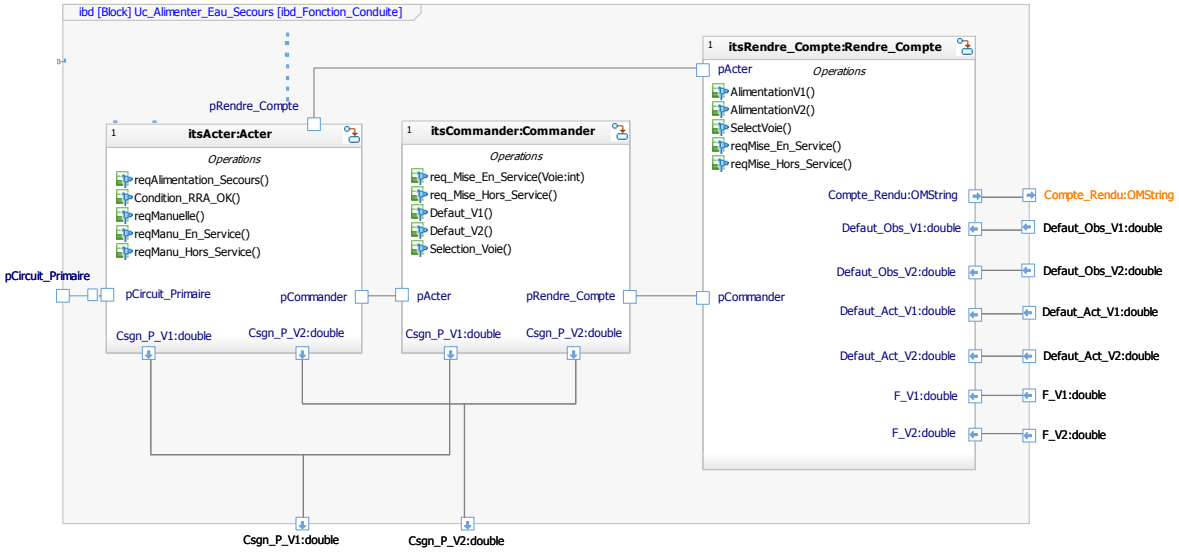


Figure 206 : Structure interne du modèle de commande

Annexe 6-3 : Environnement de co-simulation pour la validation du sous-système de commande de la partie logique de la plate-forme CISPI

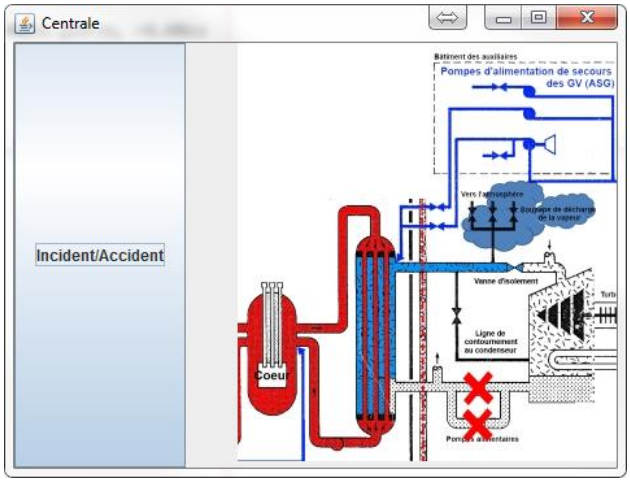


Figure 207 : Modèle de l'environnement de la plate-forme CISPI (circuit primaire) pour la validation du modèle de commande

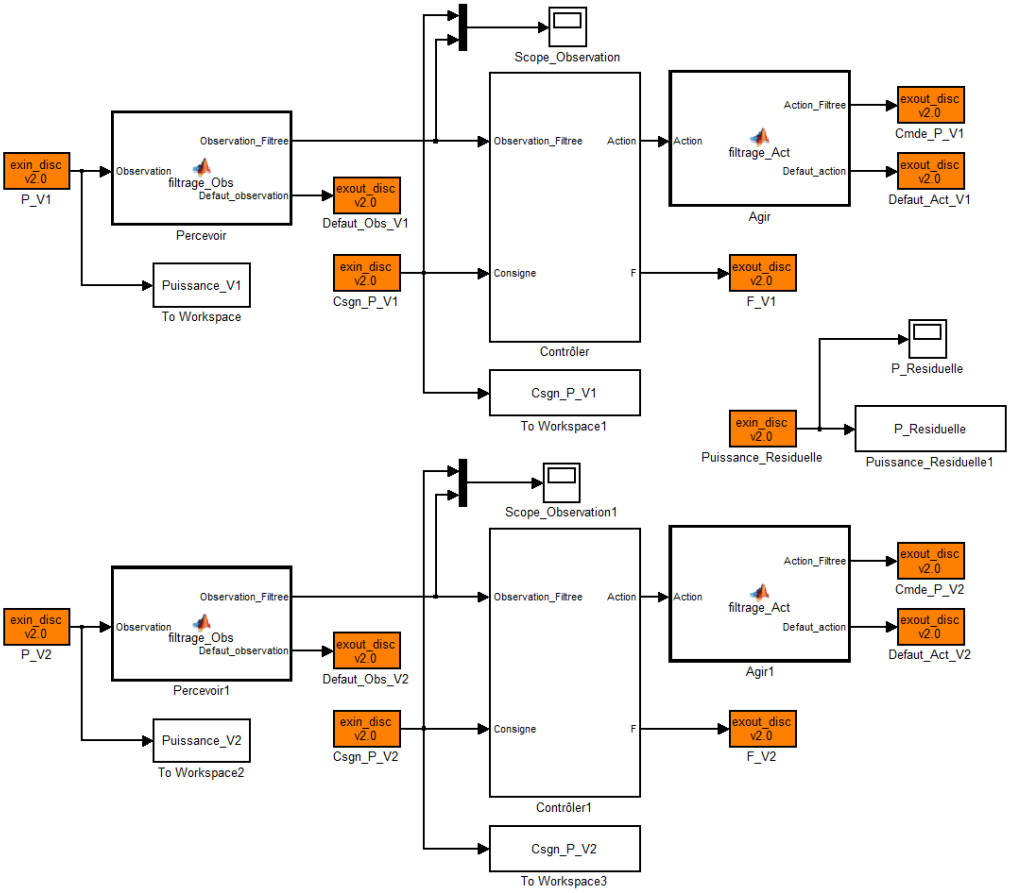


Figure 208 : Modèle de la partie contrôle de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle de commande

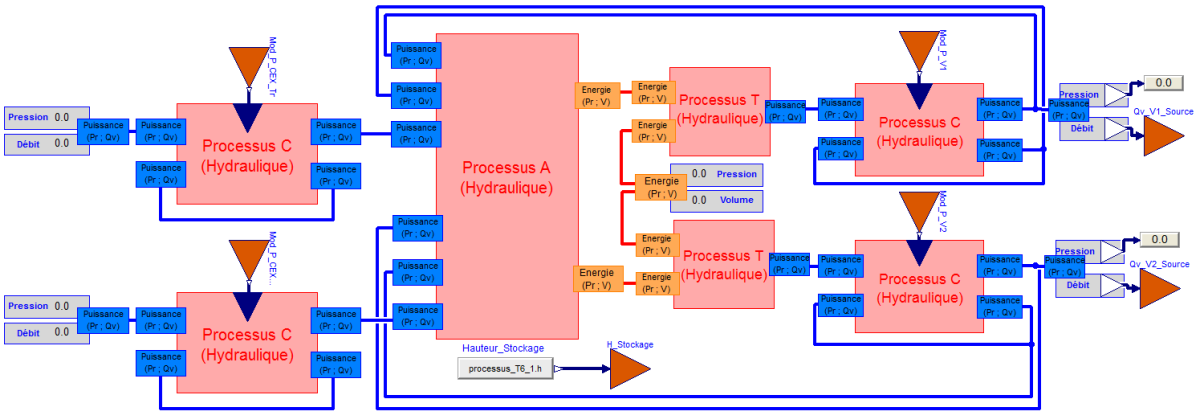


Figure 209 : Modèle du processus de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle de commande

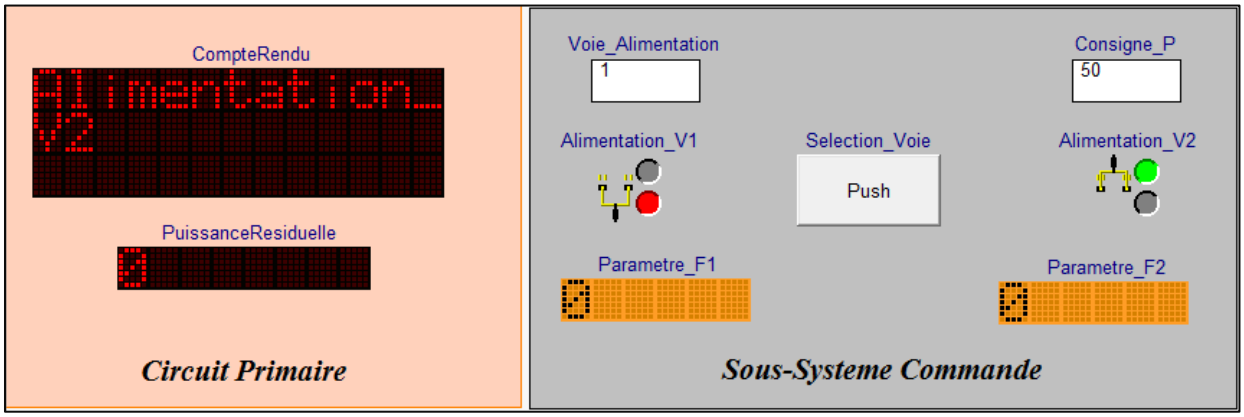


Figure 210 : Panel Diagram - Interface de supervision avec les modèles de commande et des acteurs

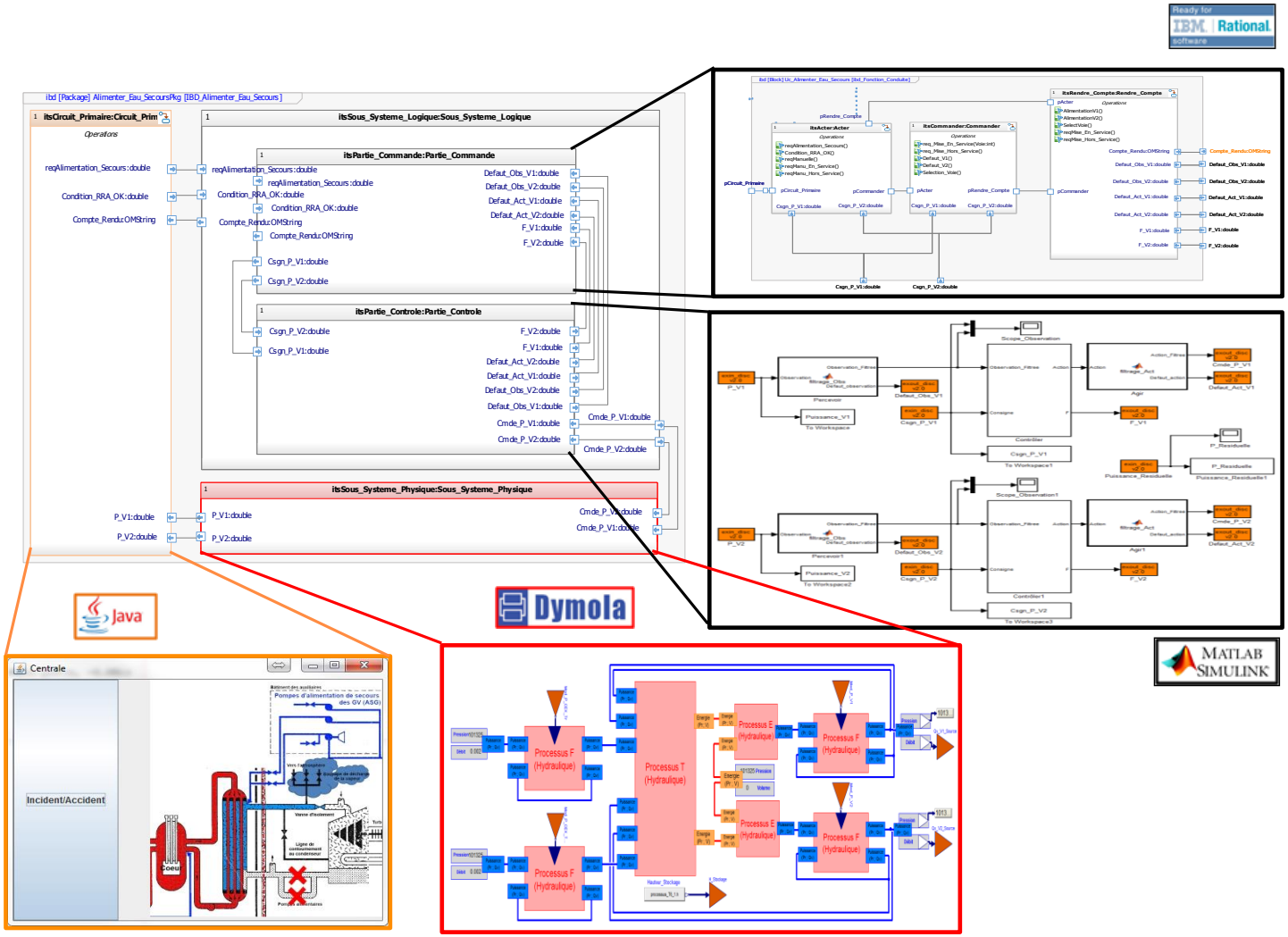
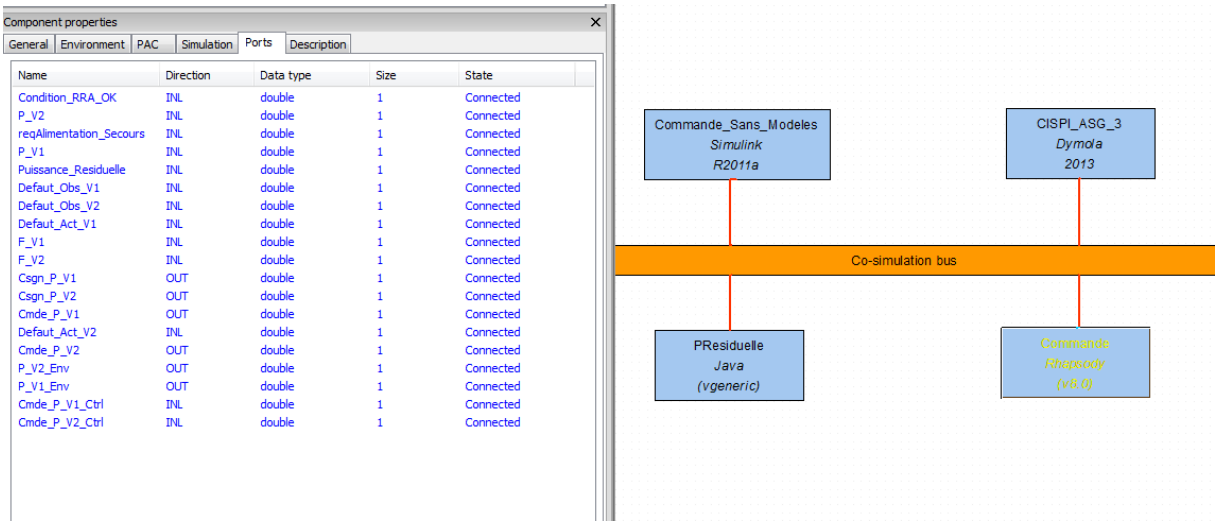


Figure 211 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle de commande de la partie logique de la plate-forme CISPI



Name	Direction	Data type	Size	State
Condition_RRA_OK	INL	double	1	Connected
P_V2	INL	double	1	Connected
reqAlimentation_Secours	INL	double	1	Connected
P_V1	INL	double	1	Connected
Puissance_Residuelle	INL	double	1	Connected
Defaut_Obs_V1	INL	double	1	Connected
Defaut_Obs_V2	INL	double	1	Connected
Defaut_Act_V1	INL	double	1	Connected
F_V1	INL	double	1	Connected
F_V2	INL	double	1	Connected
Csgn_P_V1	OUT	double	1	Connected
Csgn_P_V2	OUT	double	1	Connected
Cmde_P_V1	OUT	double	1	Connected
Defaut_Act_V2	INL	double	1	Connected
Cmde_P_V2	OUT	double	1	Connected
P_V2_Env	OUT	double	1	Connected
P_V1_Env	OUT	double	1	Connected
Cmde_P_V1_Ctrl	INL	double	1	Connected
Cmde_P_V2_Ctrl	INL	double	1	Connected

Figure 212 : Configuration de l'environnement de co-simulation pour la validation du modèle de commande de la partie logique de la plate-forme CISPI : définition des interactions entre les modèles spécialistes au travers des ports de type "Cosimate" selon le diagramme interne de blocs (Figure 204)

Annexe 7-1 : Analyse des exigences systèmes par l'espace solution fournisseur d'actionneur intelligent

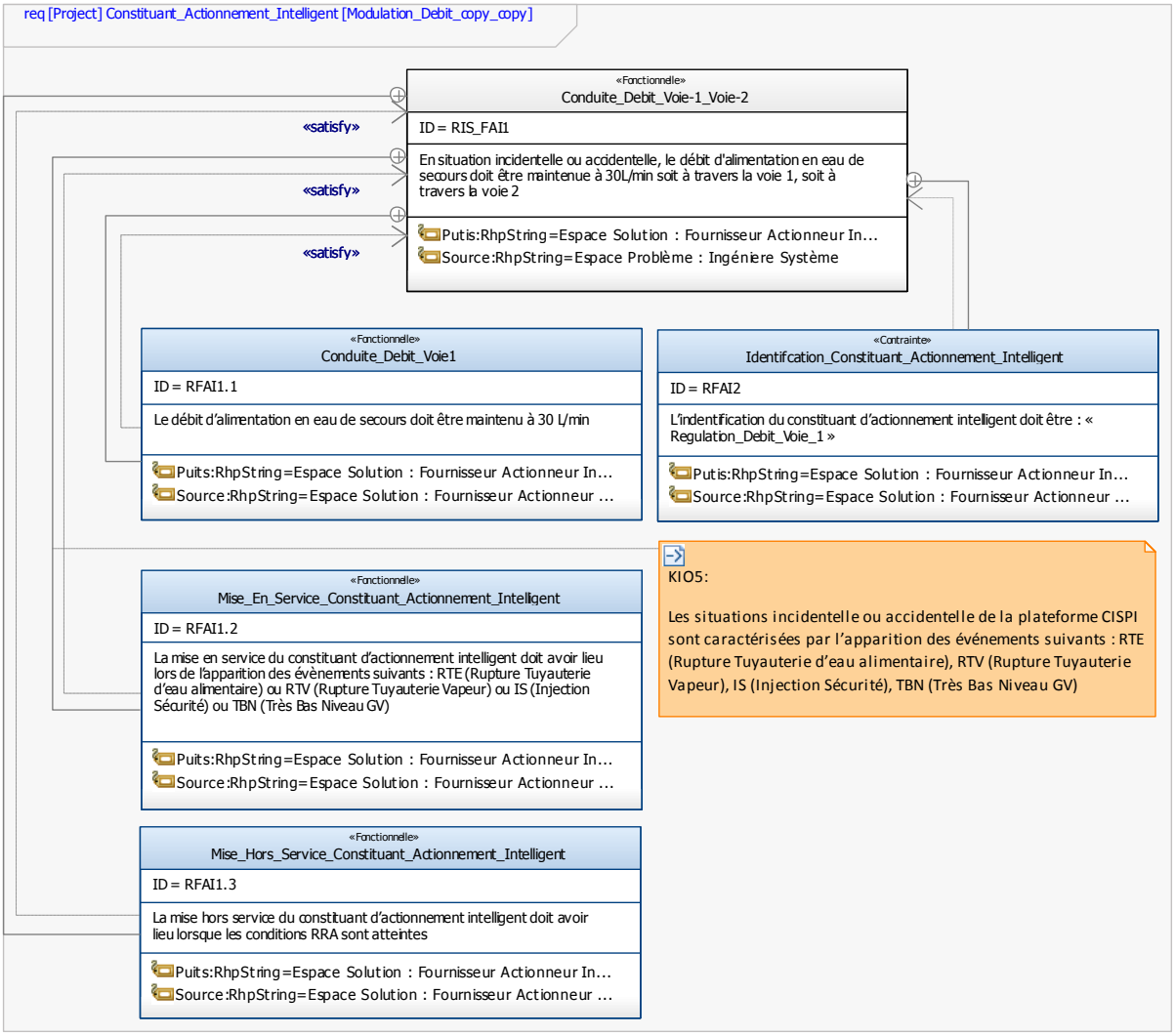


Figure 213 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence système « R_{IS_FAI1}» par l'espace-solution fournisseur d'actionneur intelligent

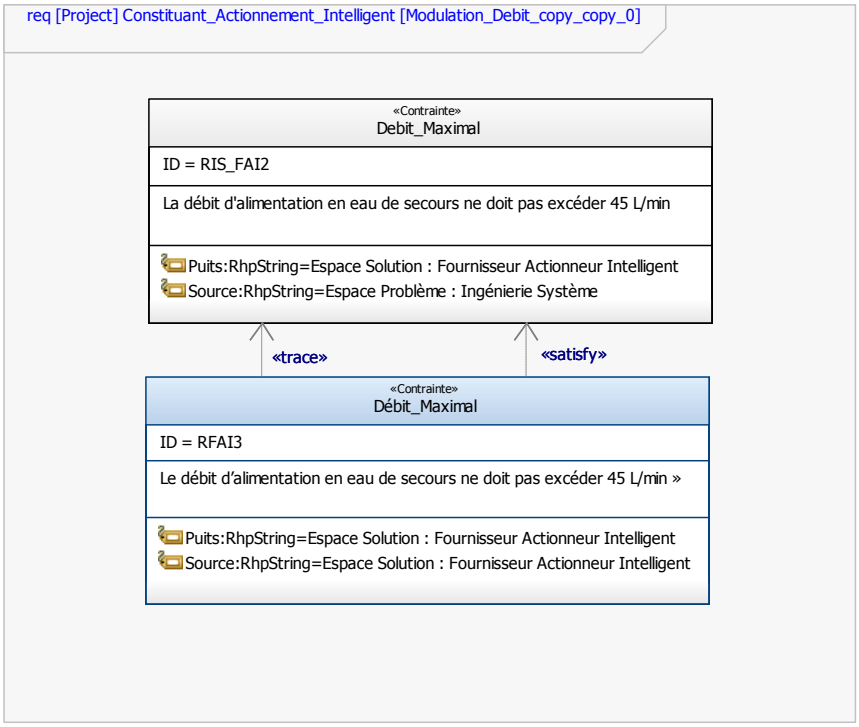


Figure 214 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence système « R_{IS_FAI2}» par l'espace-solution fournisseur d'actionneur intelligent

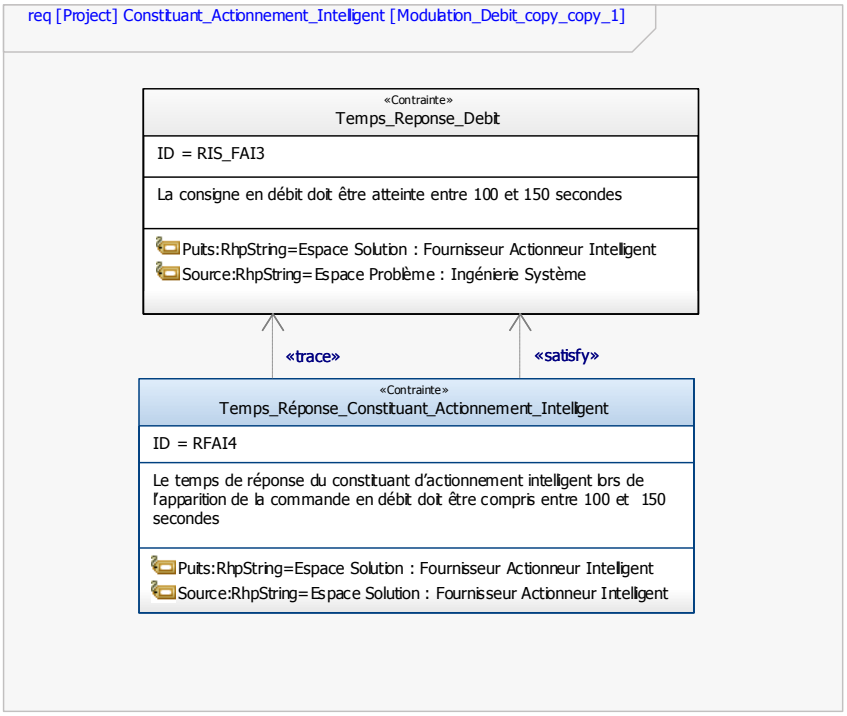


Figure 215 : Diagramme d'exigences (req) SysML - Raffinement de l'exigence système « R_{IS_FAI3}» par l'espace-solution fournisseur d'actionneur intelligent

Annexe 7-2 : Spécification du constituant d'actionnement intelligent de modulation de débit

```
function
[rDefautActionner,rP_V1,obQv_V1]
= fcn(obP_V1,rQv_V1,rDefaut)
%#codegen
PrAtm = 101325;
%Pression Atmosphérique en Pascal
obQv_V1 = obP_V1/PrAtm*60000;   %
Objectif Debit (L/min)
rP_V1= rQv_V1*PrAtm/60000;    %
Rapport Puissance(W)
rDefautActionner = rDefaut;
```

Interaction Logico-Technologique

```
function [Defaut_Act_V1,obvQv_V1]  = Acter(obQv_V1)
%#codegen
Debit_Min = 0;
Debit_Max = 50;
% Test Valeur Objectif Debit

if ((obQv_V1 < Debit_Min)||(obQv_V1 > Debit_Max))
Defaut_Act_V1 = true;
obvQv_V1 = 0;
else
Defaut_Act_V1  = false;
obvQv_V1 = obQv_V1;
end
```

Acter

```
function [ObvQv_V1, Defaut_Obs_V1] =
filtrage_Obs(ObQv_V1)
%#codegen
Debit_Min = 0;
Debit_Max = 75;
if (ObQv_V1 < Debit_Min) || (ObQv_V1 >
Debit_Max)
ObvQv_V1 = 0;
Defaut_Obs_V1 = true;
else
ObvQv_V1 = ObQv_V1;
Defaut_Obs_V1 = false;
end
```

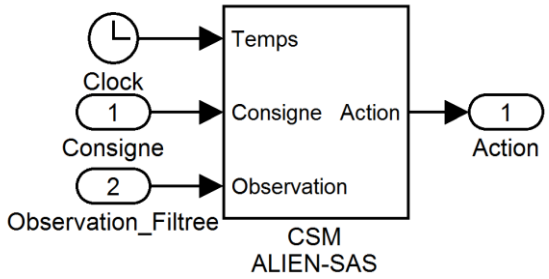
Percevoir

```
function
[Defaut_Ag_V1,Action_Pos_V1] =
Agir(Pos_V1)
%#codegen

Pos_Max = 1;
Pos_Min = 0;

if (Pos_V1 <= Pos_Min) || (Pos_V1
>= Pos_Max)
Action_Pos_V1 = 0;
Defaut_Ag_V1 = true;
else

Action_Pos_V1 = Pos_V1;

Defaut_Ag_V1 = false;
end
```

Agir

```
function [rDefaut,rQv_v1] =
Rendre_Compte(Defaut_Act_V1,Defaut_Ag_V1,ObsQv_V1,Defaut_Obs_V1)
%#codegen
if ((Defaut_Act_V1 == true)&&(Defaut_Ag_V1 ==
false)&&(Defaut_Ag_V1 == false))

rDefaut= 1 %'Defaut Acter - Erreur Action:Position';
rQv_v1=0;
end
if ((Defaut_Ag_V1 == true)&&(Defaut_Act_V1 ==
false)&&(Defaut_Ag_V1 == false))
rDefaut= 2 %'Defaut Agir - Erreur Objectif:Debit';
rQv_v1=0;
end
if (Defaut_Obs_V1 == true)&&((Defaut_Act_V1 ==
false)&&(Defaut_Ag_V1 == false))
rDefaut= 3 %'Defaut Percevoir - Erreur Observation:Debit';
rQv_v1=0;
end
if ((Defaut_Act_V1 == false)||(Defaut_Ag_V1 ==
false)||(Defaut_Ag_V1 == false))
rDefaut= 0 %'Pas de Defaut - Regulation Puissance OK';
rQv_v1=ObsQv_V1;
end
```

Rendre-Compte



Actionner

Tableau 24 : Spécification du comportement des blocs : « Interaction Logico-Technologique », « Acter », « Percevoir », « Agir », « Rendre-Compte » et « Actionner » composant le constituant d'actionnement intelligent de modulation de débit

Annexe 7-3 : Environnement de co-simulation pour la vérification de la spécification (S_{FAI}) du constituant d'actionnement intelligent

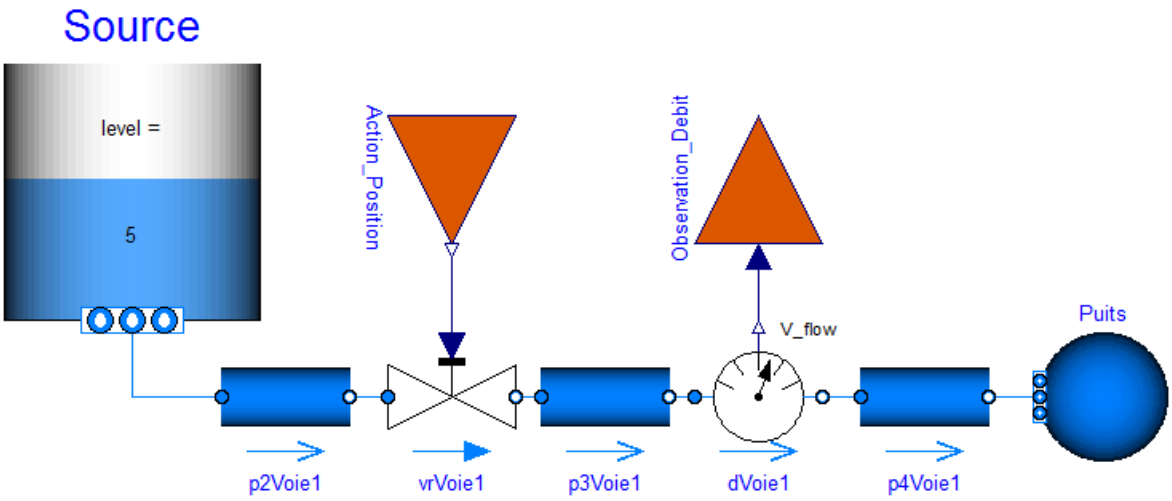


Figure 216 : Modèle Modelica de la partie technique du constituant d'actionnement intelligent pour la vérification de la spécification (S_{FAI}) du constituant d'actionnement intelligent

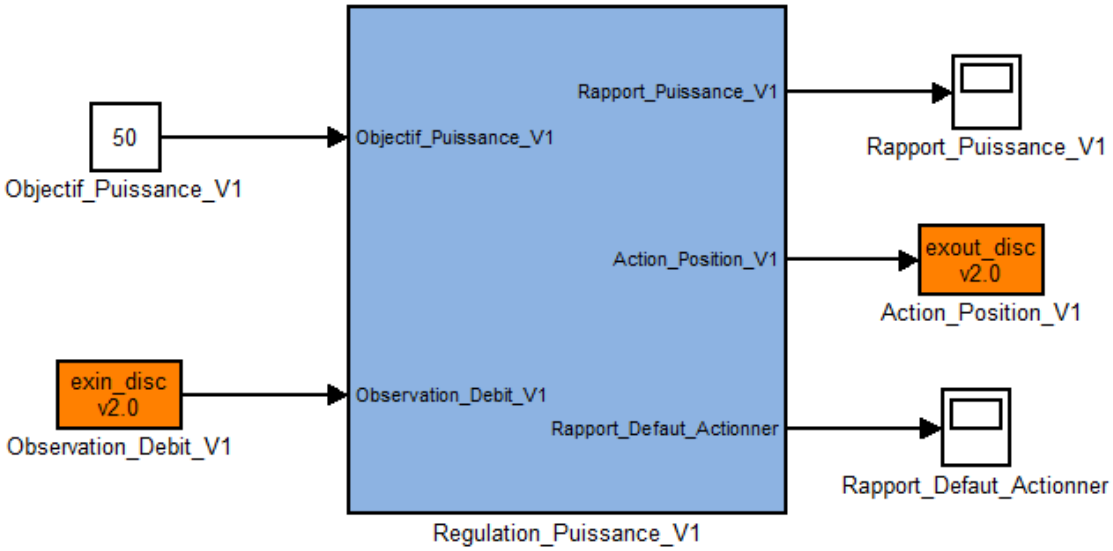


Figure 217 : Modèle Simulink de la partie logique du constituant d'actionnement intelligent pour la vérification de la spécification (S_{FAI}) du constituant d'actionnement intelligent

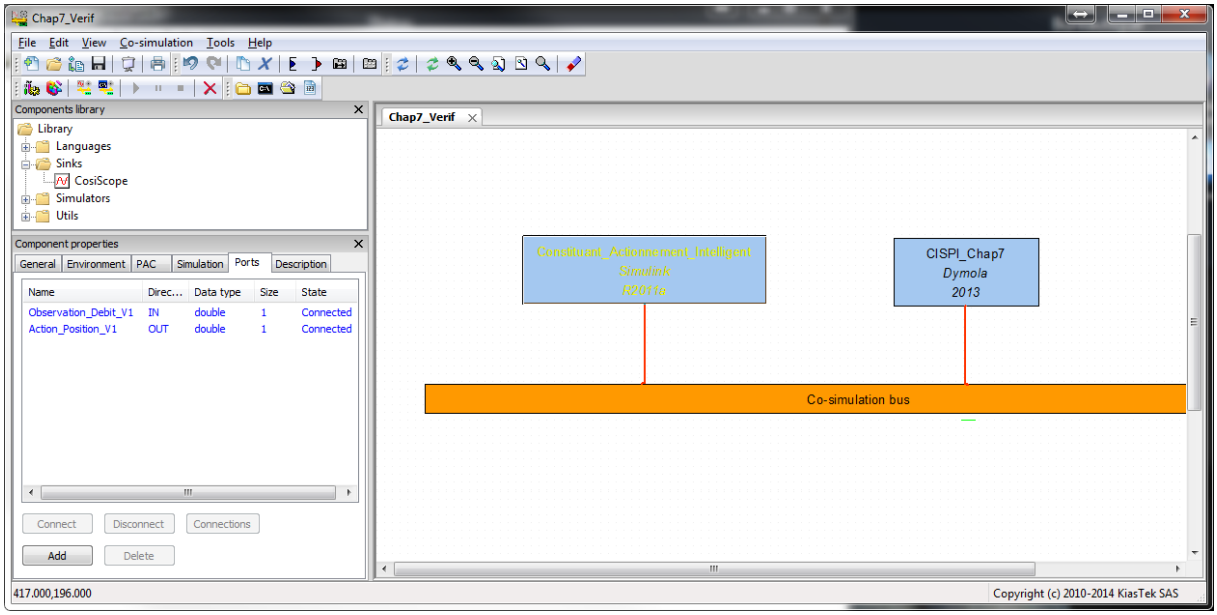


Figure 218 : Configuration de l'environnement de co-simulation pour la vérification du modèle du constituant d'actionnement intelligent de modulation de débit

Annexe 7-4 : Environnement de co-simulation pour la validation de la spécification (S_{FAI}) du constituant d'actionnement intelligent

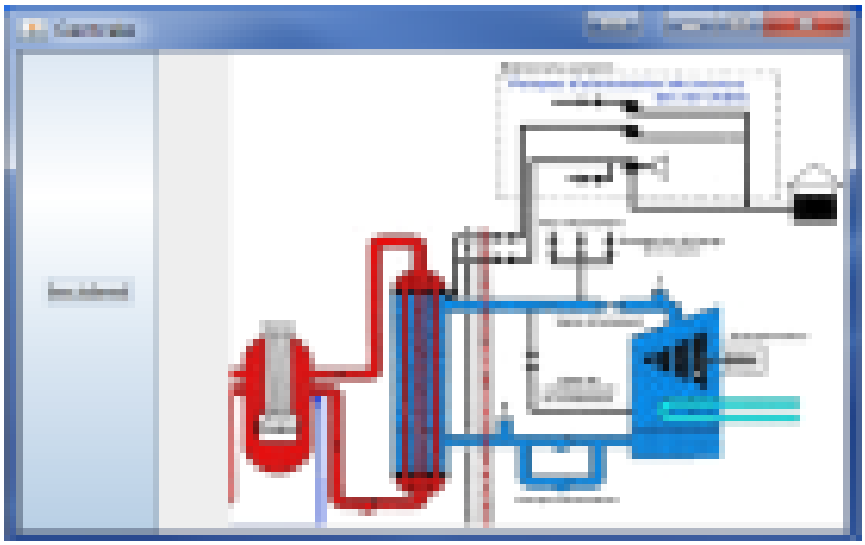


Figure 219 : Modèle de l'environnement de la plate-forme CISPI (circuit primaire) pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit

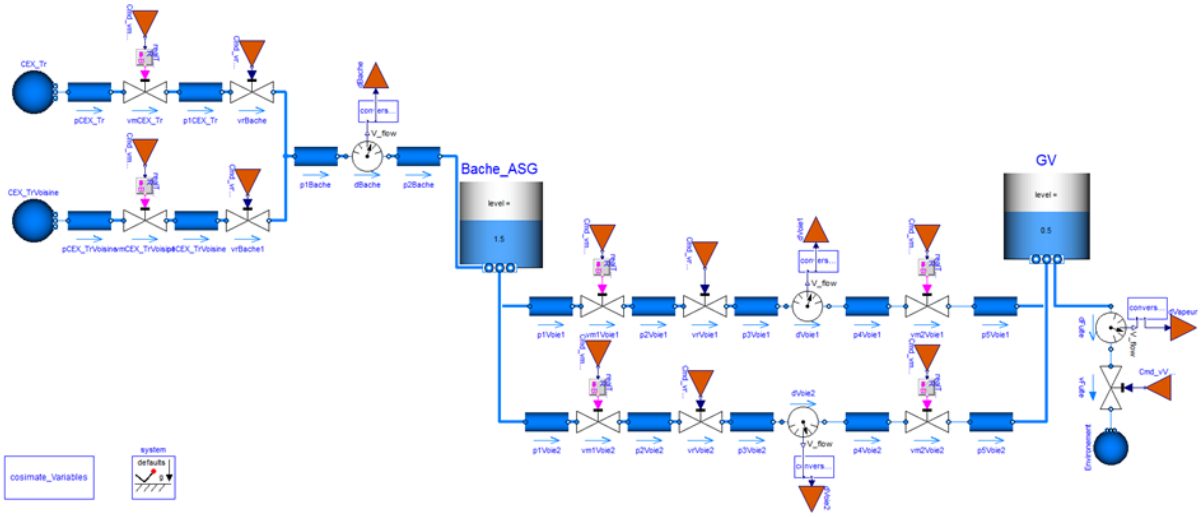


Figure 220 : Modèle du sous-système physique de la plate-forme CISPI (*incluant la partie technique du constituant d'actionnement intelligent*) avec des ports de type "Cosimate" pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit

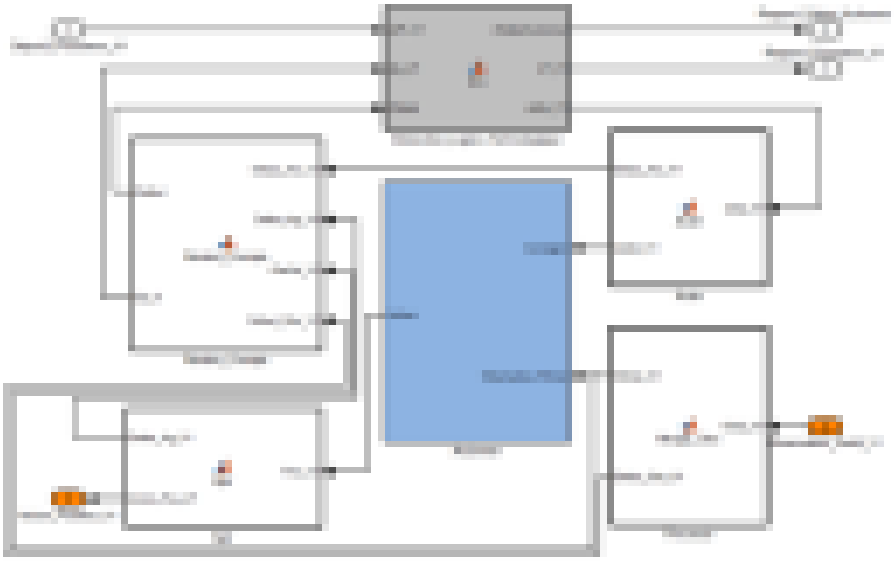


Figure 221 : Modèle de la partie logique du constituant d'actionnement intelligent pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit

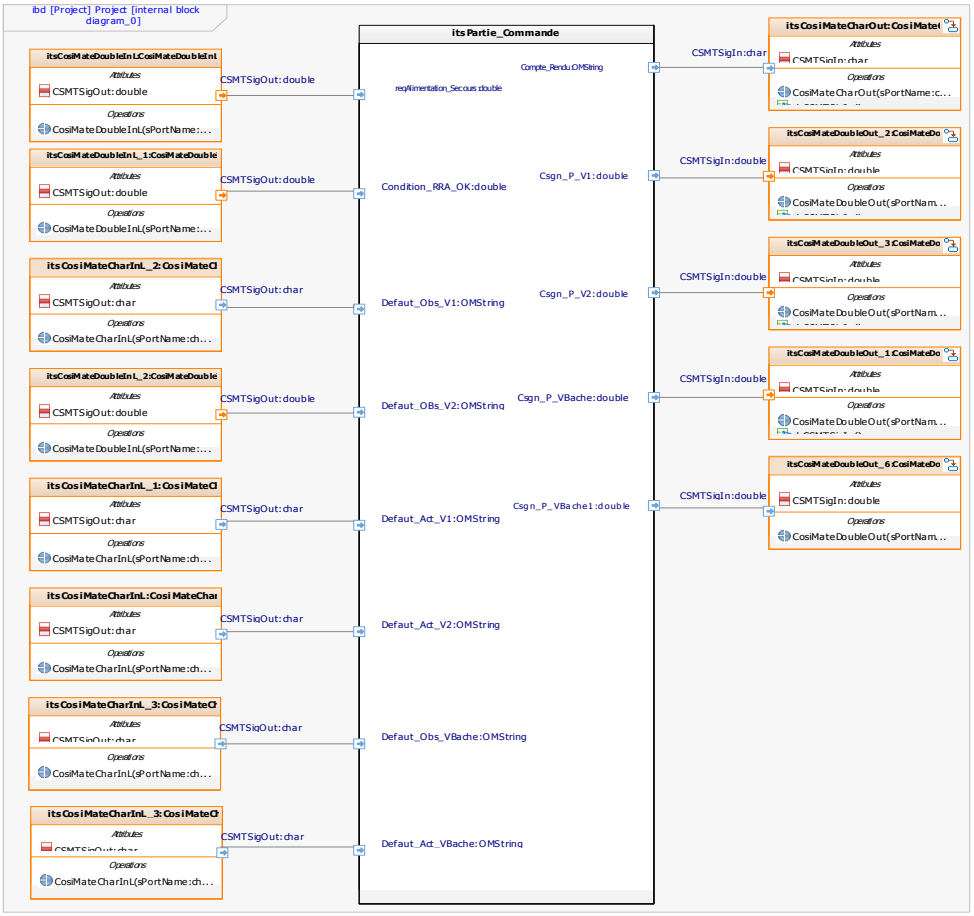


Figure 222 : Modèle de la partie commande du sous-système logique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit

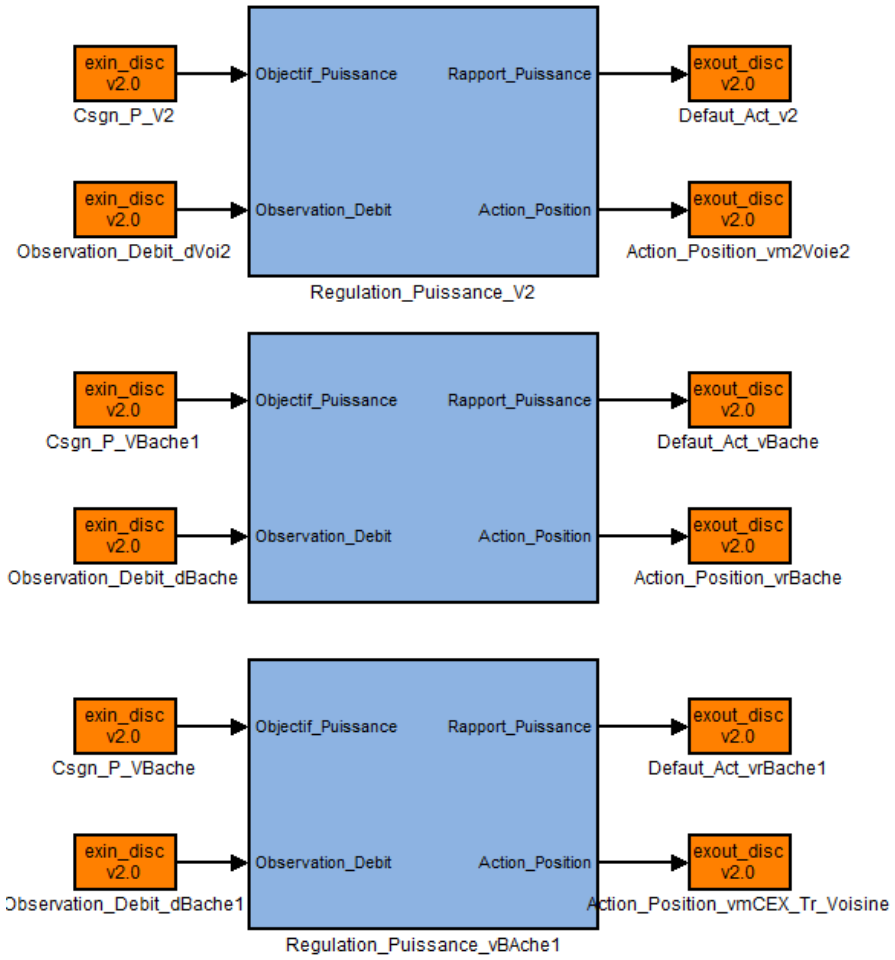


Figure 223 : Modèle de la partie contrôle du sous-système logique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit

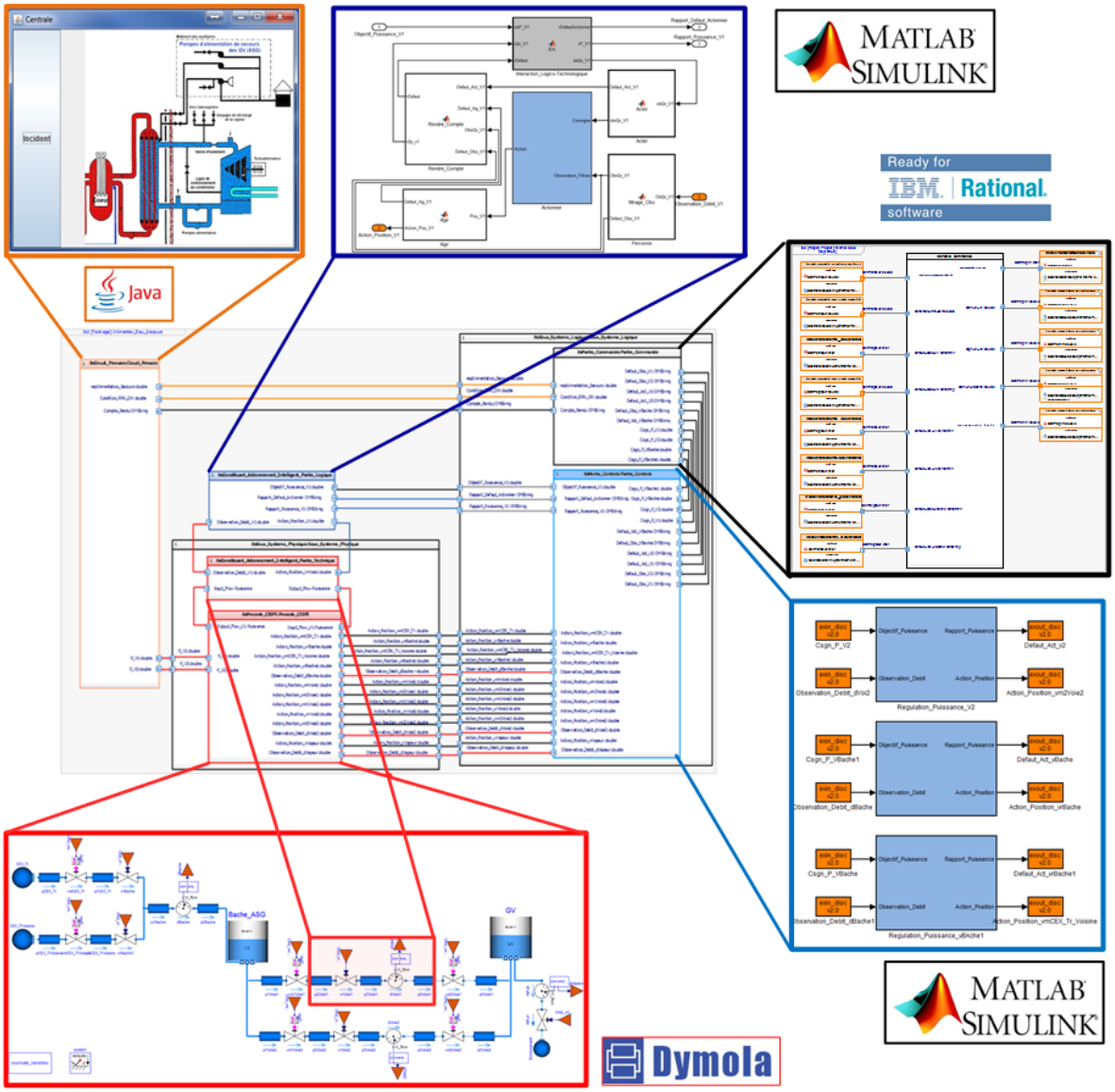


Figure 224 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle du constituant d'actionnement intelligent de modulation de débit

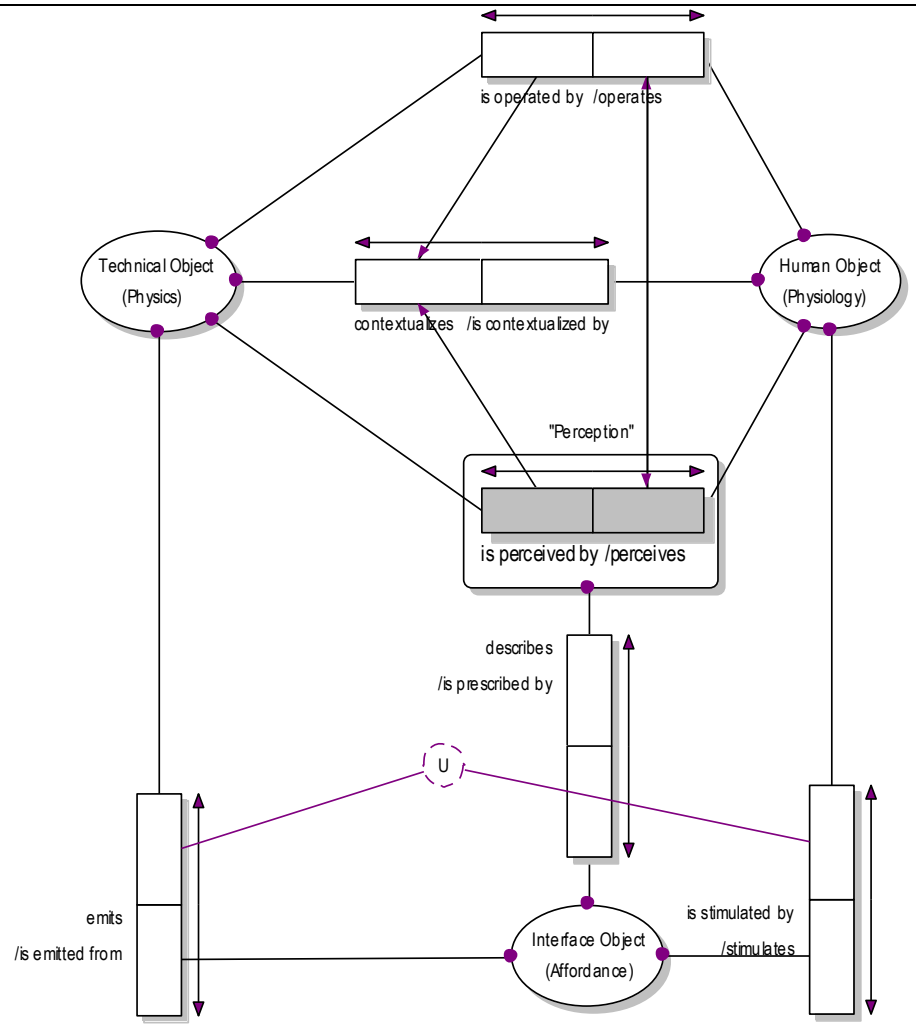
Annexe 7-5 : Module de couplage Serveur OPC / Bus de co-simulation (Cosimate)

```
import com.cosimate.coupling.java.*;
import java.io.IOException;
import java.net.UnknownHostException;
import java.util.concurrent.ScheduledThreadPoolExecutor;
import java.util.logging.Level;
import java.util.logging.LogManager;
import java.util.logging.Logger;
import org.openscada.opc.lib.common.ConnectionInformation;
import org.openscada.opc.lib.da.Server;
import org.jinterop.dcom.common.JIException;
import org.jinterop.dcom.core.JIVariant;
import org.openscada.opc.lib.common.AlreadyConnectedException;
import org.openscada.opc.lib.common.NotConnectedException;
import org.openscada.opc.lib.da.AddFailedException;
import org.openscada.opc.lib.da.DuplicateGroupException;
import org.openscada.opc.lib.da.Group;
import org.openscada.opc.lib.da.Item;
import org.openscada.opc.lib.da.ItemState;
import org.openscada.opc.lib.da.ServerConnectionStateListener;

/**
 * @author Remi PANNEQUIN
 *         Fabien BOUFFARON
 */
public class Station1 {

   public enum ConnectionState {disconnected, connected,};
   static final String user = "cispi";
   static final String pwd = "cispi";
   static final String ip = "193.50.39.229";//"10.100.2.1";
   static final String clsid = "B6EACB30-42D5-11d0-9517-0020AFAA4B3C";
   static final String GROUP_NAME = "CosiMate";
   static final String ITEM_NAME_MEASURE = "Automate Centrale.CPU.DB_Com.S1_debit_mesure";
   static final String ITEM_NAME_SETPOINT = "Automate Centrale.CPU.DB_Com.S1_ordre_debit";

   private static ConnectionState connect = ConnectionState.disconnected;
   private static Item item_measure, item_setpoint;

   private static void initOPC() {
      try {
         ConnectionInformation connectionInformation = new ConnectionInformation(user, pwd);
         connectionInformation.setHost(ip);
         connectionInformation.setClsid(clsid);

         final Server s = new Server(connectionInformation, new ScheduledThreadPoolExecutor(10));
         s.addStateListener(new ServerConnectionStateListener() {

            @Override
            public void connectionStateChanged(boolean c) {
               if (c) {
                  try {
                     connect = ConnectionState.connected;
                     Group group = s.addGroup(GROUP_NAME);
                     item_measure = group.addItem(ITEM_NAME_MEASURE);
                     item_setpoint = group.addItem(ITEM_NAME_SETPOINT);
                  } catch (UnknownHostException ex) {
                     Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
                  } catch (DuplicateGroupException ex) {
                     Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
                  } catch (AddFailedException ex) {
                     Logger.getLogger(Station1.class.getName()).log(Level.WARNING, "failed to add item " + s, ex);
                  } catch (NotConnectedException ex) {
                     Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
                  } catch (IllegalArgumentException ex) {
                     Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
                  } catch (JIException ex) {
                     Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
```

```
                }

            } else {
                connect = ConnectionState.disconnected;
            }
        }
    });
    s.connect();
  } catch (IllegalArgumentException ex) {
    Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
  } catch (UnknownHostException ex) {
    Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
  } catch (JIException ex) {
    Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
  } catch (AlreadyConnectedException ex) {
    Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
  }
}

public static void main(String[] args) {
    initOPC();
    //wait for connection to OPC
    while (connect != ConnectionState.connected) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException ex) {
            Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    double dPort1;
    double dPort2=30;
    boolean kill = false;

    // CosiMate bus initialization
    CCSMTBus cBus = new CCSMTBus("CispiStation1", 1.0);

    // CosiMate ports declaration
    CCSMTDoubleOut cPort1 = new CCSMTDoubleOut("FlowRate1");
    CCSMTDoubleIn cPort2 = new CCSMTDoubleIn("FlowSetPoint1",dPort2);

    // Stop if the program is running out of the co-simulation environment
    cBus.StopOnStandAloneMode();

    // End of the initialization phase
    cBus.EndInit();

    // Computation loop

    while (!kill) {
        try {
            // Compute and send Port1
            ItemState state = item_measure.read(true);
            dPort1 = (Float) state.getValue().getObject();
            cPort1.SetData(dPort1);

            // Synchronize
            int r = cBus.Synchronize(-1.0);

            switch (r) {
                case 0:
                case 1:
                    break;
                case 2:
                    System.out.println("Simulation stopped");
                    kill = true;
                    break;
                default:
                    System.out.println("Synchronization error");
                    kill = true;
```

```
            }
        } catch (JIException ex) {
            Logger.getLogger(Station1.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    // End of the co-simulation
    cBus.Stop();
  }
}
```

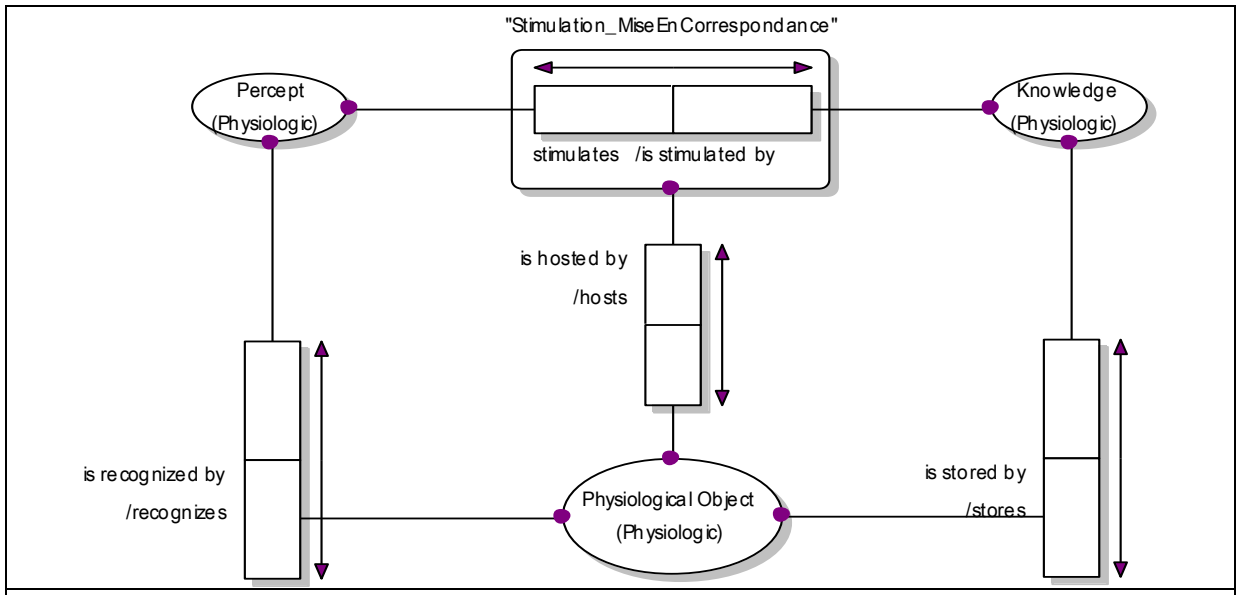
Annexe 8-1 : Modèles NIAM-ORM d'une interaction Homme-Machine (Lieber, 2013)



(F1): It is possible that some Perception describes more than one Interface Object and that some Interface Object is prescribed by more than one Perception.
(F2): Every Interface Object is identified by one distinct Affordance.
(F3): It is possible that some Interface Object is emitted from more than one Technical Object and that some Technical Object emits more than one Interface Object.
(F4): It is possible that some Interface Object stimulates more than one Human Object and that some Human Object is stimulated by more than one Interface Object.

(F5) : For each Human Object (Field Operator) and Technical Object (Sound Alarm) there is at most one Interface Object that stimulates Human Object and is emitted from Technical Object **(Uniqueness_Constraint1)**

Figure 225 : Modèle ORM de l'interaction de perception pour agir en contexte. L'objet interface affordant (F1 & F2) au centre de la relation de perception (F3 &F4) représente l'élément synergique entre objets hétérogènes techniques et humains (F5) qu'il faut arrivé à mesurer (Cette nécessité de mesure est représentée par l'affordance dans l'objet interface)



(F6): It is possible that some Percept stimulates more than one Knowledge and that some Knowledge is stimulated by more than one Percept.

(F7): Stimulation_MiseEnCorrespondance is an entity object type; Stimulation_MiseEnCorrespondance is an alias for the nested fact type 'Percept stimulates Knowledge' **(Nested_Fact3)**

(F8): It is possible that some Stimulation_MiseEnCorrespondance is hosted by more than one Physiological Object and that some Physiological Object hosts more than one Stimulation_MiseEnCorrespondance.

(F9): It is possible that some Physiological Object stores more than one Knowledge and that some Knowledge is stored by more than one Physiological Object.

(F10): It is possible that some Physiological Object recognizes more than one Percept and that some Percept is recognized by more than one Physiological Object.

Example: Sound alarm detection stimulates knowledge (Incident-Accident Procedures)

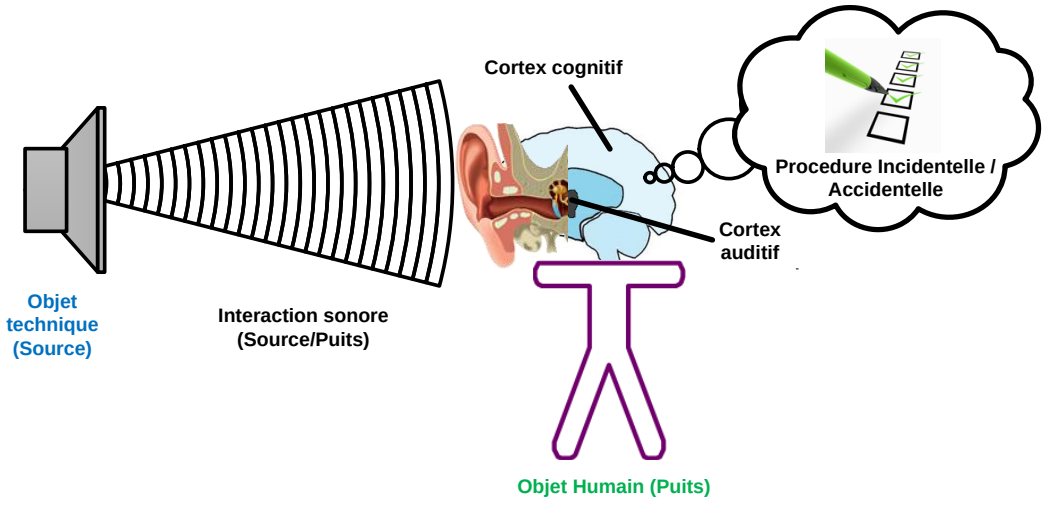
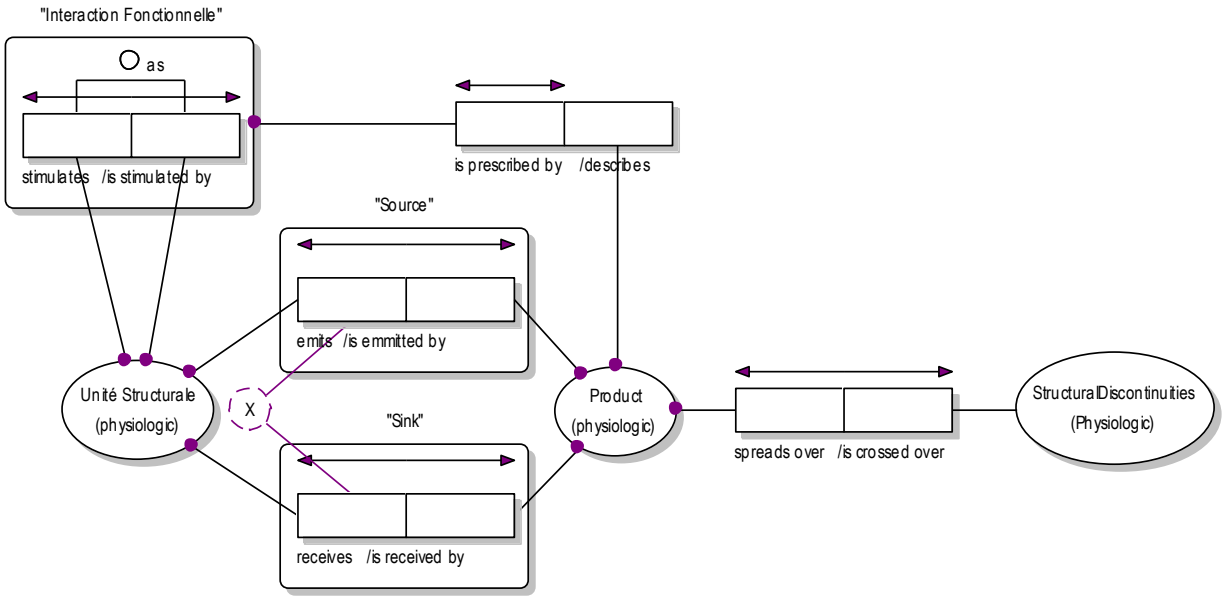


Figure 226 : Modèle ORM de l'interaction de détection sonore entre un objet technique et un opérateur humain, envisagée comme la mise en correspondance d'un percept, son de l'alarme, et d'une propriété cognitive ou connaissance de l'opérateur (Procédure incidentelle ou accidentelle)



(F11): It is possible that some Unité Structurale stimulates more than one Unité Structurale and that some Unité Structurale is stimulated by more than one Unité Structurale.

(F12): Interaction Fonctionnelle is an alias for the nested fact type 'Unité Structurale stimulates Unité Structurale'

(F13): If Unité Structurale u1 stimulates Unité Structurale u2 then it cannot be that Unité Structurale u2 stimulates Unité Structurale u1. **(Ring constraint)**

(F14): It is possible that some Product spreads over more than one StructuralDiscontinuities and that some StructuralDiscontinuities is crossed over more than one Product.

(F15): It is possible that some Unité Structurale emits more than one Product and that some Product is emitted by more than one Unité Structurale.

(F16): Source is an alias for the nested fact type 'Unité Structurale emits Product'

(F17): It is possible that some Unité Structurale receives more than one Product and that some Product is received by more than one Unité Structurale.

(F18): Sink is an alias for the nested fact type 'Unité Structurale receives Product'

Figure 227 : Modèle ORM de l'atome élémentaire d'interaction fonctionnelle d'un processus physiologique. L'interaction fonctionnelle émerge de la relation de stimulation entre deux unités structurales et est prescrite in fine par le produit émis par la source u1 ou reçu par le puis u2 qui traverse des discontinuités structurales. De telles propriétés des mécanismes physiologiques conduisent à une structuration de l'organisme vivant en niveaux d'organisation hiérarchique et de ses unités structurales (architecture organique), ainsi que de ces processus physiologiques (architecture fonctionnelle)

Annexe 8-2 : Specification prescriptive d'exigences physiologiques

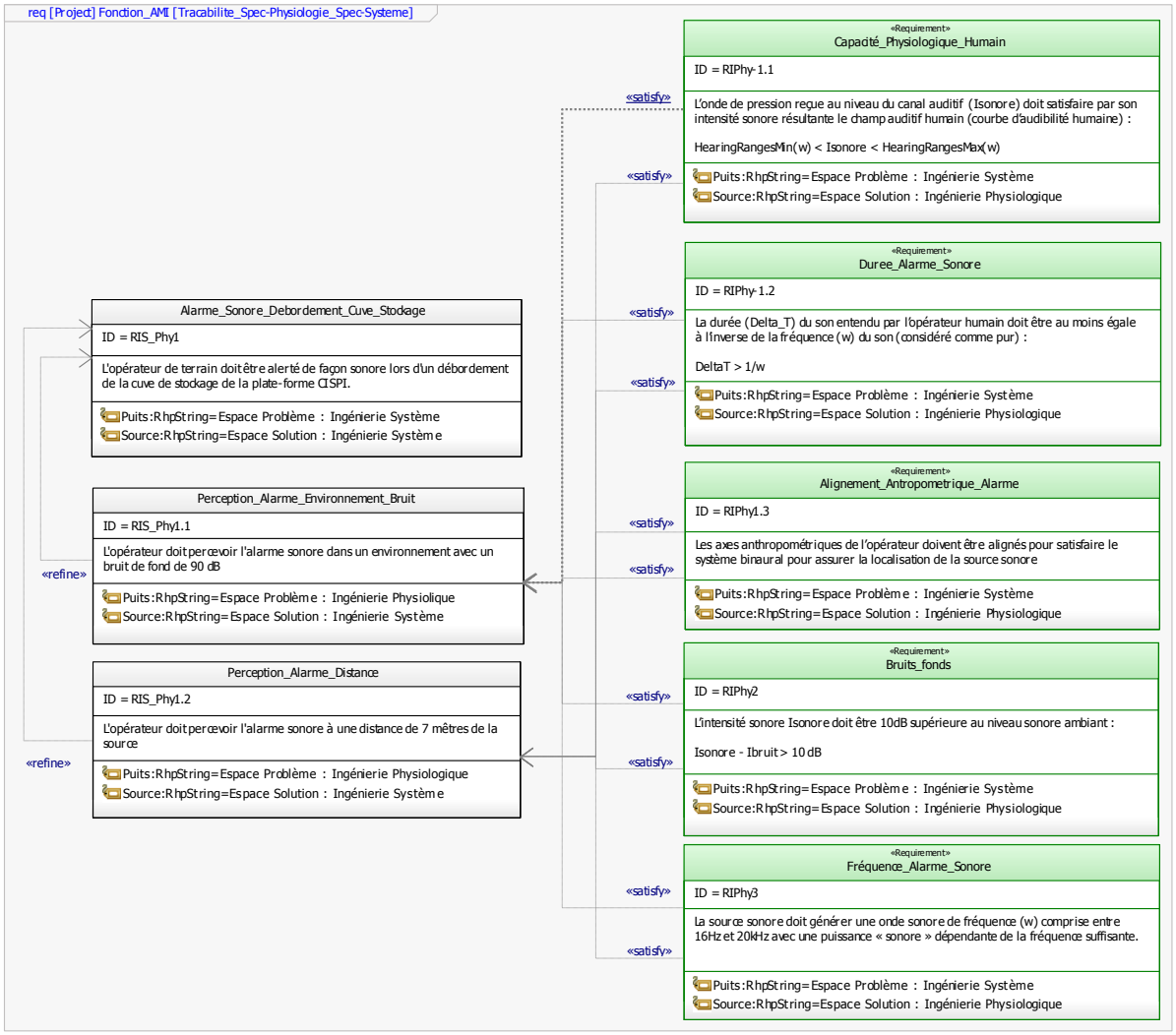


Figure 228 : Traçabilité entre exigences physiologiques (R_{IPhy}) et exigences systèmes (R_{IS})

Annexe 8-3 : Modèle Simulink de l'interaction physico-physiologique de détection sonore

Fonction Perception (*Agent humain de détection sonore*)

```
function Perception =
fcnPerception(Frequence_Sonore,Puissance_Physiologique,Bruit_Fonds)

%#codegen
ValueMax=1000;
ValueMin=1000;
Pa=10^(-5);

if (Puissance_Physiologique == 0)
    IdB = 0;
else
    IdB=10*log10(complex(Puissance_Physiologique/Pa^2))
end

coder.extrinsic('spline');

Xfmin=[16;20;30;40;50;60;70;80;90;100;200;300;400;500;600;700;800;900;1000;
2000;3000;4000;5000;6000;7000;8000;9000;10000;14000;15000;19000;20000];
Ymin=[120;80;63;55;50;45;42;40;38;35;22;15;11;9;6;4;3;2;1;-5;-10;-
9;0;3;8;10;12;12;15;20;40;70];

ValueMin = spline(Xfmin,Ymin,Frequence_Sonore)

Xfmax=[16;20;30;1000;2000;3000;4000;5000;7000;8000;14000;20000];
Ymax=[120;127;130;130;122;115;112;115;120;122;110;70];

ValueMax = spline(Xfmax,Ymax,Frequence_Sonore)


if (ValueMin<IdB<ValueMax)&&(IdB-Bruit_Fonds>10)
    Perception=1;
else
    Perception=0;
end
```

Fonction Amplitude (*Interaction fonctionnelle*)

```
function Amplitude  = fcnAmplitude(Puissance_Source,X_Source,
Y_Source,rho,c, X_Puits, Y_Puits)
%#codegen


Distance = sqrt ((X_Puits-X_Source)^2 + (Y_Puits-Y_Source)^2); %# Calcul de
la distance entre le Puits et la source

Amplitude = (rho*c*Puissance_Source)/(4*pi*Distance*Distance);
```

Annexe 8-4 : Environnement de co-simulation pour la validation du sous-système d'interaction de détection sonore de la plate-forme CISPI

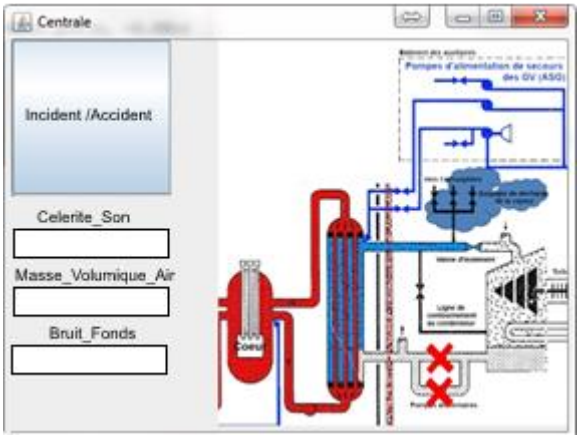


Figure 229 : Modèle de l'environnement de la plate-forme CISPI (circuit primaire) pour la validation du modèle d'interaction de détection sonore

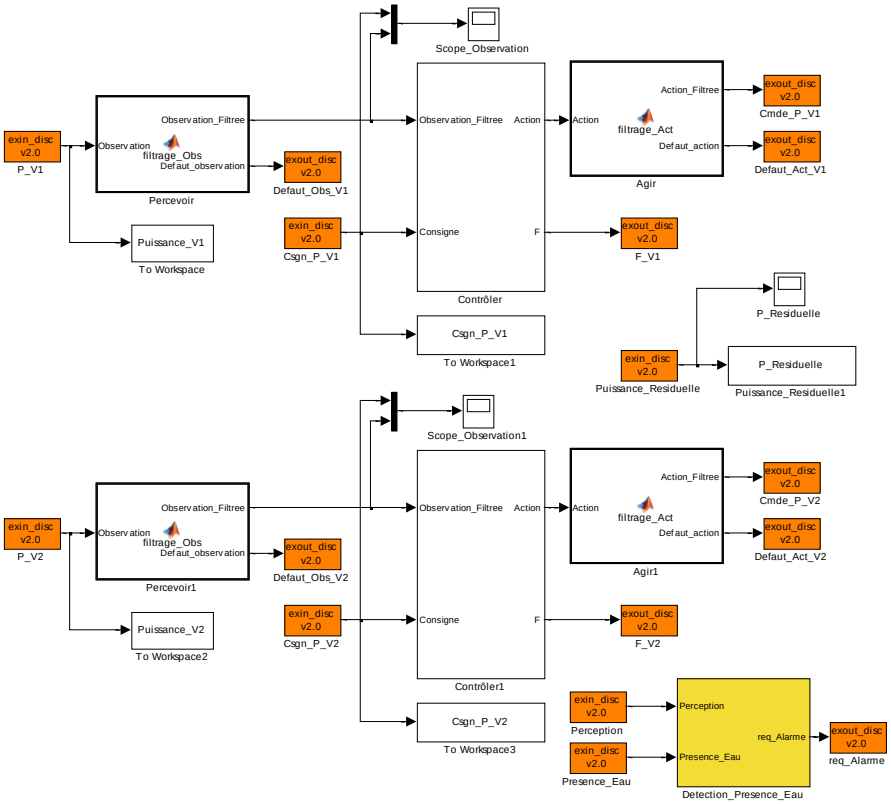


Figure 230 : Modèle de la partie contrôle du sous-système logique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle d'interaction de détection sonore

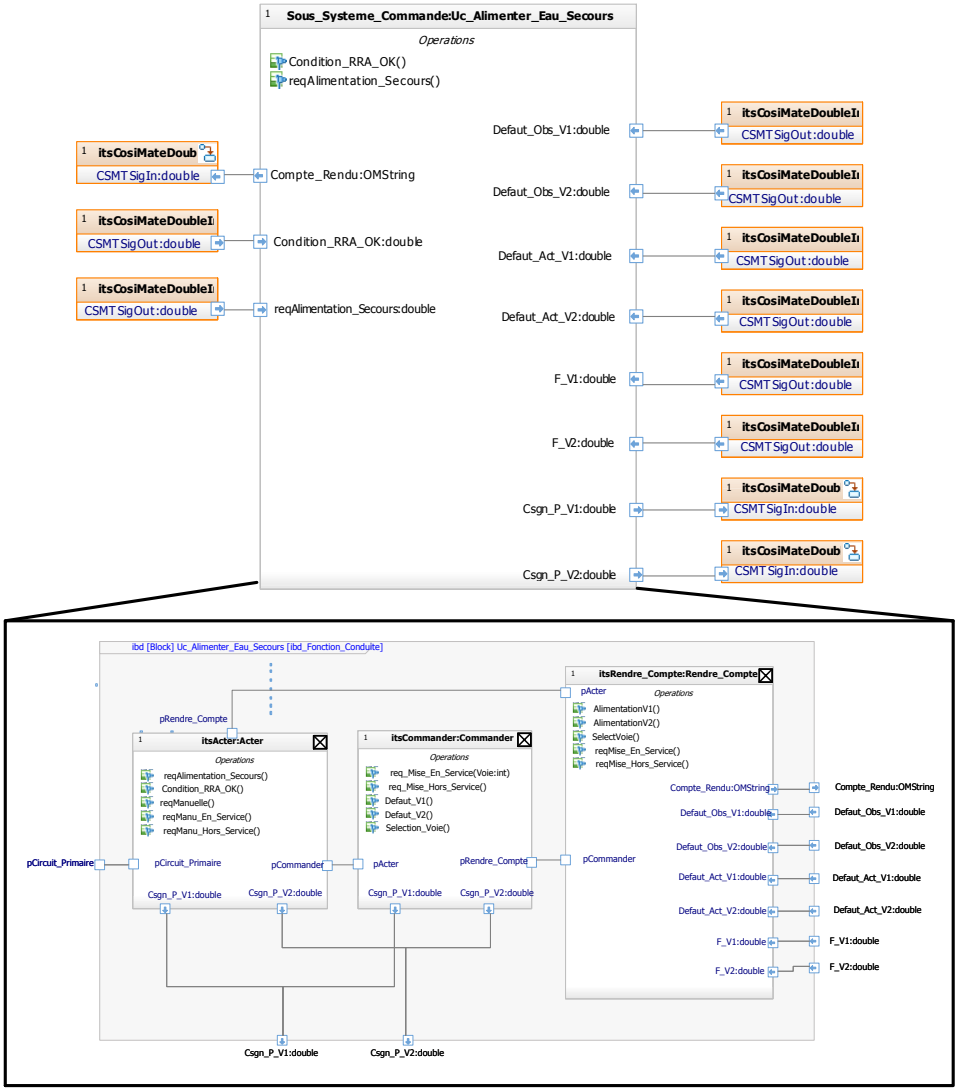


Figure 231 : Modèle de la partie commande du sous-système logique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle d'interaction de détection sonore

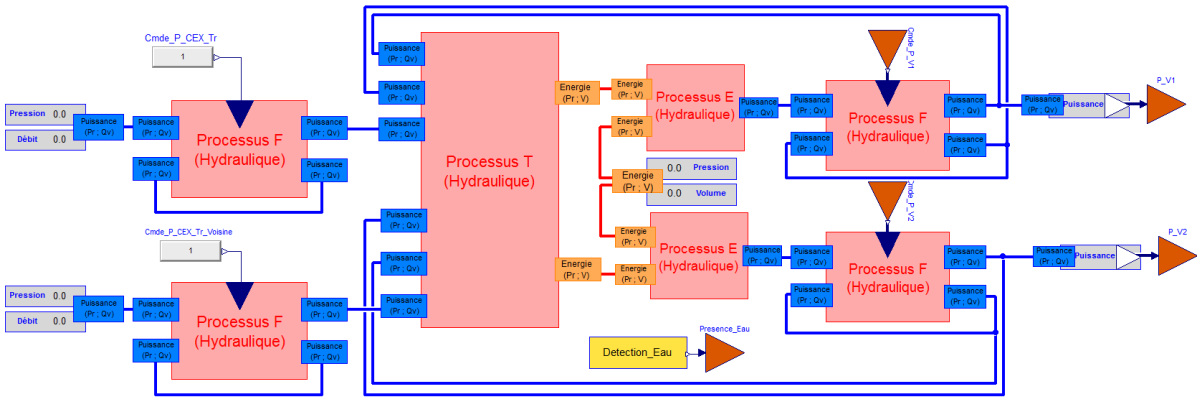


Figure 232 : Modèle du sous-système physique de la plate-forme CISPI avec des ports de type "Cosimate" pour la validation du modèle d'interaction de détection sonore

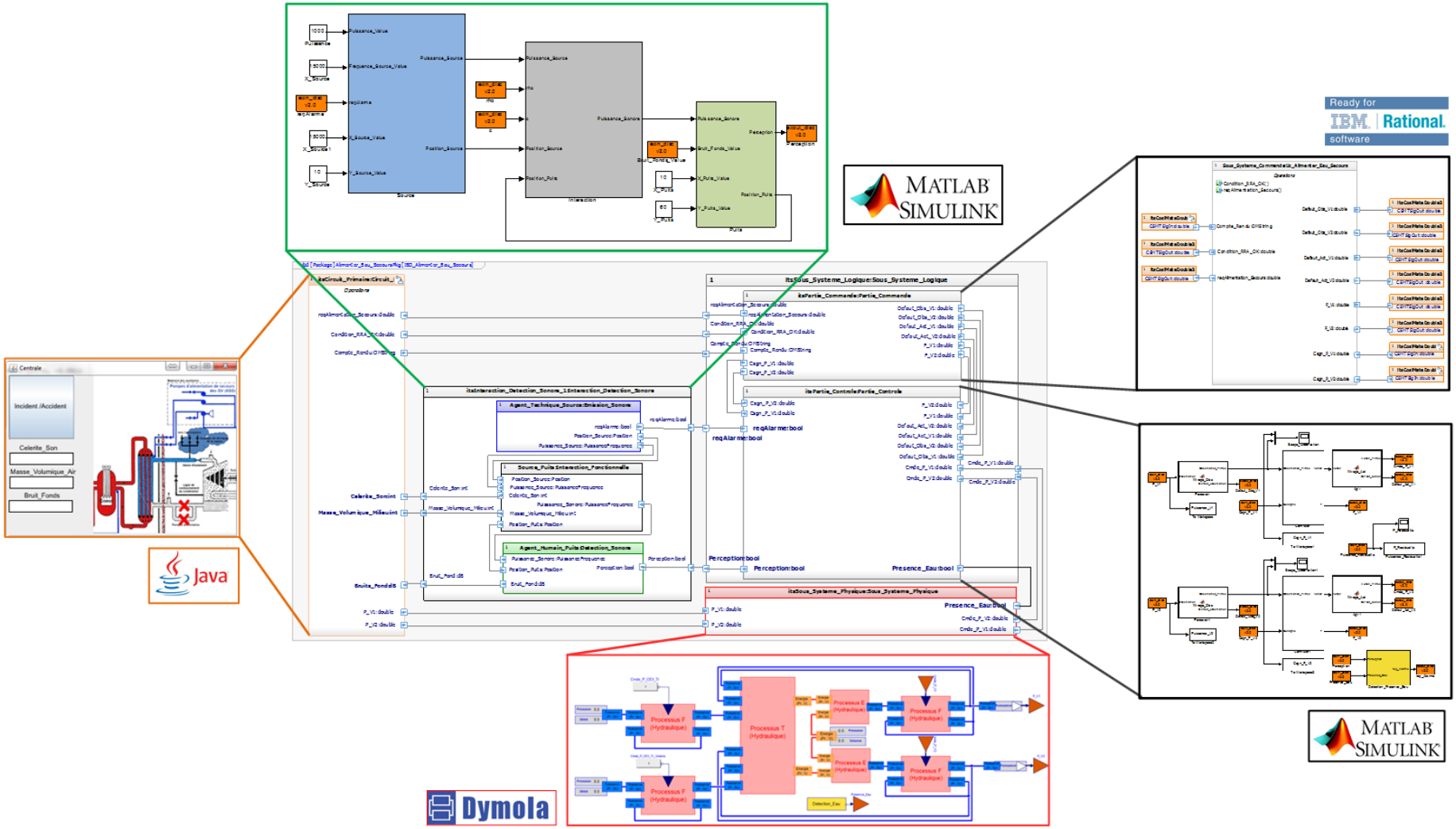


Figure 233 : Diagramme interne de blocs SysML : Architecture des modèles d'ingénierie spécialistes pour la validation du modèle d'interaction de détection sonore de la plate-forme CISPI

Résumé

Dans la mesure où un système est un ensemble d'éléments en interaction dont il faut gérer la complexité, la difficulté pour un architecte système est de guider l'architecture d'un modèle « total » de la situation système de conduite étudiée en tant qu'ensemble de modèles « locaux » d'ingénieries interdisciplinaires en interaction.
Les travaux présentés dans ce mémoire s'intéressent plus précisément à la nature heuristique, spécifiante et exécutable de cette relation « totale » de couplage de modèles descriptifs de problèmes ($\boldsymbol{R_H}$) et prescriptifs de solutions ($\boldsymbol{M_h}$) pour construire un modèle virtuel ($\boldsymbol{M_H}$) du système à faire. La perspective holonique retenue permet de considérer cette relation de couplage de façon descriptive (***H***) du TOUT et prescriptive (***h***) de chacune des PARTIES aussi bien en regard de la situation-système à percevoir que des constitutifs-système à architecturer.

En ce sens, notre environnement collaboratif de co-spécification système vise à supporter un processus itératif basé sur la requête d'éléments de connaissances ($\boldsymbol{K_h}$) auprès de chacune des ingénieries spécialistes délivrant en retour les modèles constitutifs ($\boldsymbol{M_h}$) satisfaisant des exigences élémentaires ($\boldsymbol{R_h}$). Cet environnement de co-modélisation-système est lui-même constitué d'un ensemble d'environnements élémentaires de modélisation de constituants-système, avec pour objectif assigné de préserver les outils, méthodes et processus de travail de chacune des parties prenantes afin de faciliter l'expression de leurs connaissances métier. Le modèle constituant ($\boldsymbol{M_H}$) vise à refléter la satisfaction de la mission ($\boldsymbol{R_H}$) du système requis dans le contexte connu ($\boldsymbol{K_H}$) d'une situation-système d'intérêt, limitée pour notre cas d'étude au facteur d'échelle de la plate-forme d'expérimentation CISPI (Conduite Interactive et Sûre de Procédé Industriel) du projet SAFETECH du CRAN. La technologie de modélisation et de simulation mise en œuvre s'appuie sur le langage de modélisation système « SysML » pour le partage de connaissances et permet la vérification ainsi que la validation système de l'ensemble par co-exécution de modèles autour d'un bus de co-simulation, y compris in-situ avec la plate-forme CISPI.

Abstract

Insofar as a system is a set of interacting elements whose complexity must be managed, the difficulty for a system architect is to guide the whole model architecture of a control system-situation studied as a set of interdisciplinary engineering part models interacting. The work presented in this thesis are specifically interested in the heuristic, specifying and executable nature of this whole relationship coupling descriptive models of problems ($\boldsymbol{R_H}$) and prescriptive models of solutions ($\boldsymbol{M_h}$) to design a virtual model ($\boldsymbol{M_H}$) of system-of-interest. The holonic perspectives allows us to consider this coupling relationship as descriptive of a WHOLE (***H***) and prescriptive of each parts (***h***) as well in regards to system situation to perceive, as system-elements to architect.

In this sense, our collaborative environment of system co-specification aims to support an iterative process based on the quest of knowledge elements ($\boldsymbol{K_h}$) with each specialist engineering delivering constitutive models ($\boldsymbol{M_h}$) satisfying basic requirements ($\boldsymbol{R_h}$). This system co-modelling environment is itself composed of a set of system-components modelling environment, with the stated objective to preserve tools, methods and works of each stakeholders in order to facilitate the expression of their skills. The model ($\boldsymbol{M_H}$) aims to reflect the mission satisfaction of the required system ($\boldsymbol{R_H}$) in system-situation of interest ($\boldsymbol{K_H}$) limited for our case-study to the scale factor of the experimental platform CISPI (Safe and Interactive Control of Industrial Process) of CRAN SAFETECH project. Implementation of modelling and simulation technology is based on the system modelling language (SysML) for knowledge sharing. Moreover, it allows the WHOLE system verification and validation by co-execution of models around a co-simulation bus, including CISPI platform.