Towards decision support for complex system architecture design with innovation integration in early design stages

Marie-Lise Moullec

**ÉCOLE CENTRALE DES ARTS
ET MANUFACTURES
« ÉCOLE CENTRALE PARIS »**

THÈSE

présentée par

Marie-Lise MOULLEC

pour l'obtention du

GRADE DE DOCTEUR

Spécialité : Génie Industriel

Laboratoire d'accueil: Laboratoire de Génie Industriel

SUJET :

Towards decision support for complex system architecture design with innovation integration in early design stages

Vers une méthode d'aide à la décision pour l'intégration d'innovations dès la conception préliminaire des architectures de systèmes complexes

soutenue le : 24 janvier 2014

devant un jury composé de :

Jean-Claude Bocquet, Professor, Ecole Centrale Paris	Supervisor
Marc Bouissou, Professor, Ecole Centrale Paris	Co-supervisor
Jean-François Boujut, Professor, Grenoble INP	Examiner
Claudia Eckert, Professor, Open University	Examiner
Samuel Gomes, Professor, UTBM	Reviewer
Marija Jankovic, Assistant Professor, Ecole Centrale Paris	Co-supervisor
Olivier Maas, Industrial supervisor, Thales Air Systems	Examiner
David Wynn, Senior Research Associate, University of Cambridge	Reviewer

2014ECAP0010

Abstract

The aim of this research work is to propose a method allowing innovation integration in early design stages and supporting architecture design of complex systems that have significant implications for the rest of overall system life-cycle. Focusing on system architectures generation support, this method proposes to use Bayesian networks combined with Constraint Satisfaction Problem (CSP) techniques in order to semi-automatically generate and evaluate complex systems architectures. Bayesian network model is used to represent the design problem in terms of decision variables, constraints and performances. Furthermore, an architecture generation algorithm is proposed to generate feasible solutions and to cluster them with regard to a given confidence level threshold. This confidence level is representing the estimation of the uncertainty on the overall system. Estimation of architecture performances are also calculated within the Bayesian network. Once the system architectures are generated, a CSP model optimises the component placement regarding placement constraints and optimisation objectives defined by designers. Software has been developed for the purpose of problem modelling and solutions visualisation. Two industrial implementations yielded in a generation of a high number of architecture solutions. In order to test the feasibility of architecture selection in an industrial environment, a study was conducted integrating four system designers. This study underlined the difficulties in defining architecture selection criteria and provides recommendations for the future system architecture selection support.

Key words: *Design Methodology, Complex Systems, System Architecture, Bayesian Networks, Constraint Satisfaction Problem*

Résumé

L'objectif de ce travail de recherche est de proposer une méthode d'aide à l'intégration d'innovations dès la conception de l'architecture d'un système complexe. Cette étape de la conception a en effet de forts impacts sur le reste de cycle de vie du produit. En se focalisant sur l'aide à la génération d'architectures de systèmes complexes, cette méthode utilise un réseau Bayésien combiné à un problème de satisfaction de contraintes (CSP) pour générer et évaluer automatiquement des architectures de systèmes complexes. Le modèle de réseau Bayésien proposé est utilisé pour représenter le problème de conception de l'architecture en termes de variables de décision, de contraintes et de performances. Un algorithme parcourt le graphe ainsi défini afin de générer les solutions d'architecture qui sont considérées comme faisables et qui présentent un niveau de confiance acceptable. Ce niveau de confiance estime l'incertitude associée à chaque architecture générée. Les performances des architectures sont aussi calculées grâce au réseau bayésien. Une fois les architectures générées, un modèle de problème de satisfaction de contraintes permet d'optimiser le placement des composants au vu des contraintes de placement et des objectifs d'optimisation préalablement définis par les concepteurs. Un logiciel a été développé pour faciliter la modélisation du problème et la visualisation des solutions. Deux cas industriels ont permis de tester la méthode et de nombreuses solutions d'architecture ont été générées. Afin de tester la faisabilité de l'étape de sélection d'architectures dans un cadre industriel, un atelier de sélection d'architectures a été organisé afin d'être par la suite analysé. Il a impliqué quatre concepteurs de Thales et portait sur un des cas industriels précédents. Cette dernière étude a souligné des difficultés dans la définition des critères de sélection des architectures et propose des recommandations pour un futur support à la sélection d'architecture système.

Mots clés: *méthodologie de conception, systèmes complexes, architecture de systèmes, réseaux bayésiens, problème de satisfaction de contraintes*

Table of contents

List of tables

List of figures

List of abbreviations

BB	Building Block
BN	Bayesian Network
C4ISRAF	Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance Architecture Framework
CSM	Concept selection method
CSP	Constraint Satisfaction Problem
DoDAF	Department of Defence Architecture Framework
M&S	Modelling and Simulation
MCDA	Multi Criteria Decision Aid
MCDM	Multi Criteria Decision Method
MoDAF	Ministry of Defence Architecture Framework
OOPRM	Object-Oriented Probabilistic Relational Models
PSM	Problem Structuring Method

Foreword

This PhD thesis dissertation results from a collaboration between Ecole Centrale Paris and Thales Air Systems under a CIFRE (Conventions Industrielles de Formation par la REcherche) contract between January 2011 and December 2013. Chapters 1 and 2 ensure introduction and present overall scheme of this research work broken in several complementary parts: Chapters 3, 4 and 5 are scientific papers published or submitted for publication in international journals and conferences. Chapter 6 summarises the limits of the overall research contributions and discusses further research needed.

The following papers included in the dissertation have been published:

- Moullec, M. L., Bouissou, M., Jankovic, M., Bocquet, J. C., Requillard, F., Maas, O. and Forgeot, O. (2013). "*Toward system architecture generation and performances assessment under uncertainty using bayesian networks* ", Journal of Mechanical Design, 135(4): 13. (Chapter 3)

- Moullec, M. L., Jankovic, M., Bouissou, M. and Bocquet, J. C. (2013). "*Proposition of combined approach for architecture generation integrating component placement optimisation*". In Proceedings of the IDETC/CIE 2013. Portland, OR/USA (Chapter 4)

The following paper included in the dissertation has been submitted for publication:

- Moullec, M. L., Jankovic, M. and Eckert, C. (2014). "*Investigating the importance of criteria selection in system architecture: Observations from industrial experiment*", submitted to Research in Engineering Design. (Chapter 5)

Some repetitions between Chapters 3, 4 and 5 can be witnessed, which is inherent to the dissertation by papers style. The chapter "Bibliography" includes overall references. However, references used in a "paper chapter" can also be found at the end of the chapter.

Acknowledgements/Remerciements

First of all, I would like to warmly thank the jury members of this PhD thesis : David Wynn and Samuel Gomes for accepting to review this thesis and Jean-François Boujut for his interest in my work. I appreciated their encouragement, insightful comments, and hard questions. Also, I would like to offer my special thanks to Claudia Eckert, not only for chairing my jury, but also for advising me and supporting me all along these last months.

J'aimerais remercier tout autant mes encadrants (et ils sont nombreux !). Tout d'abord, Jean-Claude Bocquet, mon directeur de thèse, pour la confiance et la liberté qu'il m'a accordées ainsi que pour son écoute et ses conseils car ils m'ont été précieux. Viennent ensuite mes deux co-encadrants de thèse, Marija Jankovic et Marc Bouissou. Tous deux ont dû passer beaucoup de temps à relire et corriger mon anglais… Merci à Marc Bouissou pour son expertise sur les réseaux Bayésiens et ses encouragements tout au long de la thèse; et merci à Marija Jankovic pour son soutien dans les difficiles moments du début de thèse, pour sa connaissance du domaine de la conception ainsi que pour son initiation aux bonnes pratiques de la recherche. Je n'en oublie pas pour autant mes encadrants à Thales ! Leur implication constante durant ces trois ans m'a permis d'intégrer des tests en « conditions réelles ». C'est ce qui a été, je pense, un des points les plus appréciés de ce travail. Je remercie donc Olivier Maas pour son enthousiasme croissant ainsi que pour sa détermination lorsqu'il n'était pas d'accord avec moi (le résultat n'en a été que meilleur !), François Réquillard pour sa disponibilité et son ouverture d'esprit (car il en fallait lors des premiers tests !) et Olivier Forgeot pour la communication sans faille qu'il a menée au sein de Thales.

Je tiens ensuite à remercier ceux sans qui cette thèse n'aurait pas été la même de par leur participation à mes différents travaux, soit toutes les personnes qui ont accepté de participer au workshop de sélection des architectures, avec une mention spéciale à Eric Jeanclaude pour l'intérêt qu'il a porté à la thèse et ses retours en tant qu'architecte système. Certaines de ses remarques se sont avérées décisives quant au contenu de ce travail. Je tiens aussi à remercier Vincent Mousseau pour ses explications sur les méthodes d'aide à la décision multicritères et ses conseils à propos de l'oganisation du workshop. Enfin merci à Matthieu Marissal et Lonni Besançon, les deux stagiaires que j'ai encadrés à Thales. Leur travail s'est avéré très complet et efficace. Je leur souhaite une belle carrière à tous les deux!

J'en viens maintenant à remercier ceux qui m'ont permis d'évoluer dans un, et je dirais même deux environnements plus qu'agréables durant ces trois années de thèse. Je pense ici à tous mes collègues de Thales et du LGI. Grâce à eux, j'ai pu entre autres:

1) devenir une (presque) spécialiste du poker grâce à l'ancienne génération des doctorants maintenant tous devenus docteurs !
2) devenir une spécialiste de l'organisation d'événements de grande envergure grâce à mon expérience en tant que GO du séminaire de labo ;
3) me rappeler tout ce que j'ai eu tendance à oublier grâce à Carole, Sylvie, Delphine et Corinne qui ont toujours été là pour m'y faire penser!
4) me maintenir en forme en participant (quasi !) hebdomadairement à des matchs et parfois tournois de volley de très haut niveau ! Merci donc à Julien(s), Eric, Olivier, Mourad et Alexandre pour cela. (Eh oui Anthony ! Je t'avais dit qu'un jour ou l'autre tu regretterais de ne pas y avoir pris part…)
5) entretenir le besoin de me maintenir en forme grâce aux nombreux pique-niques/restaurants/soirées organisés par la nouvelle génération de doctorants (ils sont trop nombreux pour tous les citer !).

Je voudrais de plus accorder une mention spéciale à Hakim, Liza et Sena avec qui j'ai partagé de nombreux « goûters » et ainsi pu participer à des débats culturels « mode », « glamour », « orthographe » (et j'en passe…) passionnants !

Enfin, et je ferai court cette fois-ci : un grand merci à ma famille et mes amis (notamment Aline, Saka, Elise, Julia, Sing-Joe, Will, Rémi, Vivien, Maud, David…) qui ont réussi, et j'espère continueront, à me supporter (dans tous les sens du terme) pendant toutes ces années !

Résumé étendu

Contexte

L'architecture d'un système est constituée d'un ensemble de vues abstraites représentant les éléments dudit système ainsi que leurs relations (Crawley, Weck De et al. 2004). C'est l'un des moyens les plus adéquats pour définir et gérer les systèmes dits « complexes ». En effet, un système intègre un nombre d'éléments qui, par leurs interactions, vont faire émerger des propriétés qui lui sont propres. Cependant, cette intégration passe principalement par la gestion des interfaces et peut s'avérer complexe : en théorie, le comportement complet d'un système peut être prédit si tous les couplages et interdépendances sont identifiés et pris en compte ; mais cela est rarement possible. Parce que l'architecture représente les interfaces du système sous des points de vue principalement fonctionnel, structurel et comportemental, elle est un moyen de gérer la complexité du système ainsi que les propriétés qui en découlent. Pour ces raisons, les décisions inhérentes à la définition de l'architecture d'un système vont impacter l'ensemble du cycle de vie de celui-ci, ainsi que fixer environ 80% des coûts qui y seront associés.

Les méthodes et outils actuels de conception proposés dans le cadre de l'ingénierie système permettent de représenter l'architecture d'un système sous différents points de vue ; et de s'assurer que l'ensemble des exigences système sont bien satisfaites. Cependant, ils n'apportent aucune aide quant à sa définition. De plus, les outils de simulation et de modélisation utilisés en conception détaillée ne sont pas non plus adaptés : ils nécessitent de manipuler des données précises, ce qui n'est pas possible en conception préliminaire car l'information y est généralement incomplète, floue et incertaine. Cela rend les architectures envisagées pour le système très difficiles à évaluer. De nouveaux outils d'aide à la conception préliminaire, et notamment à la définition des architectures système, sont donc requis (Yannou, 2001).

Objectifs

Cette thèse a été financée par Thales Air Systems et réalisée dans le cadre d'un contrat CIFRE (*Conventions Industrielles de Formation par la REcherche*) entre Thales Air Systems et le Laboratoire Génie Industriel (LGI) de l'Ecole Centrale Paris, entre janvier 2011 et décembre 2013.

L'activité principale de Thales Air Systems est de concevoir des radars à des fins civiles et militaires. Les départements du bureau d'études mécaniques et du bureau d'études « antenne » travaillent en étroite collaboration sur la conception des antennes, et notamment sur les études de conception amont. Malgré cela, les ingénieurs de ces départements rencontrent certaines difficultés au cours du développement de leurs antennes, ce qui se traduit par des performances a priori sous-optimales et des temps et coûts de développement croissants. Ils ont donc conjointement initié ce travail de recherche afin d'améliorer leur processus de conception amont. L'objectif principal de cette thèse est de proposer une méthode de conception amont des systèmes complexes, permettant d'intégrer des solutions techniques innovantes et d'explorer l'espace de conception qui en résulte par la génération et l'évaluation d'un nombre élevé de concepts d'architectures pour à terme sélectionner quelques concepts prometteurs.

Vue d'ensemble des travaux de recherche

Cette thèse a été menée selon un protocole de « recherche action » qui consiste à impliquer le chercheur directement au sein du problème observé. Ainsi, cette recherche inclut plusieurs phases d'audit et de diagnostic, de reformulation des problèmes rencontrés, de proposition de nouveaux modèles et méthodes pour finir par leurs tests en environnement industriel.

Audit Industriel & Etat de l'art

Thales Air Systems conduit une politique d'ingénierie système dans le cadre de laquelle ils ont développé leurs propres outils. Néanmoins, ces outils n'apportent pas un soutien spécifique au processus de définition de l'architecture système, ce qui donne lieu aux problèmes suivants:

- le manque de temps et la complexité du système limitent le nombre d'alternatives d'architecture qui sont étudiées ;
- le développement de solutions techniques qui ne prend généralement en considération qu'un seul domaine d'expertise (électronique, thermique, mécanique, …) ne permet pas d'évaluer de façon objective les différentes solutions ;
- les problématiques liées au placement des composants sont difficiles à intégrer et à prendre en compte. Elles sont pourtant nombreuses ;
- le processus de définition de l'architecture utilise peu les modèles préexistants. Leur réutilisation serait cependant utile à l'étude de nouvelles solutions.

Ces difficultés entraînent l'augmentation des coûts et temps de développement, et contribuent au fait que les antennes ne sont pas optimisées. Ces problèmes sont accentués quand il s'agit de considérer des solutions techniques innovantes, sur lesquelles il n'existe encore que peu de données. Afin de pallier ces problèmes, il est nécessaire d'apporter une méthodologie aidant à la conception préliminaire des systèmes en permettant l'intégration d'innovations technologiques, une meilleure exploration de l'espace des solutions d'architecture, ainsi qu'une comparaison et une sélection des architectures selon une approche multicritères et multi-domaines.

Au vu de la taille des espaces de solutions d'architecture, notre revue de littérature s'est rapidement concentrée sur les méthodes (semi-)automatisées de génération d'architectures de système. Selon Cagan (2005), de telles méthodes doivent systématiquement suivre quatre étapes : représentation du problème de conception, génération des solutions, évaluation de ces solutions pour finir par une étape de guidage permettant de raffiner le modèle. Les méthodes actuelles permettent de représenter le problème de conception et d'en générer les solutions. Cependant, la plupart d'entre elles ne permet pas d'évaluer les architectures générées sur la base des exigences système, et ce, essentiellement parce que les données nécessaires à cela sont incertaines et incomplètes. En conséquence, les étapes de sélection et de guidage ne peuvent être effectuées.

Compte-tenu de l'audit industriel et de notre revue de littérature, il semble que les principaux challenges de ce travail de recherche concernent la gestion de la complexité du système ainsi que l'intégration de solutions innovantes, et donc la prise en compte de données incertaines. Ainsi, notre analyse a souligné qu'une méthode d'aide à la conception des architectures systèmes devrait :

1. intégrer les opinions et les connaissances des concepteurs ;
2. permettre la modélisation de données incertaines ;
3. effectuer une recherche exhaustive dans l'espace de solutions ;
4. permettre d'évaluer et de comparer les solutions trouvées selon des critères multiples et multidisciplinaires.

Cet état de l'art nous a permis d'affiner nos questions de recherche dont la formulation et les réponses apportées sont présentées dans le paragraphe qui suit.

Axe de recherche 1 : exploration des solutions d'architectures

La phase de conception des architectures système repose essentiellement sur la créativité et l'intégration des connaissances des concepteurs. Automatiser ce processus est compliqué parce qu'il nécessite de représenter des éléments de natures diverses (fonctions, composants, interfaces) d'une manière structurée et intégrant la part d'incertitude associée à ces données, notamment lorsque des technologies innovantes sont considérées. De plus, ces éléments ont trait à des domaines différents et peuvent se trouver en très grand nombre, ce qui interroge sur le type de représentation à adopter. La première question de recherche est donc la suivante :

Comment intégrer les innovations dès les phases amont de la conception et explorer l'espace des solutions des architectures de systèmes complexes qui en résulte ?

Dans ce but, nous proposons une méthode constituée de deux étapes permettant 1) de générer des solutions d'architectures puis 2) d'optimiser le placement des composants de chacune d'entre elles.

Dans la première étape, le concepteur modélise le problème de conception sous la forme d'un réseau bayésien en déclarant des variables de décision, des contraintes et des performances. Ces éléments sont suffisamment génériques pour apporter la flexibilité nécessaire à l'intégration d'innovations dans la modélisation d'un problème de conception. La modélisation sous forme de réseau Bayésien permet quant à elle d'estimer le niveau d'incertitude associée à l'utilisation de ces innovations. Pour chacun de ces éléments, nous proposons un modèle de réseau bayésien de référence. Une fois le problème modélisé, l'algorithme que nous avons créé parcourt l'espace des solutions afin d'identifier les architectures système respectant les contraintes déclarées, et ainsi considérées comme faisables. A chaque solution sont associées des estimations de performances ainsi qu'un indice de confiance global, représentatif de l'incertitude associée à la solution.

⇒ Cette étape a fait l'objet d'une publication retranscrite dans le chapitre 3.

Dans une seconde étape, nous proposons d'optimiser le placement des composants de chaque solution générée en utilisant un problème de satisfaction de contraintes (CSP). Ce CSP est défini au préalable par les concepteurs à l'aide de contraintes de placement à satisfaire et de performances à optimiser. Les données d'entrée du CSP sont le type, le nombre et les dimensions de tous les composants constituant une solution d'architecture générée par le réseau bayésien. Le CSP détermine alors une solution de placement optimisé, exprimée sous la forme d'une liste de quadruplets indiquant l'orientation et les coordonnées {x, y, z} de chaque composant de la solution d'architecture. Pour en faciliter la lecture, nous proposons de visualiser ladite solution en 3D. Toutes les solutions d'architectures peuvent être optimisées de la même façon en utilisant le même CSP.

Au final, cette méthode nous permet de générer un ensemble d'architectures innovantes faisables, d'en estimer les performances, et d'en proposer une visualisation 3D. Les deux modèles servant à la génération, le réseau bayésien et le CSP, peuvent être modifiés à tout moment par le concepteur pour être affinés ou corrigés. Ainsi, le concepteur peut changer les hypothèses de modélisation, fixer ou ajouter des variables, supprimer ou ajouter des contraintes afin d'explorer de manière itérative l'espace des solutions qui s'offre à lui. La figure 1 montre une vue d'ensemble de la méthode et fait le lien avec les étapes préconisées par Cagan.

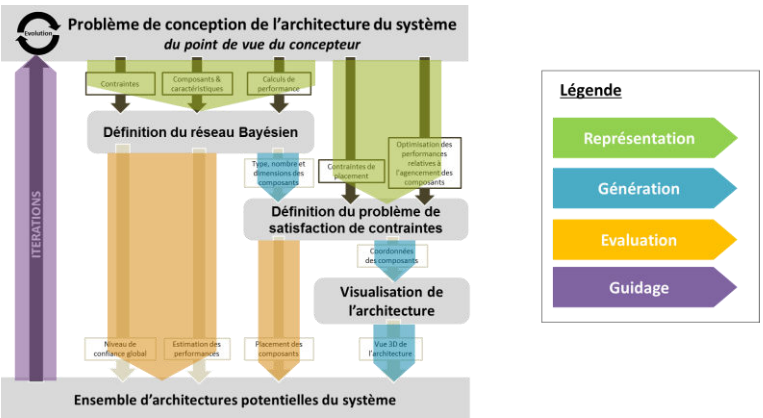


Figure 1. Vue d'ensemble de la méthode

Cette méthode a été appliquée à deux cas d'étude industriels. Le premier concernait l'architecture du système de refroidissement d'une antenne radar donnée. Seule le réseau Bayésien a été réalisé. Elle a permis de trouver 4 solutions, toutes considérées comme faisables par les concepteurs. En outre, la solution à laquelle les concepteurs étaient arrivés manuellement a été identifiée. Le second cas d'étude visait à proposer de nouvelles architectures pour un module électronique comprenant les principales fonctions de l'antenne. Ce problème de conception était bien plus exploratoire et très peu contraint. De ce fait, 800 architectures ont été générées par le réseau Bayésien. Toutes ont été optimisées au moyen d'un CSP qui avait pour objectif d'en optimiser la profondeur. Ce grand nombre d'architectures générées mène à notre seconde question de recherche qui porte sur la comparaison et la sélection d'architectures de système.

⇒ Cette étape a fait l'objet d'une publication retranscrite dans le chapitre 4.

Axe de recherche 2: comparaison et sélection d'architectures système

L'étape suivant la génération d'architectures, soit la sélection d'architectures potentiellement prometteuses, est très importante puisqu'elle va déboucher sur le choix d'une architecture spécifique dont les impacts affecteront l'ensemble des étapes du cycle de vie du système. Cependant, chacune de ces étapes est caractérisée par des objectifs différents, parfois contradictoires, souvent interdépendants, et pour lesquels il est parfois difficile d'évaluer les architectures. La seconde question de recherche est ainsi formulée :

Comment comparer et sélectionner des architectures de systèmes complexes alors que les critères de sélection sont encore mal définis et que l'information caractérisant chaque architecture est floue, incertaine et incomplète ?

Une réponse à cette question de recherche réside dans l'usage des méthodes d'aide à la décision multicritères (MCDM). En théorie, de telles méthodes ont pour but d'aider les « décideurs » à choisir la solution qui présente le meilleur compromis. Ces méthodes sont basées sur un ensemble de critères préalablement défini. Afin d'évaluer l'utilité et la plus-value des MCDM dans le cadre de la sélection d'architectures système, un atelier de sélection d'architectures a été organisé : quatre experts de Thales étaient chargés de définir les critères sur lesquels les architectures devaient être évaluées. Par la suite, le groupe a été scindé en deux : le premier binôme devait sélectionner les 5 architectures les plus prometteuses parmi les 800 qui avaient générées. Le deuxième binôme devait faire de même mais avait pour obligation d'utiliser la méthode Prométhée, une MCDM adaptée à la détermination de portefeuilles de projets et basée sur l'élicitation des préférences. Une dernière étape consistait à comparer et classer les 10 architectures sélectionnées par les deux groupes afin de déterminer quelles solutions, parmi celles sélectionnées avec ou sans l'aide de Prométhée, étaient de meilleure qualité. Au final, les experts ont fait face à plusieurs difficultés, notamment dans la détermination et l'utilisation des critères de sélection. De plus, ils n'ont pas réussi à classer les solutions sélectionnées.

Nous avons transcrit et analysé l'enregistrement vidéo de l'atelier, ce qui nous a permis de soulever certains points :

- les critères de sélection des architectures sont interdépendants ;
- l'information caractérisant les architectures ne permet pas de définir une famille de critères de sélection qui soit exhaustive ;
- les interdépendances entre les critères rendent difficile l'établissement de critères non-redondants.

Ces trois points montrent deux choses : les critères de sélection d'architectures ne vérifient pas les postulats de la plupart des méthodes d'aide à la décision multicritères ; leur identification est loin d'être immédiate et évidente, contrairement à ce que les méthodes de sélection actuelles (en conception de produit) laissent à penser (Okudan and Tauhid 2009). En conséquence, cet atelier a souligné le besoin de méthodes de sélection adaptées à la problématique de l'architecture système.

⇒ Cette étude a fait l'objet d'un papier soumis pour publication et retranscrit dans le chapitre 5.

Cette dernière étude a ouvert de nouvelles questions portant spécifiquement sur la sélection d'architectures. Bien qu'il n'ait pas été possible d'y répondre dans le temps alloué à cette thèse, il conviendrait d'y répondre afin d'assurer une utilisabilité totale de notre méthode.

Apports et perspectives

A l'issue de cette thèse, les principales contributions de nos travaux sont :

- une méthode d'aide à la génération d'architectures donnant la possibilité d'intégrer des solutions innovantes, accordant au concepteur un rôle prépondérant dans la modélisation du problème de conception, générant automatiquement des architectures système et estimant leurs performances tout en intégrant les incertitudes liées aux données utilisées et finalement proposant une visualisation 3D du placement optimisé de leur composants ;
- un logiciel permettant une modélisation facilitée du problème de conception et une visualisation synthétique des solutions et de leurs performances ;
- une amélioration du processus de conception préliminaire des architectures système : l'utilisation de cette méthode permet une meilleure compréhension du problème de conception au travers de la modélisation qui y est proposée. De plus, l'exploration de l'espace de conception se veut exhaustive et multidisciplinaire, ce qui devrait réduire les itérations en phase de conception. L'estimation des performances de chaque solution permet de mieux justifier les choix de conception. Finalement, le tout donne lieu à une capitalisation des connaissances accrue;
- un éclairage sur les difficultés liées à l'évaluation et la sélection multicritères des architectures systèmes ainsi que sur le besoin de méthodes de sélection adaptées;
- l'application et la validation de ces contributions théoriques sur deux cas d'étude industriels dans une grande entreprise internationale. Cette application a permis de souligner le caractère robuste et généralisable de nos travaux.

Ce travail est une première étape vers une méthode d'aide à la conception des architectures système. Les principales perspectives qui en découlent sont liées à l'amélioration et l'enrichissement du modèle de génération d'architectures par l'utilisation de modèles relationnels probabilistes orientés objets (Torti, Wuillemin et al. 2010) et l'intégration de nouvelles performances et de nouvelles contraintes, notamment en ce qui concerne le placement des composants. De plus, il reste encore une étape importante à franchir : celle de la sélection des architectures. En effet, cette étape ne semble pas faisable en utilisant les méthodes de sélection actuelles, et nécessite une approche spécifique permettant de gérer des critères dépendants ainsi que l'intégration des incertitudes, encore fortement présentes à cette étape de la conception.

1

Introduction

Complex system architecture constitutes a major bridge between conceptual design, where a number of design principles are envisaged, and detailed design in which all elements of the system need to be defined, integrated and validated. System architecture, representing all major system elements and their interconnections, is a way that designers go about in defining the system in a coherent way. However, system architecture design is still not supported by methods and tools and complex systems that contain hundreds of variables are hard to define. This chapter aims at giving an overview of the context and underlying problems as an introduction to further research problem development and positioning.

1.1 Context

1.1.1 Role of architecture in complex systems

System architecture can be viewed *"as an abstract description of the entities of a system and the relationships between them"* (Crawley, Weck De, et al. 2004). It is one of the most convenient ways to define and manage complex systems. Indeed, a system integrates a set of elements so that their collective behaviour achieves a priori defined needs. This integration must pass through the proper management of its interfaces. Together these interfaces raise new system properties that no subsets of its elements have (Crawley, Weck De, et al. 2004; Krob 2009). However, there exist numerous types of interfaces: Sanchez (1999) proposed a classification of system interfaces in nine categories ranging from physical properties to consideration of system environment, including data and energy flows between system elements. Consequently, considering and managing all interfaces in a consistent way may become complex: in particular parameter or component couplings relating to different domains must be exhaustively identified, which is rarely possible. As result, the global behaviour of a complex system cannot be entirely predicted, often leading to the emergence of non-anticipated properties. But a way of controlling this complexity is defining the system through its architecture.

In this work, we base our definition of system architecture on the framework proposed by the Gero and Kannengiesser with their Function Behaviour Structure (FBS) model (Gero and Kannengiesser 2004). System architecture is a scheme that represents the elements that map from functional domain to structural domain in order to estimate system performance. Therefore, it consists of functional, structural and behavioural architectures that are mutually interrelated (Jankovic 2013). Other perspectives can also be described in order to bring more details on interactions and behaviours of the system (Crawley, Weck De, et al. 2004; Levis 2011). In this thesis, we mainly deal with the behavioural and structural views of system architecture.

Because system architecture defines all the interfaces, its design is a key step during system development. The behaviour of the system, its complexity, as well as its properties, such as robustness, adaptability, flexibility, safety, are direct results of system architecture choices (Crawley, Weck De, et al. 2004). For these reasons, system architecture impacts the whole system life-cycle (Whitney 2004; Fixson 2005) and commits around 80% of the associated costs (Whelton, Ballard et al. 2002b).

1.1.2 Increasing need for system architecture design tools

Defining complex system architecture can help designers to create systems with desired behaviours and avoid undesirable properties (Crawley, Weck De, et al. 2004). However, defining complex system architectures consists in integrating a high number of interacting subsystems. The parameters that define the latter are heterogeneous and the resolution of complex systems therefore requires multidisciplinary approach. Considering that complexity is the source of severe problems and failures, a general trend consists in attempting to decrease the complexity (Maurer 2007). Even so, an increase of system complexity is noted in engineering development and concerns not only systems and products, but also markets, organisations and processes (Lindemann, Maurer et al. 2008). The reasons for this are multiple and include increase of mechatronic components and software use (Davies and Hobday 2005) as well as increase of product customisation (Piller, Moeslein et al. 2004). In this context, control of complex systems, and thus mastery of their architectures, is becoming a major challenge.

For this purpose, systems engineering approaches are adopted by most of companies. This research field comprises concepts, methods and best practises developed in industry in order to master complexity. The research field is articulated around two main dimensions which are:

- an engineering dimension focused on the industrial system that is to be made;
- an organizational dimension focused on the management of resources used for making the system.

From this, standards of systems engineering (ANSI/EIA 632, ISO/ICE 15288) and architectural frameworks (C4ISRAF, MoDAF, DoDAF) emerged. The latter aim at supporting development and documentation of system architectures and prescribe models that are mainly descriptive (Nordqvist, Edberg et al. 2006). In particular, SysML is a semi-formal language that is especially adapted for their implementation (Krob 2009). Descriptive models are useful for sharing a common view of complex systems as well as improving global understanding and management of them, notably for monitoring initial system requirements; but, as simulation capacities are not fully integrated, none of these models allows defining and exploring architecture design alternatives. They are consequently insufficient to support the creative phase of system architecture design. An alternative way to deal with system architecture design is using modelling and simulation (M&S) tools. However, current M&S tools require precise information and can be used only when the architecture concept is already defined. Indeed, M&S tools cannot be used earlier because defining system architecture involves heterogeneous parts: architecture elements belong to different engineering domains and do not always have the same level of granularity (Scaravetti 2004). In addition, the knowledge about architecture elements is incomplete, uncertain and fuzzy. These difficulties hamper modelling and assessment of system behaviour and properties making the definition of architecture a delicate step in system development. Specific methods and tools are still needed to help designers in this process (Chakrabarti, Bligh et al. 1992; Yannou 2001).

1.1.3 Thales Air Systems: Designing Radar Antennas

This research was conducted in a collaboration between Industrial Engineering Laboratory *(Laboratoire de Génie Industriel*, LGI) at Ecole Centrale Paris and Thales Air Systems. Thales Air Systems is the subsidiary firm of Thales group which designs radar antennas for civil and military applications.

The basic functions of a radar antenna consist in transmitting and receiving electromagnetic signals in order to detect the presence of objects in a given area. To be usable, the transmitted signals must be amplified before being radiated and the received signals must be amplified before being processed. Active antennas amplify the signals in close proximity to the radiating elements; in general within electronic modules organised as building blocks (Figure 1-1). Designing such antennas is extremely complex and costly, in particular because amplifiers, which heat up the most, are located in the antenna and require an efficient cooling system. In addition, military applications require cutting edge technology and therefore constant innovation. The choice of specific technologies occurs very early in the design process and may require significant investments. This is even more important as radar antennas are generally manufactured in small series.

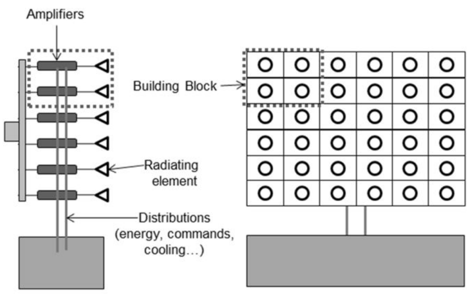


Figure 1-1. Active antenna (adapted from (Roger 1999))

Radars are incremental products which require several years of development and have an intended life span of several decades. The obligation to provide highly reliable and safe products makes innovative solutions to be considered more risky than those established in the company for a long time; however innovation remains the best way to compete in the market. To assess the impacts of introducing innovations, different solution alternatives are studied at the very early design stages, in which different antenna architectures are envisaged and defined. Such studies are time consuming and require a multidisciplinary investigation in order to consider interactions related to different domains, such as thermal effects on electronic devices. As result, only a few architecture alternatives are studied.

This collaboration had been set up at the joint request of two Thales Air Systems departments:

- Antenna design department is in charge of defining antenna solutions in order to ensure operational system requirements;
- Mechanical department is responsible for mechanical integration of the antenna solutions proposed by the antenna design department.

Due to their complementary missions, these departments work in close collaboration throughout antenna development phases organised in a stage-gate process (Figure 1-2), using a system engineering approach. A stage-gate process is decomposed in discrete stages, or predefined activities, preceded by Go/No Go decision points or gates. It aims at helping manager in the mastering of the product development process from idea to launch (Cooper 1994). However, several difficulties are detected during development process:

- Lack of means: the investigation of one architecture solution requires developing specific evaluation models. Due to time and cost constraints, only a few partial architecture solutions are studied.
- Sub-optimal performances: the studied solutions are not guaranteed to offer the best trade-offs in terms of performances, especially since not all performances are assessable for a given solution.
- Design iterations: as the assessment of some performances is missing, design problems are often identified only when the antenna development is advanced, i.e. when most of system design choices have already been freezed.
- Increase of time and development costs: as result, some technical solutions need to be redesigned, causing changes on other system parts and thus leading to the increase of time and development costs.

The main objective of this collaboration is to support preliminary design of radar antennas, in particular during architecture definition and innovation integration.

1.2 Research objectives and methodology

1.2.1 Research Objectives

This work aims at supporting early design stages of complex systems in an industrial context. Based on the needs identified within Thales Air Systems, the objective is to propose a methodology that helps designers to better steer the choices of antenna architecture throughout divergent and convergent phases of design exploration (Cross 2000): a support should be provided to 1) explore as much as possible the design space while giving the possibility to integrate innovant technical solutions; and 2) identify the most promising architectures to be further deeply studied.

Considering the process of architecture design followed by Thales Air Systems (Figure 1-2), this method should be used during early architectural design phase in order to propose potential architecture solutions to be chosen before System Requirements Review (SRR).

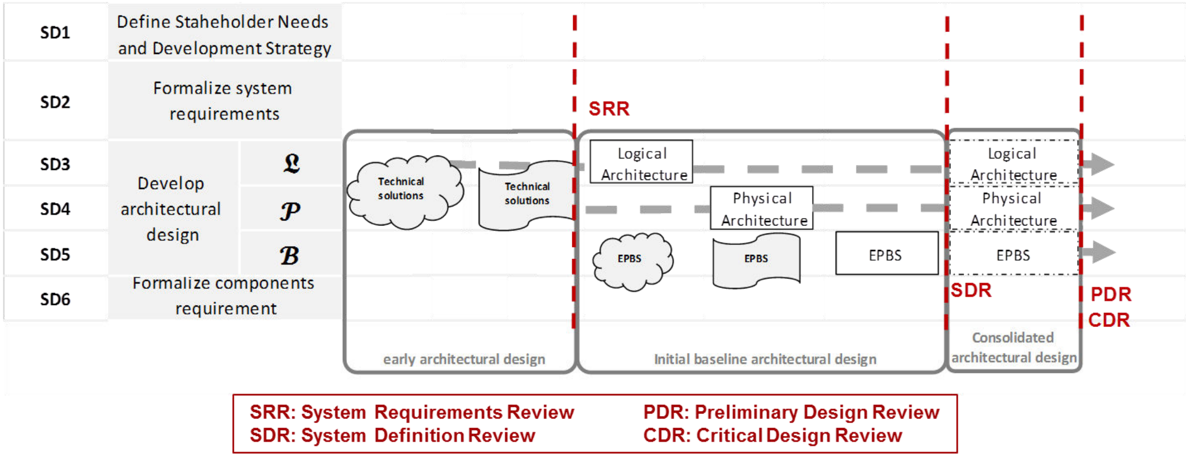


Figure 1-2. System development process of Thales (focused on system architecture design)

1.2.2 Research Methodology

The opportunity of industrial collaboration prompted us to deploy an action research protocol. Action research comes from social and management sciences (Yannou and Petiot 2011). It consists in involving actors that face a problem directly within research process. Meyer (2000) emphasised the fact that action research draws its strength from generating solutions to practical problems; as well as from getting practitioners engaged with research and subsequent implementation activities.

The methodology followed in this study encompasses and sometimes iterates distinct phases of industrial audit and diagnostic, formulation of encountered scientific issues, proposition of new models and methods to end up with industrial implementations (Figure 1-3).

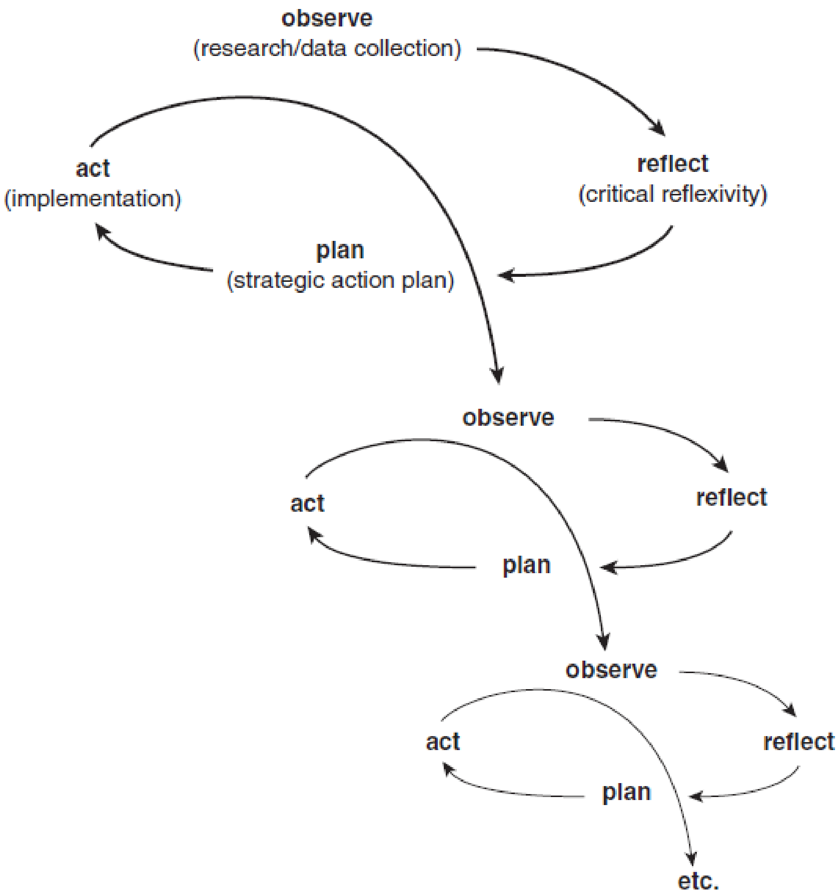


Figure 1-31-3. Process of action research (O'leary, Rao et al. 2004)

This process involved engineers from antenna design and mechanical departments, but also innovation engineers as well as system architects. Through the implementations of industrial cases, each of them was free to give feedback on the proposed methods as well as recommendations. A total of three research cycles (in the definition of O'Leary, Rao et al. (2004)) have been encompassed.

1.3 Dissertation structure

The thesis is organised by papers, each of them corresponding to one research cycle as presented in the previous section. The structure of the chapters is shown in Figure 1-4.

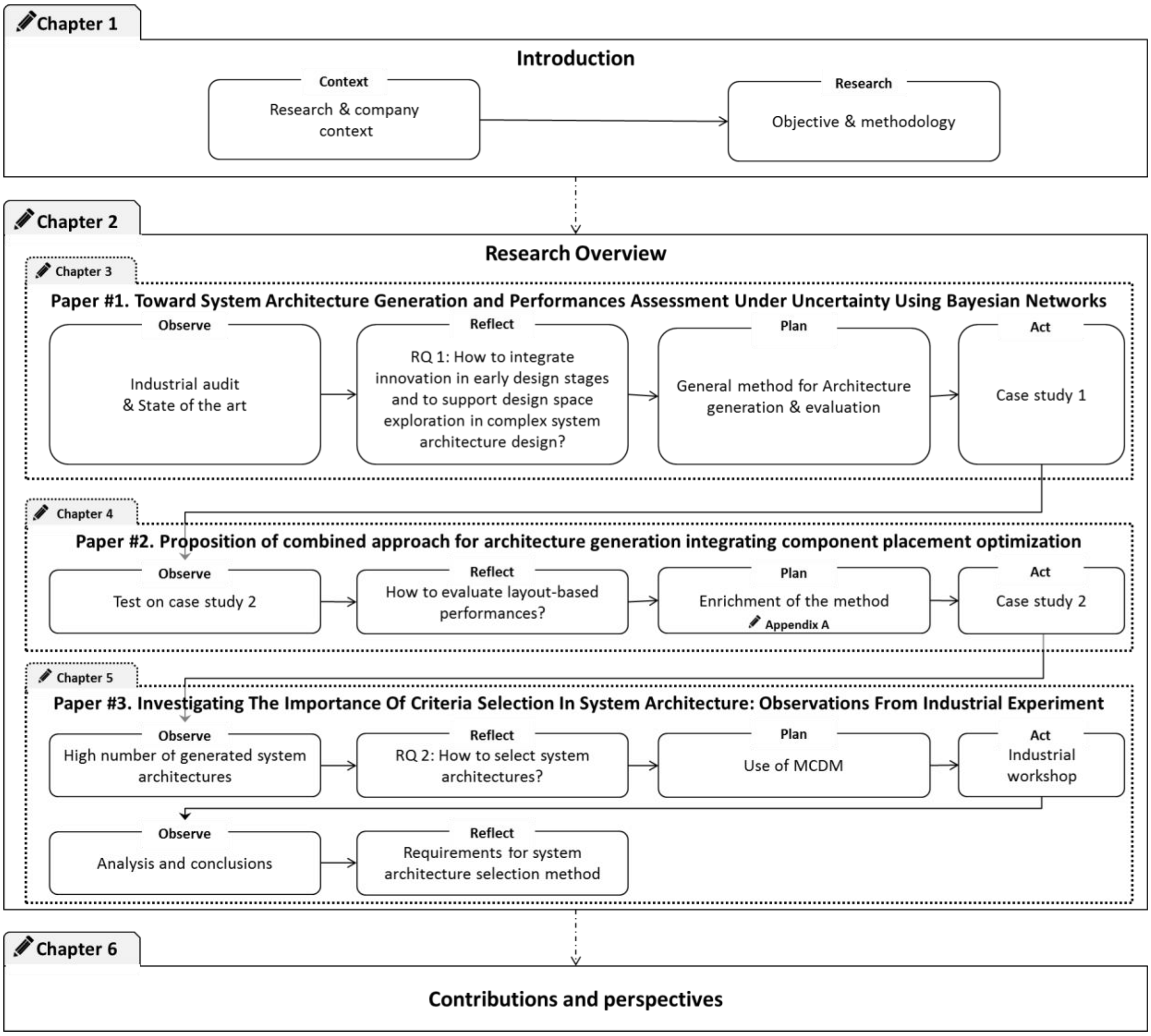


Figure 1-41-4. Research overview

Chapter 1 introduces the general problem and the context in which this research has been conducted. Based on this, the objectives and the scientific approach used in the present thesis are detailed and discussed.

Chapter 2 gives on overview of the dissertation structure, defines research questions and their connection; as well as related contributions. In particular, it provides details about the industrial audit as well as research questions which are not explicitly explained in other chapters. In addition, it gives a synopsis of chapters 3, 4 and 5, as well as industrial achievements.

Chapter 3 describes the first research cycle. It presents the state of the art regarding needs identified within industrial audit, and that allowed to refine research questions. The general approach for integrating innovation and exploring design space is presented. It relies on automatic generation and evaluation of system architectures. This approach is illustrated on a "toy" case or simplified case; validation is made using a real industrial case.

Chapter 4 presents step two of the method in addition to step one presented in Chapter 3. Indeed, the latter did not allow evaluating layout-based performances. However, this evaluation turned out to be necessary when testing the method on a second industrial case. An additional step, considering layout-based performances, is therefore proposed. This second research cycle ends with the validation of the overall method on the industrial case that was initially problematic.

Chapter 5 explores the second research question relating to the system architecture selection. The definition of selection criteria is investigated through a selection workshop involving four experts of Thales. This part emphasised the difficulties inherent to system architecture selection and provides recommendations for a future architecture selection method.

Chapter 6 sums up the contributions and limitations of the thesis and describes possible starting points for future research.

2

Research overview

Chapter 2 aims at presenting the steps that structured the dissertation. First of all, an industrial audit has been conducted in Thales Air Systems in order to identify the needs and difficulties met when designing system architectures. Then, a state of the art has been conducted regarding conclusions resulting from the audit. Both are developed in Section 2.1. The confrontation of the industrial needs with the solutions found in literature review has helped us to precise the research questions and related challenges (Section 2.2). In response to them, the overall research is presented in Section 2.3: the first part details the global approach developed for system architecture generation and evaluation while the second one focuses on the investigation of criteria for system architecture selection. Finally, the method has been validated through industrial implementations described in Section 2.4. Contributions for the company are also listed.

2.1 Industrial Audit & State of the Art

2.1.1 Industrial audit

Thales Air Systems is pursuing an active policy of systems engineering, in which they develop their own tools for complex systems design and management. In particular, recent software, aiming at managing system architecture, has been implemented. However, this software, based on SysML, is mainly used for complex system management, i.e. once the system is already designed. The *architecting* process (Maier and Rechtin 2000) is not supported and results in a number of issues that yield not optimised systems with an increase of development time and costs. In order to identify these issues, the step "observe" of the methodology began with the investigation of the current practice of Thales engineers: interviews with system architects and engineers from radio frequencies to mechanics, including radar domain have been conducted. They were asked to describe their working process, their main difficulties as well as their wishes for a future design methodology. Data analysis led to identification of the main design issues and inconsistencies which are illustrated in Figure 2-1.

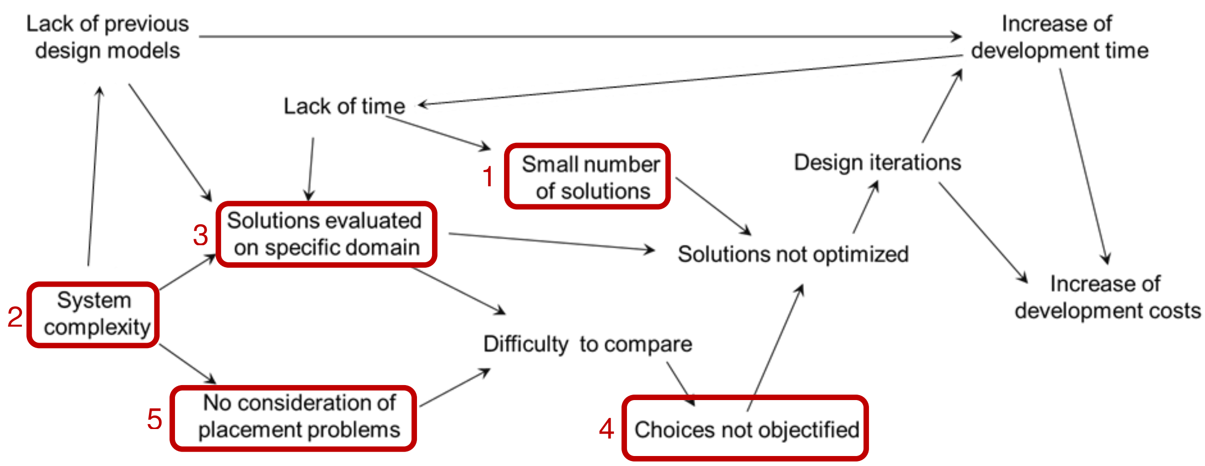


Figure 2-12-1 Cascading issues encountered in antenna preliminary design within Thales

All these issues are interrelated and can be summarised by four main points:

- Lack of time limits design space exploration to a small number of concepts;
- The development of technical solutions considers only a single domain, making their impacts on other disciplines difficult to be assessed. This also affects the comparison of potential architectures and results in numerous design iterations from the different Thales departments;
- Placement constraints and layout-based performances are difficult to consider when evaluating an architecture solution;
- Models of pre-existing systems are missing, which does not permit to use them for investigating new system solutions and therefore to gain time.

In order to address these problems, a method supporting preliminary design and allowing to better steer antenna design choices seems necessary. One objective of the method is also allowing innovation integration, which means design alternatives that potentially include innovative technical solutions. In addition, this method targeted the issues 1, 2, 3 and 4 in priority. Indeed, proposing a multidisciplinary evaluation of a high number of architecture solutions should allow identifying solutions with better performances, as well as comparing them on a regular basis. Design iterations, and by extension time and cost development, should therefore be reduced. Ultimately, the generated architectures as well as models for their evaluation and selection could feed a knowledge database. Initially, issue 5 was not identified as a prior issue, but it became necessary to process it as the research was progressing.

A state of the art of existing approaches, as well as analysis of their strengths and weaknesses regarding these issues has been done subsequently.

2.1.2 State of the art

In view of the industrial audit that requires the exploration of wide design spaces, our literature review quickly focused on automated design methods for preliminary design. An overview is presented in this section in view to define main research directions and highlight related challenges. Specific literature reviews are presented more in depth in Chapter 3, 4 and 5.

According to Cagan (Cagan, Campbell, et al. 2005), design synthesis methods should encompass four steps which are design problem *representation,* solutions *generation*, solutions *evaluation* and a final step of *guidance* providing indicators for model refinement. The main challenges are that uncertainties related to available information have to be integrated while making the best use of it in order to estimate architecture performance. Most of design methods address design of simple products, and are based on design repositories which contain catalogues of functions and components. Starting from a functional architecture or from an existing architecture, artificial intelligence techniques, like graph grammars, constraint satisfaction techniques and genetic algorithms are used to generate potential architectures. Due to lack of information related to components and their interfaces, most of methods do not provide architecture evaluation. If they do, architecture is evaluated according to complexity metrics, or using expert judgements combined within weighted objective function, but never regarding system requirements. Given that *selection* and *guidance* steps require the *evaluation* step, methods generally end with concept generation. The sole methods that allowed system architecture evaluation relates to parameter optimisation when system architecture is already defined. This literature review is deepened in Chapter 3 and Chapter 4.

2.1.3 Gaps between industrial practice and research

The comparison between industrial needs and state of the art highlights several gaps that need to be filled to provide a method useful and usable.

First of all, we address complex system design integrating sometimes hundreds of decision variables. However, depending on the problem addressed, not all decision variables are interesting and the method should enable to concentrate on decision variables specified by the designer: the choice of specific decision variables reduces design space to relevant design regions regarding designer point of view. This also allows an exhaustive space search which would not be possible if all system variables were considered. However, architecting decisions relate to elements that are different in nature and often interdependent: generic and multidisciplinary models are therefore required. Finally, in view to integrating innovation, uncertainties inherent to input data need to be taken into account and measured like any other system architecture performance.

In light of the audit and this discussion of the literature, we argue that system architecture design support should satisfy at least the following four requirements:

1. integration of designer opinions;
2. uncertainty management;
3. exhaustive solution space search;
4. adequate multidisciplinary evaluation and comparison of system architectures;

These requirements raise the two research questions presented in the next section.

2.2 Research Questions

The previous sections highlighted methodological lacks in both phases (divergence and convergence) of system architecture design process and suggests improvements. From them, two research questions have been defined.

2.2.1 RQ1: How to integrate innovation in early design stages and to support design space exploration in complex system architecture design?

Design space exploration of complex system architecture is essential but difficult to support computationally. This design phase mainly relies on creativity and designers' knowledge integration which is often intuitive or tacit (Hubka and Eder 2001). Automatising or at least semi-automatising of this process means to be able representing designers knowledge in a structured way as well as integrating inherent uncertainty and imprecision. This need is increased when innovant solutions have to be considered. In addition, it involves elements that highly differ in nature (functions/components/interfaces) and that interact in different domains (Scaravetti 2004), which questions the visualisation type to be adopted and the way of inputting all the needed information. This is particularly important for the evaluation stage of the method. Moreover, the number of inputs, even for complex systems, must remain reasonable if based on designers' knowledge. The first question research is therefore specified in the following terms:

How to integrate innovation in early design stages and to support design space exploration in complex system architecture design?

2.2.2 RQ2: How to compare and select complex system architectures?

System architecture impacts the whole system life-cycle, making the choice of system architecture an important but delicate step. Every stage of system life-cycle is characterised by different goals not always pulling in the same direction. This issue is compounded by the fact that it is difficult to evaluate system architectures regarding all objectives. Indeed, system architecture is not enough developed to permit a reliable assessment of manufacturing costs, for example. In addition, the trade-offs between conflicting objectives are generally ill-defined due to the fact that they may vary according to the stakeholders and according to the performances range of the considered architectures. This led us to define the second research question:

How to compare and select complex system architectures when objectives are still ill-defined and information on system architecture alternatives is incomplete, uncertain and fuzzy?

2.3 Proposition of a method for complex architecture generation integrating innovation and uncertainty in early design stages

This method aims at supporting designers in the early stages of complex system architecture design. The main challenges for this method are that it must integrate innovation and thus uncertainties on interfaces and component characteristics, and handle component placement to measure all kind of performances. In a first step, feasible architectures are generated using a Bayesian network (BN). Constraint Satisfaction Problem technique is used in a second step in order to check the consistency of BN solutions regarding their ability to reach layout-based performance requirements. Section 2.3.2 then examines the second research question, which is related to system architecture selection.

2.3.1 A multistage method to support complex system architecture generation

Before proposing any method, we have looked at the potential techniques for architecture generation with regard to their ability to integrate uncertainty and complexity issues. A first step consisted in identifying the main sources of uncertainties in design process (De Weck, Eckert et al. 2007) and classifying them. We focused on uncertainties involved in system architecture design process (Aughenbaugh and Paredis 2006), in particular uncertainties inherent to the modelling of designers' knowledge (Ayyub 2010) and the use of simplified calculation models (Aughenbaugh and Paredis 2006). Then, soft computing techniques have been investigated and evaluated with regards to their ability to integrate pre-cited uncertainties, but also their ability to handle complexity. The evaluated techniques are neural networks (Haykin 1994), genetic algorithms (Goldberg and Holland 1988), fuzzy logic (Zadeh 1988), Bayesian networks (Jensen and Nielsen 2007) and constraint satisfaction techniques (Tsang 1993). Finally, we examined them according to the design stages that they could potentially support. Figure 2-2 summarises our views on the different techniques.

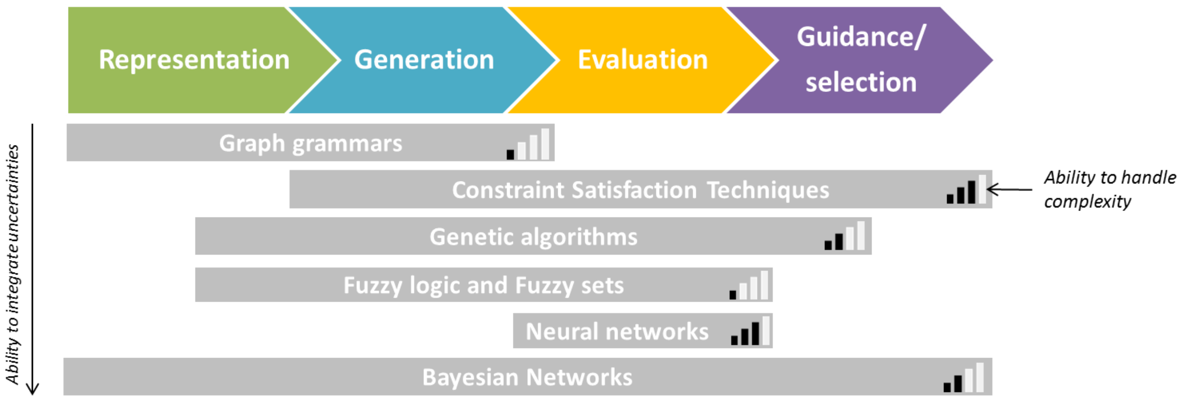


Figure 2-22-2. Evaluation of computational techniques

Besides the incapacity to integrate all kinds of uncertainty, graph grammars and fuzzy logic require qualifying each design element and each relation between two design elements, which did not appear suited to complex system design. Neural networks are good for learning complex mechanisms but the flexibility in utilisation may be difficult: its work like a black-box often hardly understandable by designers, which was not desirable. Genetic algorithms, by their chromosome structure, do not provide a quickly understandable representation of design problem. Constraint satisfaction techniques neither. Finally, Bayesian networks seem to provide appropriate representation of design elements and their interrelations thanks to its graph structure. In addition, they are able to integrate qualitative and quantitative variables, all uncertainties that we wanted to consider, as well as to perform calculations. However, we did not have a real view on their capacity to handle a high number of interrelated data.

These considerations led us to choose Bayesian networks and use them as described in Section 2.3.1.1. Then, we combined them with CSP in order to be able to consider a wider range of performances calculations (Section 2.3.1.2).

2.3.1.1 Step 1: System architecture generation using Bayesian Networks

We opted to exploit as much as possible designers' knowledge. Therefore, designers have to model their architecture design problem under the form of a Bayesian Network. This choice is supported by the fact that BNs are particularly adapted for integrating data uncertainty into a modelling structure, provide good data visualisation and are able to combine quantitative and qualitative data.

In a Bayesian Network, designers define the system architecture design problem in terms of:

- decision variables: design choices that must be made among a set of design alternatives, all described by a set of common characteristics ;
- constraints: constraints that ensure compatibilities between technologies or system requirements satisfaction;
- performances: calculations needed to estimate system architecture performances.

For this purpose, we have proposed Bayesian Network templates representing every element of the design problem. Once the Bayesian Network is complete, an algorithm makes a (potentially large) number of probabilistic inferences on the network in order to identify all feasible architectures. Architecture solutions are described by a set of decision variables. System architecture performances are evaluated directly within the Bayesian network and take the form of probability distributions. In addition, each architecture is provided with a confidence level that represents its feasibility regarding the related uncertainties. Finally, in the light of the number of architecture solutions, architecture performances and confidence level, designers are free to modify or refine the BN model. The general principle of this first step is shown in Figure 2-3.

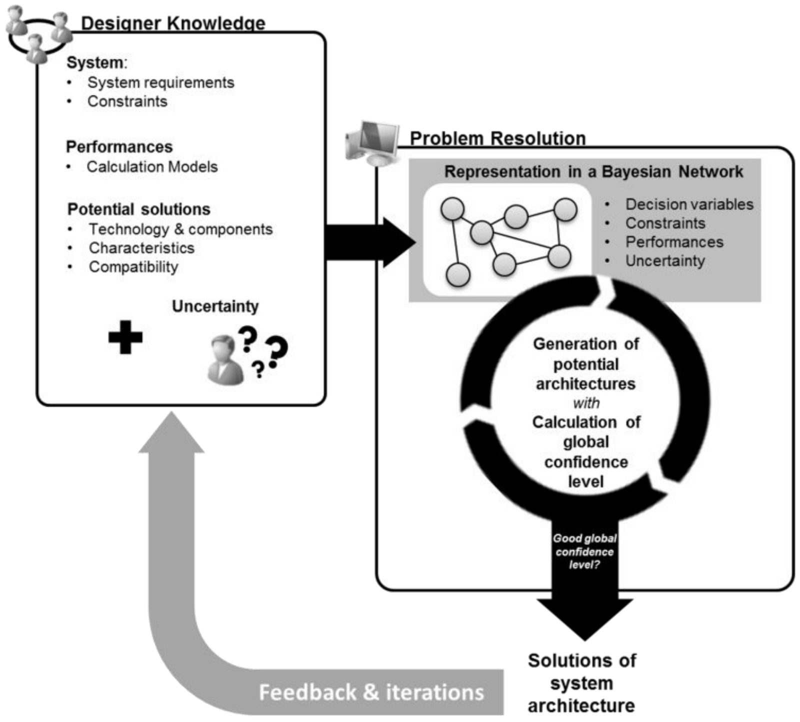


Figure 2-32-3. Bayesian Network method overview

This first step of the method is explained more in depth and illustrated in the scientific paper reproduced in Chapter 3, which has been published in 2013 in Journal of Mechanical Design.

After having successfully applied this method to the cooling system architecture design (presented in Section 2.4 and Chapter 3), we have addressed a more exploratory case study: the architecture design of a building block intended to be used in a new generation of antenna architecture. However, this building block is mainly constrained by layout-based performance requirements. We experimented that, due to the explosion of the number of placement variables, Bayesian networks were not adapted to layout problems. Despite this, handling such performances was necessary. We enriched the present method with a second step, based on constraint satisfaction problem and allowing optimising components placement regarding related requirements.

2.3.1.2 Step 2: Component placement optimisation using CSP

Component placement cannot be handled efficiently using Bayesian Networks. However, layout problems have been largely addressed and several methods have been developed (Medjoub 1996; Cagan, Shimada et al. 2002; Bénabès 2011). Among them, Constraint Satisfaction Problem (CSP) was deemed as the most adapted regarding our requirements as well as the first step of the method. Indeed, inputs of design problem are given under the form of decision variables and constraints; and performance calculations are allowed. This similarity with problem representation used for BN model is an advantage given that it enables the user to stay in the same way of thinking.

A preliminary step to placement optimisation consists in ensuring that all decision variables representing components in BN can be characterised in terms of dimensions. These dimensions may have been already declared in the BN if they are required for performance calculations. Then, CSP model is based on the list of all component alternatives present in the BN, and their corresponding characteristics. Decision variables in CSP are coordinates of all component alternatives as well as their orientation. Designers define position constraints that need to be satisfied. Layout-based performances are declared either as constraints, or as optimisation functions. One CSP is common to all solutions. Once it is defined, a solution generated by the Bayesian network is given as an input to the CSP problem. Resulting solutions are a tuple of coordinates and orientation for every component constituting the system architecture. In order to ease comprehension, the solution is then shown with a 3D visualisation.

2.3.1.3 Method overview

Regarding Cagan's framework (Cagan, Campbell, et al. 2005), the proposed approach is encompassing all necessary steps. Design problem is represented within a Bayesian Network and a CSP. The Bayesian network generates feasible system architectures whereas the CSP proposes an optimised placement of components for each of them. Both models evaluate system architecture performances. For further convenience, a 3D-visualisation is proposed. In view of generated solutions as well as their performances, designers have the possibility to modify or refine the problem. Therefore, designers can iteratively change modelling hypothesis, add or set decision variables, add or delete constraints in order to explore architecture design space. The overall method with its mapping onto Cagan's steps is shown in Figure 2-4.

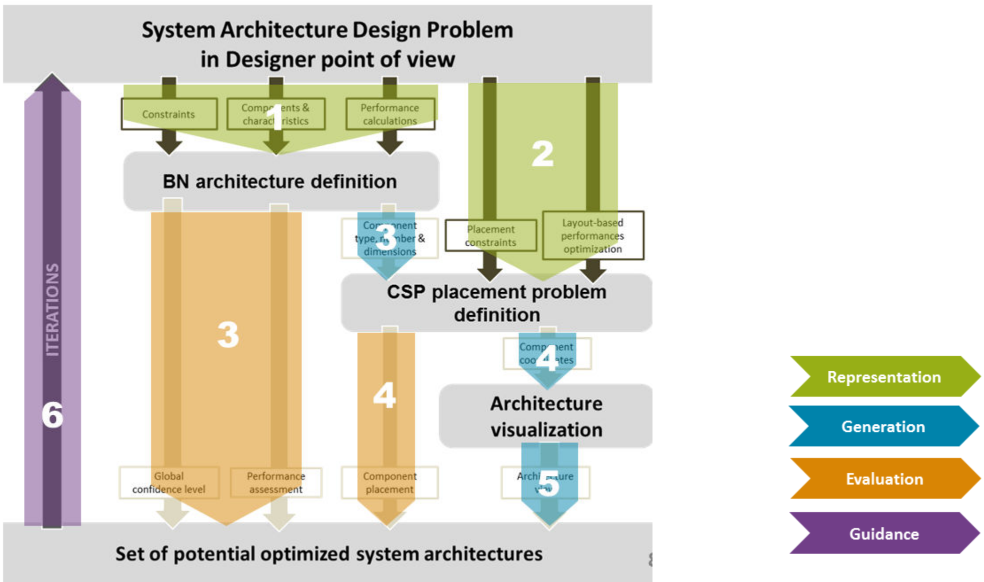


Figure 2-42-4. Overall method

The global method is deepened in the conference paper reproduced in Chapter 4, which has been presented during IDETC2013 conference.

Using this method, 800 building blocks architectures and as many 3D-visualisations have been generated. Obviously, not all architectures should be further studied. Therefore, it was necessary to identify those which were the most promising ones regarding system requirements and company objectives. This raises the question of the necessary use of some multicriteria decision method

(MCDM). In order to estimate the value of MCDM regarding system architecture selection, we organised a workshop during which four experts were asked to identify 5 promising system architectures to be further studied.

2.3.2 Towards system architecture selection: investigating criteria definition for complex system architecture selection

The second research question aims at investigating the definition of criteria used for complex system architecture choice. To this end, multicriteria decision making has been investigated, in particular:

- multicriteria decision methods that provide a number of approaches allowing to identify the best alternatives in view to a set of defined criteria;
- problem structuring methods that aim at helping decision makers to clarify the multicriteria decision problem;

In addition, the concept selection methods proposed in the field of product development, as well as selection in design synthesis methods, have been reviewed in order to determine which criteria are usually used in concept selection. The outputs of this literature review will be discussed regarding industrial needs identified through a workshop that took place in Thales Air Systems and that was dedicated to system architecture selection. This is discussed more in detail in Chapter 5.

2.3.2.1 Workshop in industrial context

In view of the high number of system architectures that the method has generated with the second case study, we decided to organise a selection workshop with Thales engineers. The workshop aimed at determining the usefulness of MCDM when a few system architectures must be chosen amongst many generated ones; as well as identifying an empirical concept selection process. Four engineers took part in the workshop that consisted in defining selection criteria, using the latter for selecting the promising architectures, and comparing the selected ones. After the step "criteria definition" had been done, engineers were split into two teams during the step "architecture selection": one team was free to select architectures in the way they wanted; the members of the other team were provided with Promethee method to help them to select the best architectures (Promethee method is a MCDM based on preferences elicitation). The final step consisted in ranking the solutions selected with and without method in order to determine whether solutions selected with the method were of better quality. During this workshop, the experts faced a number of difficulties in performing these tasks. Some criteria were hard to establish. Other ones were not used. Ten different solutions were identified (five using MCDM, five in an empirical way) but no ranking of them was possible.

2.3.2.2 Analysis & Results

All recordings of the workshop were transcribed to be further analysed. This highlighted some interesting points that question the applicability of MCDM:

- The criteria were interdependent,
- The information available does not allow defining an exhaustive family of criteria,
- The interdependencies between criteria make difficult the elicitation of non-redundant criteria.

These three points do not verify the postulates of MCDM. In addition, the data showed that, contrarily to what most design methodologies suggest, criteria for architecture selection are not straightforward to define. In summary, this workshop highlighted the need to develop selection methods suitable for selection of complex system architectures.

This last research work is deeper developed in Chapter 5.

This workshop opened up new research questions more focused on the selection step of system architecture. It has not been possible to answer them within the allocated time of this PhD. However, it remains necessary to answer them to achieve a total usability of the method proposed in this document. This point is addressed in Chapter 6.

2.4 Operational results in industry

2.4.1 Two case studies

2.4.1.1 Cooling system of a radar antenna

The Bayesian network step of the method has been applied to the design of cooling system architectures. This cooling system is to be integrated in a given radar active antenna. This case study was chosen in view to method validation given that this problem was known by engineers and was already solved. It contained five decision variables centring on the specification of heat-exchangers and fans. Thermal performances were evaluated and three main constraints were entered to ensure architecture feasibility. When this case study was initially addressed without any formal method, experts elicited and studied 12 global architecture solutions to end up with a single feasible solution. Using the proposed method, 4 architectures were generated, including those initially found by the experts, and were all recognised as feasible by the experts. This case study is described and explained in Chapter 3.

2.4.1.2 Design of a subsystem of the antenna

The entire method has been applied to the design of a building-block of the antenna. This subsystem contains the main functions of the antenna and is intended to be used in different antennas. Seven top level functions were identified. Every function could be accomplished by one or two technologies which have been dimensioned. As this case was explorative, only a few constraints about technologies compatibility were added. However, three performances were estimated: pressure loss in the cooling circuit, temperature and dissipated power. This led to the generation of 800 potential architectures. The concept currently used by Thales was found, and its estimated performances were verified. Then, every architecture has been optimised using the CSP model that engineers previously defined. This case study is described in Chapter 4.

2.4.2 *Software development*

2.4.2.1 Architecture design problem modelling

A user interface has been developed in order to facilitate the modelling of the Bayesian network. In this way, designers do not need to directly manipulate the Bayesian network templates that have been proposed. The user interface guides designers in the definition of decision variables, constraints and performances with no need for definition of conditional probability tables: the corresponding Bayesian network is automatically built. In addition, it provides a visualisation of the decision problem, and gives the possibility to store data related to components and performances in dedicated libraries. A more detailed description of this interface is available in Appendix B. However, the interface allowing the creation of the CSP model is still to be developed.

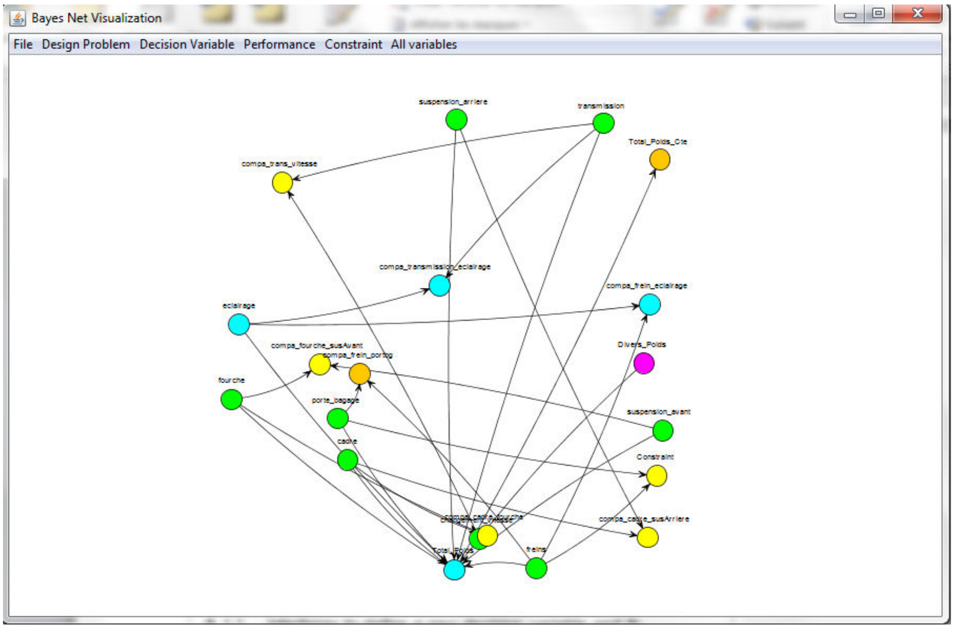


Figure 2-5. User interface of the software

2.4.2.2 Architecture solutions visualisation

A user interface (Figure 2-6) allowing the visualisation of generated solutions has been developed. The elements of the architecture are shown in a schematic view, following conventional architecture visualisations used by Thales engineers. A second view presents the performances of the architecture and allows designers to play with characteristics of components in order to visualise the impact on architecture performances. Finally, a last view shows a 3D-visualisation of the solution following the configuration proposed by the optimisation system. As of now, the interface is dedicated to building block architecture solutions. However, it is developed according to an object oriented approach and can be generalised once the CSP interface is operational. This interface is presented more deeply in Chapter 5 and Appendix B.

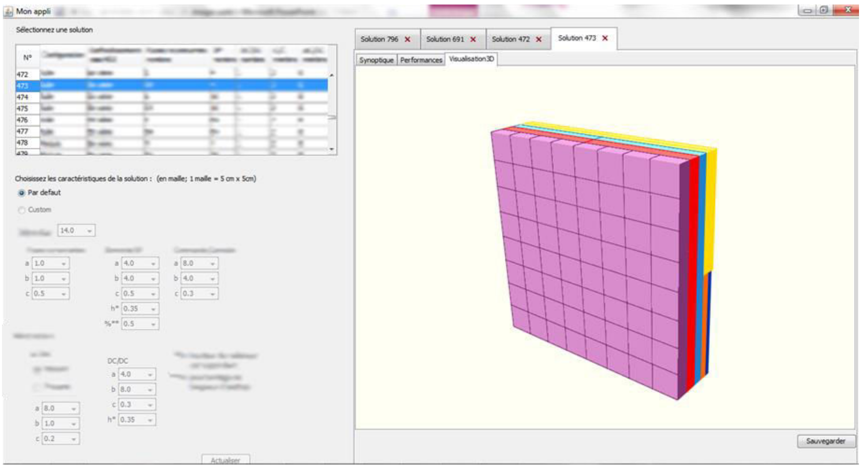


Figure 2-6. Interface for visualisation of building block architectures

2.4.3 Contributions for the company

The two previous sections presented the main contributions for the company: the case studies presented above are real industrial cases and can be enriched or modified to suit the future project needs of Thales. Moreover, the developed software can be used to model easily new architecture design problems. Besides, several less tangible contributions can also be noted:

- Better understanding of the design problems: modelling the architecture problem and visualising this problem allows sharing a common problem understanding from designers of all domains. This may result in the identification of new solutions;
- Better exploration of design space: one main initial issue was that only a few concepts were studied. Using this method, the whole design space is explored and this leads to the apparition of new concepts, not considered by experts before;
- Enhancement of concurrent design: having the constraints and performances of all domains within a same model may result in the reduction of iterations during system architectures development;
- Objectifying solution choices: another problem was the evaluation and comparison of system architectures. Using this method provides an evaluation of system architectures and therefore the possibility to compare them. The choice of one specific system architecture became more objective and can be argued in a more objective way;
- Knowledge capitalisation: reuse of some pieces of Bayesian networks and libraries provided by the developed software offers the possibility to store knowledge on both components and calculation models.

2.5 Synthesis

In this chapter, an overview of the PhD research questions and related studies has been presented. There are 3 distinct parts in this thesis: architecture generation model for innovation integration presented in Section 2.3.1.1; CSP component optimisation and feedback in system architecture process in Section

2.3.1.2; and finally, observations on the overall architecture selection in Section 2.3.2. These studies are given in the three next chapters.

3

Paper #1. Toward System Architecture Generation and Performances Assessment Under Uncertainty Using Bayesian Networks

Marie-Lise Moullec, Marc Bouissou, Marija Jankovic, Jean-Claude Bocquet, François Réquillard, Olivier Maas, Olivier Forgeot

This paper has been published in Journal of Mechanical Design in April 2013, under the following reference:

Moullec M.-L., Bouissou, M., Jankovic, M., Bocquet, J.-C., Réquillard, F., Maas, O. , Forgeot, O., 2013, "Towards System Architecture Generation and Performances Assessment Under Uncertainty Using Bayesian Networks", Journal of Mechanical Design, 135(4):041002-041002-13.

Abstract. *Architecture generation and evaluation are critical points in complex system design. Uncertainties concerning component characteristics and their impact onto overall system performance are often not taken into account in early design stages. In this paper, we propose a Bayesian Network approach for system architecture generation and evaluation. A method relying on Bayesian network templates is proposed in order to represent an architecture design problem integrating uncertainties concerning component characteristics and component compatibility. These templates aim at modelling designers' knowledge concerning system architecture. We also propose an algorithm for architecture generation and evaluation related to the Bayesian network model with the objective of generating all possible architectures and filtering them in view to a defined confidence threshold. Within this algorithm, expert estimations on component compatibilities are used to estimate overall architecture uncertainty as a confidence level. The proposed approach is tested and illustrated on a case study of bicycle design. This first case shows how uncertainties concerning component compatibilities and components characteristics impact bicycle architecture generation. The method is, additionally, tested and implemented in the case of a radar antenna cooling system design in industry. Results highlight the relevance of the proposed approach in view to the generated solutions as well as other benefits such as reduced time for architecture generation, and a better overall understanding of the design problem. However, some limitations have been observed and call for enhancements like integration of designer's preferences and identification of possible trade-offs within the architecture. This method enables generation and evaluation of complex system architecture taking into account initial system requirements and designer's knowledge. Its usability and added-value have been verified on a large-scale system implemented in industry.*

Keywords. *Systems design, creativity and concept generation, uncertainty modelling, design methodology, design automation*

3.1 Introduction

System architecture can be viewed as "an abstract description of the entities of a system and the relationships between those entities" (Crawley, Weck De, et al. 2004). System architecture generation is an essential process during the conceptual design of complex systems. It is largely accepted that this phase consists of generating possible architectures, exploring their strengths and weaknesses, and selecting one based on defined requirements and specifications (Ulrich and Eppinger 1995). Time constraints are a major issue in this phase due to the combinatorial complexity of product architecture (Chakrabarti and Bligh 1996). Moreover, this phase is characterised by a lack of precision and certainty of product data and expert knowledge (Hsu and Liu 2000): preliminary design implies fuzziness in design parameters and contexts, often due to the need to innovate. In this innovation context, interesting technological solutions are often considered risky and not being considered as innovation opportunities. In order to ensure product feasibility, designers often settle for incremental designs and end up modifying existing products (Dieter 2000).

Early design stages are known to be essential for design divergence, i.e. exploring a maximum number of feasible solutions with regard to the required performance. Choices made during the conceptual design stage are critical: numerous studies have shown the impact of decisions in early stages on overall project costs (Bellut 1990; Zablit and Zimmer 2001; Whelton, Ballard et al. 2002a). Increasing product complexity necessitates a more interdependent decision making process across disciplines and processes (Lindemann, Maurer, et al. 2008; Kreimeyer 2009). This increasing complexity contributes to the difficulties associated with achieving adequate design synthesis and identifying concepts that provide necessary trade-offs based on design objectives. Unfortunately, many computer-aided design and simulation tools cannot be implemented in this phase due to an inherent lack of product and design context data (Yannou 2001).

The issue of product architecture generation has been studied for more than 30 years (Antonsson and Cagan 2001; Chakrabarti 2002). However, research communities only recently have begun to address difficulties related to design synthesis in order to support design engineers in this process (Albers, Braun et al. 2011). Increasing computational complexity creates a need for new design support tools to meet required performance thresholds and conduct necessary uncertainty evaluations in these early stages. Moreover, the innovation integration introduces several types of uncertainties in the design process: uncertainties concerning the design parameters and their levels related to the components, concerning the capabilities of components to satisfy system functions and to reach specified system performances, concerning the component compatibilities.

In this paper we propose a Bayesian network (BN) approach for system architecture generation and exploration (Figure 3-1). First, we propose a Bayesian network model to represent system architecture design. This model represents the designer's knowledge concerning the design problem: related design decisions, possible options for each decision as well as overall constraints to be satisfied. Moreover, uncertainties concerning the problem are also integrated through expert's estimations, in particular those concerning component characteristics and possible interfaces. Second, we develop a system architecture generation and evaluation algorithm for this BN model. An overall architecture confidence level is calculated using uncertainty estimations of experts for component compatibilities. The proposed algorithm is afterwards able to select potential system architectures whose confidence level is greater than a given threshold, representing the risk that a company is prepared to take.

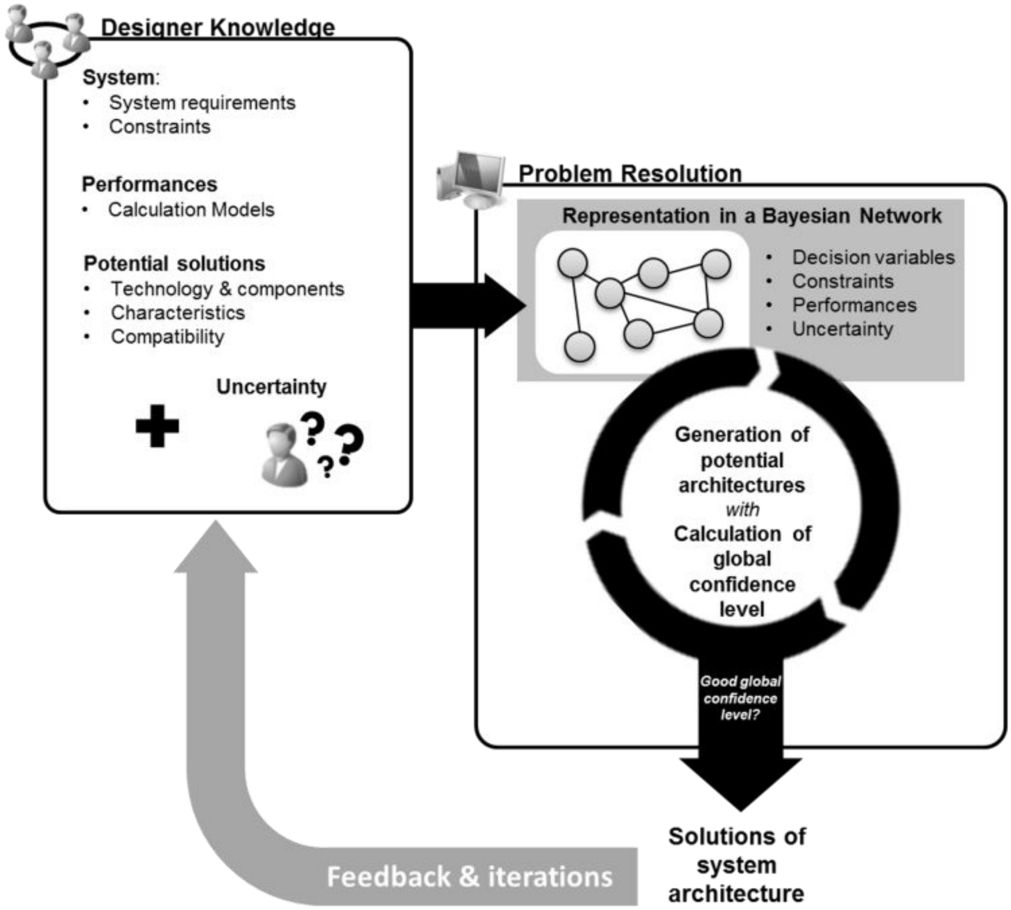


Figure 3-1. Global process of the method

Architecture generation and evaluation issues are discussed through relevant literature review in section two of this paper. In section three, we give an overview of Bayesian networks. Our Bayesian network approach, proposed templates for system architecture design problem and architecture generation and evaluation algorithm are detailed in the fourth section. A case study concerning bicycle design is used in this section to illustrate the concepts. Our approach has been tested and implemented on an industrial problem; in section five, we detail the Bayesian network approach and results concerning the design of a radar antenna cooling system. In the end, we discuss advantages and observed limits of this approach and we give an overview of future developments.

3.2 Background on system architecture and synthesis

3.2.1 Methods and models supporting system synthesis

System synthesis remains challenging (Albers, Braun, et al. 2011). Here we are basing ourselves upon the definition proposed by Cagan et al. (2005): “the algorithmic creation of designs; the organised methodological modelling, implementation and execution of design creation on a computer”. Only a few methods and models are dedicated to supporting it. These methods vary according to: (a) model inputs and data used to support system synthesis, or (b) proposed system generation techniques.

Function-oriented synthesis methods are focused on defining a functional architecture from which structural architecture can be deduced. Kurtoglu and Campbell (2009) and Bryant et al. (2005) enriched functional synthesis based upon the proposition of Stone and Wood (2000), who described functional architecture as a set of basic functions linked together by energy, material and signal flows. Kurtoglu and Campbell (2009) supplemented this definition by developing a design repository with 92 component types and their corresponding functions. A set of graph grammar rules is used in this repository to generate a configuration flow graph from a defined functional model. The generated graph represents components linked by pre-cited flows. Gupta and Okudan (2008) extended this method by proposing a framework that integrates modularization, assembly and variety considerations and yields product concepts with components grouped in modules. Bryant et al. (2005) proposed rule-based repository definitions and explored function/component allocation. They proposed to use the repository with a set of matrices that define a number of function/component allocation rules and compatibility constraints. Potential component configurations result from this concept generation process.

Helms and Shea (2012) chose the Function-Behavior-Structure (FBS) model for concept generation, as proposed by Gero and Kannengiesser (2004). This model has three levels of abstraction correlating to classical design process steps: *functional decomposition*, *allocation of physical effects to function*s, and *embodiment of physical effects with components*. A graph grammar is used to map the initial functional graph into a behavioural graph. The resulting behavioural graph is linked and mapped to structural architecture. The originality of this method lies in the adoption of an object-oriented approach that leads to fewer rules. However, one can discern that neither proposed methods nor any previously cited approach address issues associated with global system performance while satisfying initial system requirements.

However, Albarello et al. (2012) proposed an overall approach based on functional requirements, in which an algorithm generates a random set of functional architectures. Based on defined constraints and viability rules, functional architectures are mapped to structural architectures. The approach supports and integrates performance calculations and architecture adequacy relative to defined preferences. An evolutionary algorithm is used in the process to refine and discover more appropriate architectures. This iterative process stops when proposed architectures achieve sufficient performance. The main difficulty of this approach is that for each system architecture it is necessary to redefine allocations within different domains as well as system evaluation criteria.

Architecture generation problems do not necessarily address only the functions as initial inputs. Some approaches integrate diversified design choices that we refer to as design variables. These design variables mostly concern components or more specific variables such as component characteristics. For instance, Wyatt et al. (2012) developed a method in which designers formally specify their own component alternatives. Structures, behaviours, assignments and geometric relationships complete the design space definition. Based on initial user-defined component architecture, state-space search rules are used to produce product architecture variants by applying simple changes such as adding or subtracting components. Resulting component graphs are then analysed using three architectural metrics in order to identify the most promising concepts.

Another approach, HSoS method (Rosenstein and Reich 2011), uses genetic algorithms to define the design problem. For each decision variable, a set of genes specifies the ability of the decision variable to satisfy diverse constraints and its contribution to fulfilling product objectives. A genetic algorithm searches for solutions with the best performance by satisfying a weighted objective function, and a Pareto front approach allows the best concept to be identified. However, with large search spaces, simulations must be repeated in order to avoid local optimisations in favour of global optimisations.

It is noteworthy that all the methods presented above exploit knowledge about functions and components, but not knowledge related to existing products. However, Matthews (2011) proposed a tool that is able to exploit data from existing product designs in order to create a Bayesian network representing the product design problem. This Bayesian network is used to test new concepts based on the structure and probability tables defined during the automated network building process. The network predicts the performance of new concepts, enabling comparisons between several concepts. The drawback of this method is that it does not support a designer's understanding of the problem given that some resulting relationships between design variables and performance are not necessarily logical from a designer's point of view.

3.2.2 Analysis and remaining challenges

Cagan et al. (2005) described a framework for computational design synthesis and stated that successful methods require integrating four main activities:

- *Representation* captures the attributes of the design space (design alternatives, objectives and constraints are specified);
- *Generation* uses this representation to propose candidate solutions;
- These solutions are subjected to *evaluation* with regard to final objectives; and
- Evaluation provides feedback called *guidance*, which is used to steer the search process in subsequent iterations.

In order to discuss system synthesis methods and identify related challenges and drawbacks, we employ this framework as an analysis and evaluation grid.

Representation, the initial phase in which a design problem is defined, is crucial for system synthesis. A design problem representation must match designer knowledge and support as well as enhance understanding of the overall problem. Indeed, research studies and experiments by Wyatt et al. (2012) and Matthews (2011) show that problem understanding is critical in order for methods to be successful. Additionally, in order to successfully deploy and implement these solutions in industry, it is necessary to identify and integrate industrial constraints. For example, in function-oriented methods the design problem is represented through a functional model and a set of rules that govern function/component allocation. Function-oriented methods aim at exploring the design space as much as possible by using design repositories of concepts that comply with functional requirements. However, when addressing complex system design in industrial environments, the design space could possibly be too large, especially with the additional difficulties associated with eliciting an exhaustive list of design rules and characterising those that are implicit. The number of rules to be elicited may also pose a problem. A design problem describing the design alternatives and constraints and defining design objectives may be more intuitive. Furthermore, the possibility of specifying a design problem by choosing design variables, as proposed in Wyatt et al. (2012) or Rosenstein and Reich (2011), is useful for focusing design space exploration on the problem core. On the other hand, the design space, if only defined by experts, is subjected to design fixations from designers and may be over-constrained or under-constrained with the risk of neglecting some solutions. Wyatt et al. (2012) opted for a hybrid approach, given that the proposed method uses predefined generic rules with limits defined by experts as constraints. However, a potential drawback of this approach is that these constraints, as well as design alternatives, are formally and graphically specified, and fail to consider classical representations used in multidisciplinary engineering platforms such as UML or SysML (Kerzhner and Paredis 2009).

The challenge in the *generation* step essentially relates to design space exploration. Generation methods based on genetic algorithms are based on an initial *population* of random concepts that must be optimised for the function of concept performance. Exploring the entire design space requires several simulations in order to avoid local optimisations. Likewise, graph grammars and state-space search rule based methods are not exhaustive given that they originate from a subjective functional or structural architecture. As several functional/structural architectures can fulfil the same product requirements, we can possibly make an assumption that some design regions are not explored. On the contrary, the tool proposed by Matthews (2011) constructs a discretised design space in which new concepts are tested manually. However, automating this process could facilitate exploration of the entire design space. Thus, the scale issue must be considered in the representation step.

As for the *evaluation* step, approaches proposed both by Rosenstein and Reich (Rosenstein and Reich 2011) and Albarello and Welcomme (2012) used evolutionary algorithms. The first approach evaluates them qualitatively by means of a weighted function, allowing a relative comparison between candidate architectures. However, this approach does not ensure or estimate the likelihood of solutions to fulfil system requirements. In contrast, Albarello and Welcomme (2012) evaluated concepts with regard to performance and designer preferences, but did not precisely describe how this model is developed and linked to the proposed architectures. Moreover, the functions, components and their interrelations (which were declared as inputs) were already well defined and known, which is not always the case in conceptual design. In preliminary design, limited information about future components or working principles is available or usable. That is why most of the presented methods do not consider quantitative data. For instance, parametric information and sizing were not considered by Helms and Shea (Helms and Shea 2012). Likewise, Bryant et al. (2005) supported product architecture generation without integration of concept evaluation and/or ranking. In the approach proposed by Wyatt et al. (2012), candidate architectures are evaluated using three metrics based on component graph analysis without considering product requirements. This potential drawback in candidate solution evaluations could turn out to be important since this last step is the foundation for architecture comparison and selection. Moreover, without any evaluation, the *guidance* step in Cagan's framework cannot be implemented.

With regard to this discussion, it is important to highlight that system architecture evaluation provides essential support to designers in the conceptual design stage. Although only limited, vague information about design decision variables is available during preliminary design, it is important to allow designer expression, particularly when considering new technologies. This fact implies taking into account the uncertainty associated with expressed data and estimating the overall uncertainty of each solution as an indicator that will influence the final architecture selection. Therefore, in order to address this issue, the major aim of our research is to support complex system design integrating sometimes hundreds of decision variables, and the possibility of enabling a designer to focus on a specific decision variable is interesting. Finally, depending on the degree of uncertainty in an early design, a fully automated method may not be an appropriate solution to this problem. In light of this analysis and discussion of the literature and an audit that we made in industry, we argue that system architecture design support should satisfy at least the following four requirements:

1. Design problem representation support should permit the integration of designer opinions;
2. It is necessary to support uncertainty management in both the generation and the evaluation of system architectures;
3. Exhaustive solution space search mechanisms are necessary and ensure design solution consistency;
4. Adequate system architecture comparison and evaluation should be integrated into the overall approach.

Based on these four requirements, this paper proposes the use of BNs for system architecture generation and evaluation.

3.3 Bayesian Networks

In an effort to support product architecture generation while addressing weaknesses in pertinent earlier research, we have opted to use Bayesian networks (BNs) as a base for model construction. This choice is supported by the fact that BNs are particularly adapted for integrating data uncertainty into a modelling structure, good data visualisation as well as combining quantitative and qualitative data (Barton, Saloranta et al. 2008).

A Bayesian network is a probabilistic graphical model that represents a set of random variables and their joint probability distribution. Formally, a Bayesian network is a directed acyclic graph in which nodes represent variables, and edges represent conditional dependencies. The edges pointing at one variable node are associated with a conditional probability distribution defining how this variable depends on its parents. Most uses of BNs rely on what is called "inference". When the values of some observed variables are known, inference consists in updating the probability distribution of the unobserved variables. This inference is an application of Bayes' theorem:

$$P\ (A|B) = \frac{P(B|A)*P(A)}{P(B)} \qquad (1)$$

Figure 3-2 is a screenshot of the tool Netica[1] showing how, in the "Chest clinic" model (this is a famous pedagogical example in the literature on Bayesian networks) the tool has updated the probability distributions of all the remaining nodes, knowing that a patient has the symptom dyspnea, and that he does not smoke. Given this evidence, it is much more probable that the disease explaining the dyspnea of the patient is bronchitis, rather than tuberculosis or lung cancer.

This simple example illustrates various characteristics of both Bayesian Networks and software used to exploit them. The structure of the graph gives a global and intuitive view of the dependency structure between variables, but it hides completely the (maybe complex) information contained in the conditional probability tables. The kind of representation used by Netica and several other software, with "belief bars" whose length is proportional to the probabilities of the various states of variables, is convenient both for visualisation of inference results and for inputting observations. Although in the example of Figure 3-2 all variables have only two states, in the general case, variables can have any finite number of states. In theory, variables could also be continuous; however, inference algorithms giving exact results can only work on discrete variables with a finite number of states (this is why in the sequel we will always discretise continuous

[1] Netica is the tool that we have chosen to set up our method, mainly because of the following characteristics: efficient graphical user interface, good and powerful syntax to define deterministic nodes via equations, existence of a programmable interface (See: http://www.norsys.com).

variables[2]). Evidence can be input in any node of the network: the inference mechanism updates the probabilities of all nodes regardless of the links' directions. Thus, it is possible to use the same model either to propagate information from causes to effects (to do "what if" experiments on the model) or from effects to causes (to perform diagnosis). The probability distributions calculated by inference are, in most cases simply marginal distributions. This is why it is possible to represent simultaneously the distributions of all nodes on a single figure. Fortunately, this kind of result is sufficient in most applications of BNs.

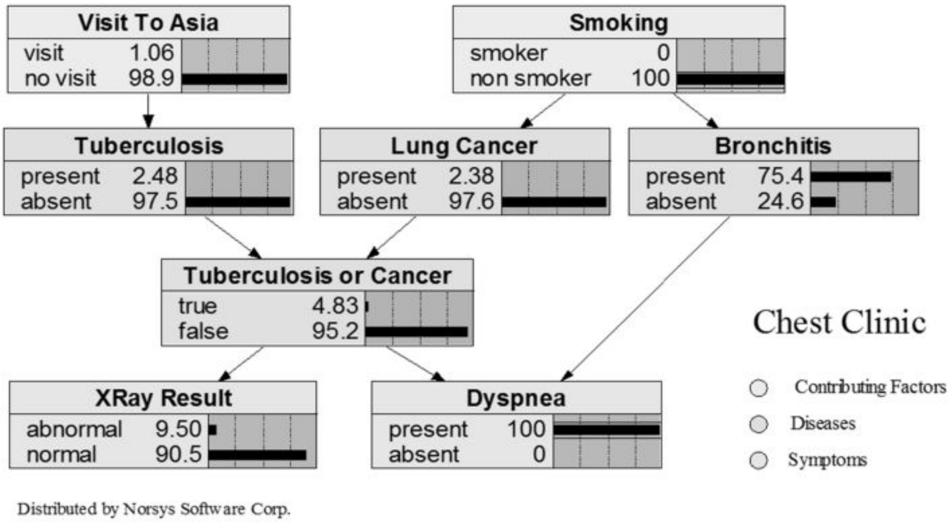


Figure 3-2. "Chest Clinic", a classical example in the Bayesian networks literature

Bayesian networks are used in many domains: marketing, intelligent man-machine interfaces, risk engineering, finance, medicine, etc. There are many inference algorithms. This is a complex topic, which itself is a field of active research. Further information about the use of BNs is available, for example, in Jensen and Nielsen (Jensen and Nielsen 2007).

3.4 Towards a Bayesian Network Based Approach for System Architecture Generation and Evaluation

3.4.1 Architecture design problem definition

Scaravetti (2004) studied the most relevant design process definition, both given in different research works and observed in industry. The results of his study show that although order of the steps can vary, a general design process can be identified: system design begins with *system functions* specification which leads to the

[2] Discretization of continuous variables requires particular attention: see Appendix for a detailed explanation of issues related to discretization.

elicitation of *system requirements*, expressed in terms of *objectives* and/or *constraints*. A second stage consists of defining a functional architecture and finding potential principle solutions, and then combined in concepts. After evaluating and comparing these concepts, the most promising one is selected to be turned into a structural system architecture: regarding global system performances, the *main system components* are chosen, as well as their physical arrangement and their sizing *characteristics*. Finally, determination of main *system dimensions* is the last step of system architecture definition. To achieve a good solution, this process is reiterated many times.

Chandrasekaran (1990) sums up the process described above in a single and generic definition of design problem : *"A design problem is*

1. *a set of functions to be delivered by an artefact and a set of constraints to be satisfied,*
2. *and a technology, that is, a repertoire of components assumed to be available and a vocabulary of relations between these components."*

Based on this definition, we propose a Bayesian network model in which designer specifies a repertoire of key components as well as their relations. The choice of these key components is motivated by their consistency regarding system functions (the functional architecture is assumed to be already determined). All component characteristics involved in component compatibilities and performances calculation are identified and described through discrete or continuous variables. Component placement choices may also be included. Afterwards, designers establish the complete set of constraints that ensure the system requirements fulfilment. Finally, designers have the possibility to characterise the uncertainty associated to some of component compatibilities, if necessary. In the second step, the architecture generation algorithm only proposes the solutions that, based on the information entered in the model, potentially satisfy system performances and requirements.

3.4.2 Case study: Bicycle design

In order to illustrate and test the architecture generation method we address a bicycle design problem and consider several types of components and compatibilities. In most cases, companies try to minimise mass when designing a bicycle. Therefore, the objective of modeling is to generate all possible bicycle configurations weighing 15 kg at most. In early design stages, potential compatibilities between product components are uncertain. Moreover, the decision variable characteristics are approximations. Thus, the mass of each component is known as a range of possible values rather than as a fixed value. In this case, we consider two types of uncertainties: those pertaining to component data and those concerning the feasibility of the architecture solution. A global overview of the design problem is provided in Table 3–1.

Table 3–1. Design Problem Specifications

9 decision variables	Rack *(2 instances)*
	Brakes *(4 instances)*
	Gears *(3 instances)*
	Transmission *(3 instances)*
	Frame *(4 instances)*
	Fork *(2 instances)*
	Rear suspension *(4 instances)*
	Front suspension *(4 instances)*
	Lighting *(3 instances)*
8 characteristics	Mass of each instance of every decision variable (except gears) in kg
	Additional mass (in kg) that represents other bicycle components
3 constraints	Rack – Brakes compatibility
	Brakes – Lighting compatibility
	Lighting –Transmission compatibility
4 constraints with uncertainty: defined by the probability of the compatibility to be true (in %)	Frame – Fork compatibility
	Frame – Rear suspension compatibility
	Fork – Front suspension compatibility
	Gears – Transmission compatibility
2 performance criteria	Total mass < 15kg
	Feasibility > 90%

In this case study, we have decided to consider only compatibility constraints, although in reality the architecture of a product might also depend on the number of components, as well as their relative locations. Moreover, constraints could address some component characteristics and be coupled. Nevertheless, this example was chosen only for the sake of model and method description. A complex system example will be introduced and discussed later.

A BN-based architecture generation and evaluation approach consists of two main steps: (a) *system architecture modeling* and (b) *system architecture exploration and clustering*. The first step consists of building a BN that encodes all the decision variables of the design problem, the compatibility constraints existing between them (either crisp or uncertain), and performances as functions of the decision variables. In order to support design engineers in this phase, we have developed templates that are used in this architecture-modeling step. These templates are explained later in the paper. The second step of the proposed method concerns architecture exploration and evaluation of overall confidence, depending on all kinds of uncertainties. For this purpose, an algorithm is proposed that generates all possible architectures with regard to defined constraints. Moreover, an overall confidence level is calculated for each candidate architecture and used to cluster and rank acceptable solutions.

3.4.3 Templates for architecture modeling

To facilitate Step 1 of the proposed method, templates have been developed to describe the architecture design problem by defining the decision variables and constraints to be satisfied. Two types of nodes are used in this architecture modeling: *chance nodes,* whose relationships with parent nodes are probabilistic, and *deterministic nodes*, which are simply functions of their parents' values. Both types of nodes can be Boolean, discrete or continuous. A dependency relationship between two elements is represented by an edge. The probabilistic distribution describing this dependency is defined by the user with a table or an equation directly entered during the node definition. Therefore, product architecture modeling can be represented by (Figure 3-3):

1. Decision nodes,
2. Characteristic nodes,
3. Performance nodes,
4. Constraint nodes,
5. Constraint nodes with uncertainty, and
6. One global confidence node.

In order to design the network, the designer has to identify decision nodes. Appropriate characteristics nodes are identified by designer according to system performances. Afterwards, constraint nodes are identified and defined between decision and characteristic nodes. In the end, in view to the problem, global confidence node is defined with regard to overall design project risk.

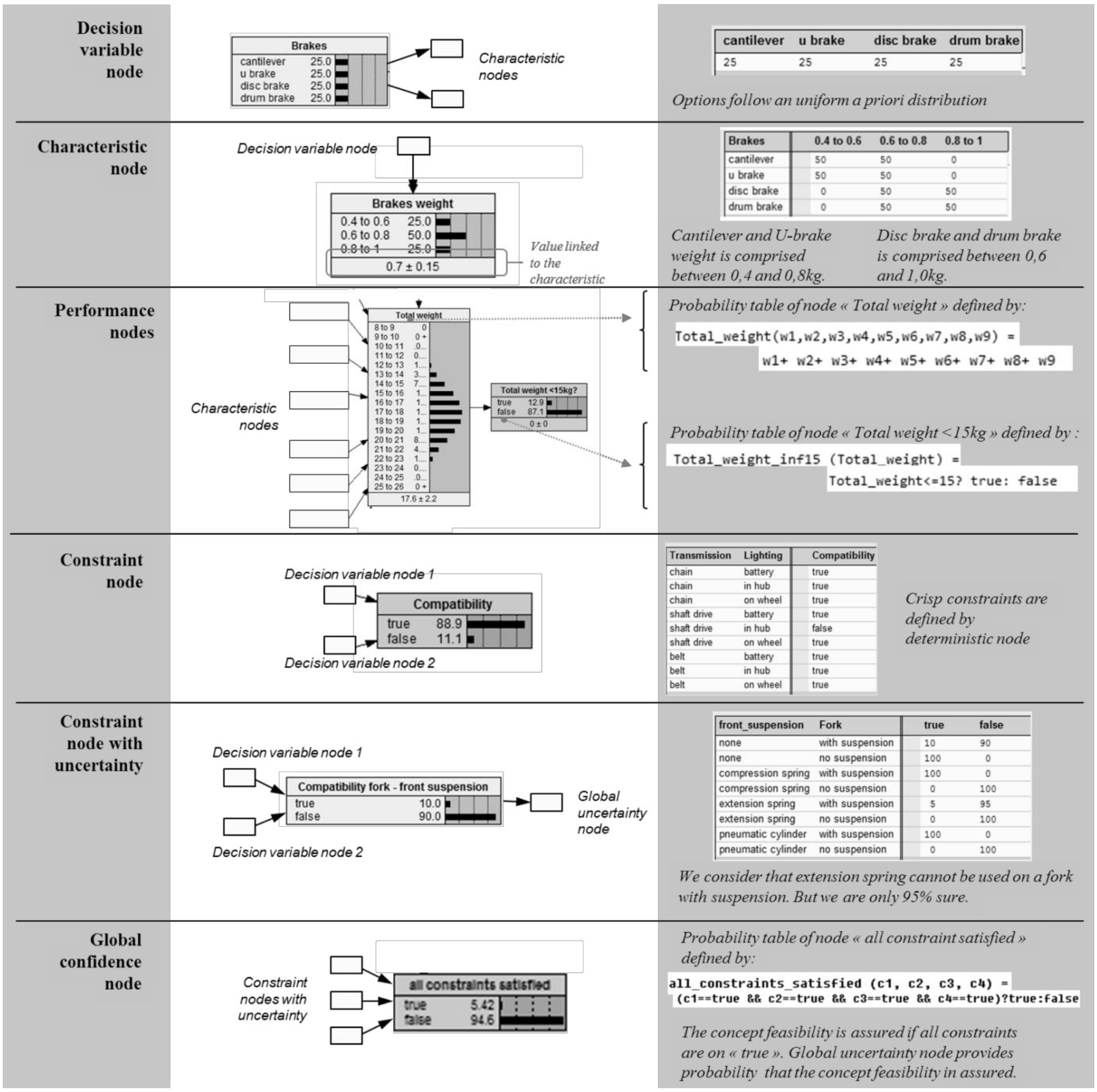


Figure 3-3. Product architecture modelling templates (probabilities are given in %)

A *decision variable node* is a chance node representing all possible product components. Modalities of this node are mutually exclusive possibilities for each product component. A priori, all options can be considered. The probabilistic distribution of the node is therefore assumed to be uniform.

A *characteristic node* represents design parameters that are inherent to different product components, i.e., decision variable nodes. A characteristic node must be represented as a chance node that is linked with the decision variable node that it describes. In order to represent the coupling effect between design parameters, two characteristic nodes can be linked. A numeric value can be associated with each characteristic node, representing global requirements related to this design parameter. This value is used for the global architecture performance calculation. In our bicycle example, we need to estimate the total weight of the bicycle: to represent the brake weight, we define a continuous node that has been discretised and we link it to the node "brake." The corresponding probability table is shown in Figure 3-3.

An *architecture performance* is represented by association of two nodes. The first node represents the performance calculation at the architecture level. This node is linked to characteristic and/or decision variable nodes. The second node is Boolean and defines the performance threshold for the product. If performance is over the threshold, its state is "true." If not, its state is "false." This node is used also in the generation step for architecture exploration and clustering. Several performances can be represented in the model. All architectures must have a weight of less than 15 kg.

Constraint nodes express experts' knowledge on compatibility constraints within the architecture. They can be certain when information or knowledge is available or include uncertainty when integrating new technologies and innovations. If the knowledge is certain, then a designer can use *constraint nodes* that are defined as deterministic Boolean nodes. They are linked to different decision variable nodes and/or constraint nodes and therefore represent the constraints between these architecture components. If the constraint is satisfied, the node state is "true." In our example, we define compatibility constraints between components: for instance, a dynamo hub is not compatible with a speed hub; in the line of the probability table corresponding to this combination, the constraint state is "false." In order to express uncertain knowledge concerning compatibility constraints, we propose a *constraint node with uncertainty*. This node is a chance node and represents expert estimations about architecture component compatibilities. For instance, a shaft drive is compatible with a speed hub, but since we have never designed a bicycle equipped with a shaft drive, we are not completely sure of that fact. Therefore, we assign a compatibility probability of 0.9.

A *global confidence node* represents overall architecture uncertainty, and therefore the degree of confidence that designers can place in the architecture solution. This node is a child node of all constraint nodes that include uncertainty. It is used in the second step to cluster all feasible architectures with defined or required confidence levels.

3.4.4 Product architecture generation algorithm

The second step concerns *product architecture generation and exploration*. Product architectures are generated based on satisfaction of all constraints and performance thresholds. The proposed algorithm is provided in Figure 3-4. Table 3–2 defines the terms which are specific to Bayesian networks.

Table 3–2. Table of definitions

Term	Definition
Input evidence	"input evidence *n = s*" means that we make the hypothesis that the variable represented by the node *n* equals the value represented by the state *s*. Then, the probability $Pr(n=s) = 1$. By convenience, we will sometimes say that we "block the node *n* to the state *s*".
Finding	State of a variable such that, after inference, the probability of this state is equal to 1. Note that an evidence is a special case of finding.
Belief	Probability that the node *n* has the value *s* : $Pr(n = s)$ (after inference)

```
N = the set of decision variable nodes
C = confidence threshold                          // in [0;1]
forall n in constraint nodes do
    input evidence n = "true"
end forall
update beliefs of all nodes                       // Bayesian inference (use of a built-in Netica function)
call generation (N,C)

// Here is the definition of the function "generation" called in the main program
function : generation (variables, c)
    variables : set of nodes
    c : real in [0,1]
    sort variables according to their number of possible states (i.e. with a belief > 0)
    select the first variable v in the sorted list
    forall s in states of v do
        if Pr(v=s) > 0 then                       // Select only the node states whose belief > 0
            input evidence v = s
            update beliefs of all nodes           // Bayesian inference (use of a built-in Netica function)
            if cardinality(variables) > 1 then
                generation (variables-{v}, c)     // recurrence on the nodeset from which v is removed
            else      // termination of the recursion
                if Pr(global_confidence_node = "true") > c then   // The current state of the BN depicts an admissible solution
                    forall u in N do
                        write finding for u                       // This solution is printed
                    end forall
                end if
            end if
        end if
    end forall
end function
```

Figure 3-4. Product architecture generation and exploration algorithm

The proposed algorithm is comprised of three main steps. First, for all constraint nodes that do not include uncertainty, the evidence "true" is input. This ensures that all combinations respect constraints defined with certainty and performance thresholds. Second, node after node and state after state, decision variables must be blocked to a given state. Only the states with strictly positive beliefs are examined. BN exploration is performed through a recursive function. In order to minimise the solution space exploration, variables are chosen according to their number of possible states: a variable with the smallest number of possible (i.e. with a non-null belief) states is chosen first. This simple heuristic reduces the number of intermediate branches in the exploration tree, while ensuring the exhaustiveness of the obtained solutions[3]. Third, when all decision variables have been processed, the belief of the global confidence node is compared to the threshold C. If this belief is higher than C, the findings of all decision variable nodes constitute a solution which is printed in the output file of the program.

The bicycle architecture model (Figure 3-5) is composed of nine decision variables, each with two to four options. They are described by eight characteristic nodes related to a single performance node. Seven compatibility constraints, among which four include uncertainty, are applied on decision variables. The

[3] We have experimentally checked the efficiency of this heuristic. However one cannot expect a drastic complexity reduction, whatever the exploration strategy chosen, because the final number of solutions, which are the leaves of the explored tree, is fixed and depends only on the characteristics of the design problem.

Cartesian product of the decision variable state sets encodes 7,648 possible bicycle architectures. Modelling this bicycle design problem using our proposed model took 2 hours.

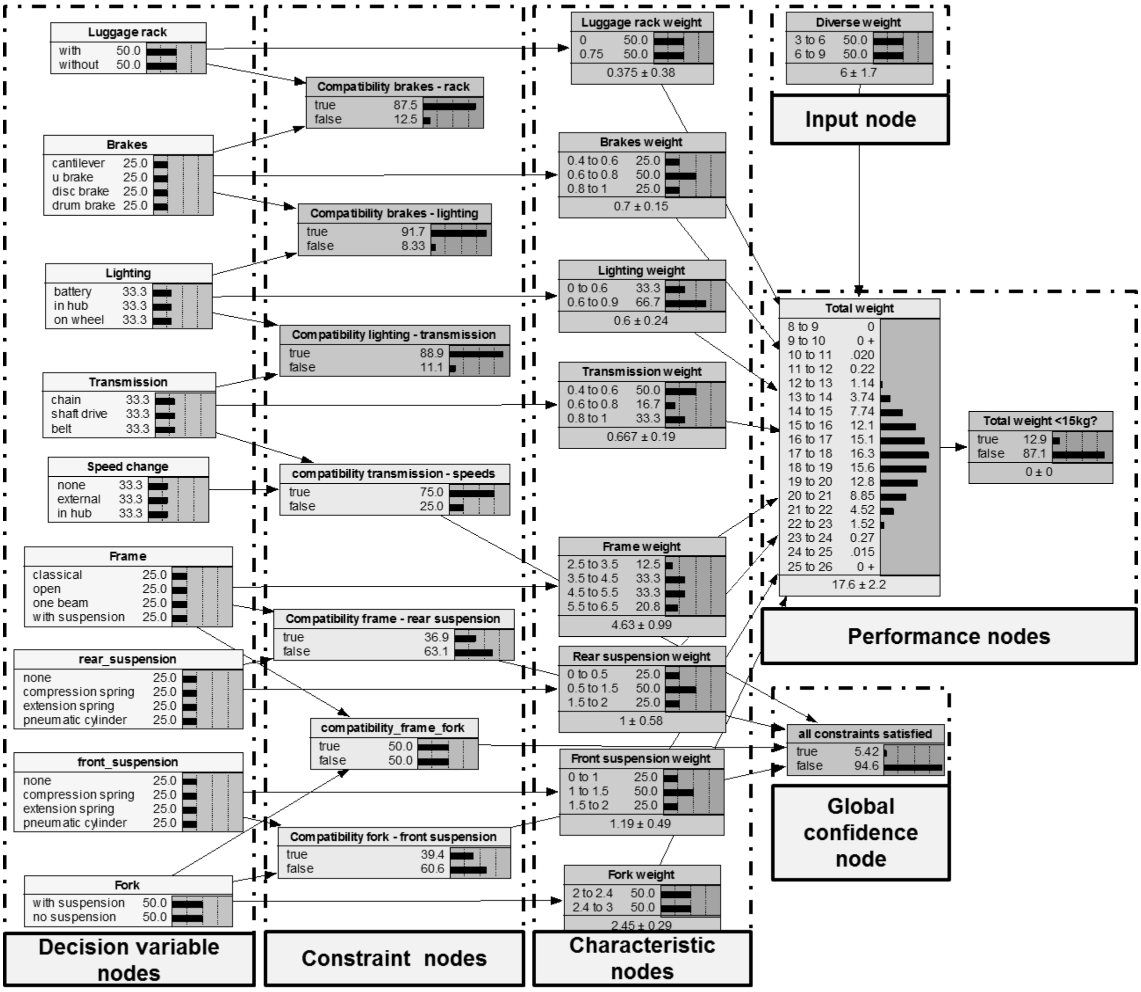


Figure 3-5. Bayesian Network bicycle architecture model

If the confidence threshold is defined to be 90%, in the second step (product architecture generation), 133 architectures are generated in a few seconds on a standard desktop computer with a confidence level greater than the confidence threshold (90%). The global confidence level is used to cluster the generated architectures (Figure 3-6); the number of acceptable architectures drops as the confidence threshold grows. Therefore, at a 70% global confidence threshold there are 899 bicycle architectures possible and at a 40% global confidence threshold there are 1,107 architectures possible.

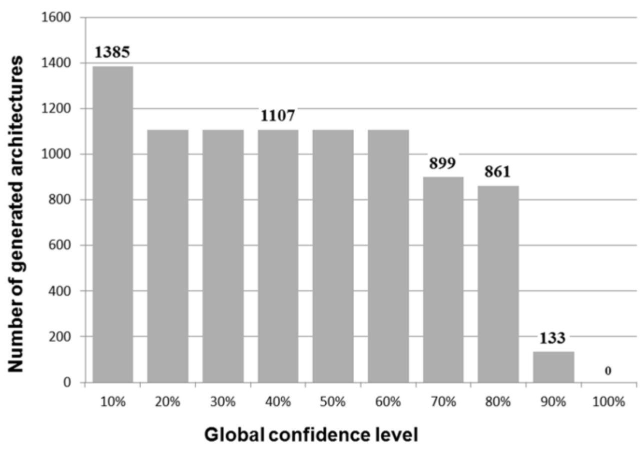


Figure 3-6. Bicycle architecture clusters based on global confidence level

3.5 Industrial Case Deployment: Antenna Cooling System

When addressing architecture generation and evaluation issues, it is important to verify and test proposed approaches on large scale problems. We used the BN approach for architecture generation and evaluation for an antenna cooling system in an industrial environment. The basic purpose of an antenna is transmitting and receiving electromagnetic signals in order to detect the presence of objects in a given area. These functions are fulfilled by mechanical and electronic devices. However, in order to be usable, the transmitted signals must be amplified before being radiated, and the received signals must be amplified before being sent to the calculation system. An active antenna amplifies the signals between the radiating elements and their inputs (Figure 3-7).

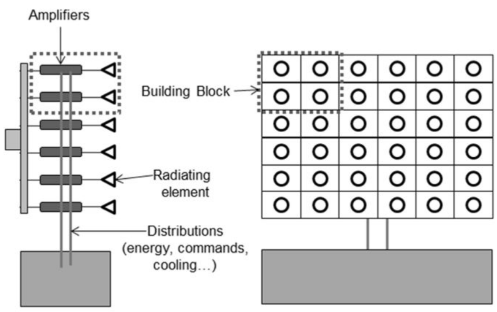


Figure 3-7. Active antenna description (adapted from(Roger 1999))

Designing such an antenna is extremely complex and costly, in particular because amplifiers, which heat up the most, are located in the antenna and require an efficient cooling system. Designing an active antenna cooling system is a long process because of the number of performance considerations as well as the number of potential architectures. The main decision variables for this design problem were:

- Cooling components: their type (mainly heat exchangers), number and placement;

- Fans: their type and placement.

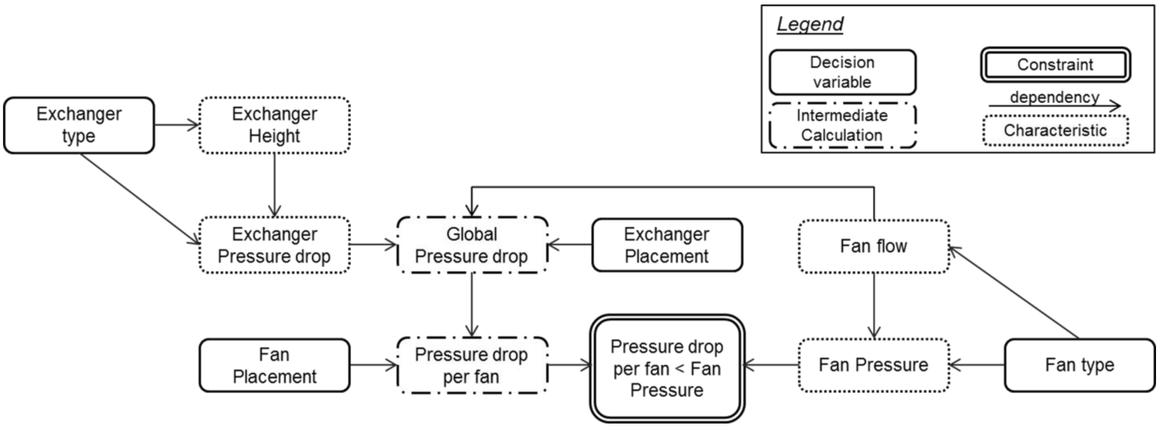


Figure 3-8. Interdependencies between variables

The difficulty of the architecture design problem lies in choosing adequate product components based on different interdependencies between performance, components and their characteristics. For example, the height of the heat exchanger will have an influence on its pressure drop. This pressure drop also depends on the circulating flow, and its value is needed in order to calculate the entire pressure drop that must be compensated for by the fan pressure. Finally, the fan pressure depends on the type of fan, which also determines the flow that can be provided (Figure 3-8).

The aim of modelling is to identify all the feasible solutions of the antenna cooling system. As this problem is highly constrained, other performances like cost or global volume are not considered.

3.5.1 Problem definition

The data gathered for this case study were collected from experts in their respective fields with considerable experience (15-20 years) in this type of design. Based on the current industrial design of antenna cooling systems, we consider the system to be a net composed of several identical electronic modules that we call building blocks (BBs) which are lined up in columns and rows (see explanation of the principle in Figure 7). As inputs, we know the number of BBs, their placements and dimensions. Each BB dissipates a power P_{BB} under the form of heat that must be evacuated by the cooling system. Otherwise, the BBs would overheat and become out of use. Thus, the air temperature in the BBs T_{BB} must not exceed a maximum temperature T_{max} determined by experimentation. This is the main feasibility condition of the system. Another condition of system feasibility is the ability of the cooling system to be integrated into the antenna system: its dimensions must not exceed a maximum height H_{max}, a maximum width W_{max} and a maximum depth D_{max}. The last feasibility condition to be considered is that it must provide a level of pressure P_{system} sufficient to compensate for the pressure drop dP_{system} induced by the BBs and the cooling system itself. The architecture design problem is summarised in Table 3–3.

Table 3–3. Architecture Design Problem for the Cooling System

Decision variables	Cooling component type	6 possible types
	Cooling component number	1 or 2
	Cooling component placement	3 possible configurations
	Fan type	3 possible types
	Fan placement	2 possible configurations
Inputs	Number of BBs	N_{BB_row}; N_{BB_column}
	Placement of BBs	Distance between two BBs
	Heat power dissipated per BB	P_{BB}
Feasibility conditions	Minimum pressure to provide	$P_{system} > dP_{system}$
	Maximum temperature of a BB	$T_{BB} < T_{max}$
	Cooling system maximum dimensions	$D_{system} < D_{max}$

3.5.2 Bayesian network building

In summary, in the design of an antenna cooling system, the feasibility of potential architectures is ensured by 3 conditions:

1. System pressure P_{system} must compensate for the overall system pressure drop dP_{system}: $dP_{system} < P_{system}$;
2. A BB's temperature T_{BB} must not exceed T_{max} : $T_{BB} < T_{max}$;
3. The height H_{exch}, the width W_{exch} and the depth D_{exch} of the cooling elements, i.e., heat exchangers, must not exceed H_{max}, W_{max} and D_{max}, respectively.

These three conditions represent the starting point for the proposed methodology to build the BN architecture model, to define decision variables and characteristics. The modelling process employed for the creation of this network, from condition 1 to the decision variables definition as well as the definition of constraints, is the following (Figure 3-10):

1. **Creating decision variable nodes**. Decision variables are known a priori and created using templates proposed in the BN method (see Section 4.3). The probability distribution is uniform.
2. **Creating feasibility constraint node.** As shown in Figure 3-8, dP_{system} can be represented by $dP_{system/fan}$, corresponding to the overall system pressure drop that needs to be compensated by each fan. Therefore, three nodes are created: (1) the Boolean constraint node *C1* that will define that fan pressure P_{fan} must be higher than $dP_{system/fan}$, and its parent nodes (2) $dP_{system/fan}$ that is a deterministic continuous node used for intermediate calculation, and (3) P_{fan}, that represents the pressure provided by each fan, and that is thus a characteristic node. At this step, the two nodes $dP_{system/fan}$ and P_{fan}, that are continuous values, are discretised in a convenient way (see Appendix).
3. **Defining the probability table** of each node.

a. constraint node $C1$: the logical relation (2) is used;

$$\textbf{if } P_{fan} > dP_{system/fan} \textbf{ then } C1=true \textbf{ else } C1=false \quad (2)$$

b. characteristic node Q : Q only depends on the type of fan. The characteristic node Q is created and linked to Fan_{type}. Designers, who know current industrial capacities, directly define the probability table of node Q. For instance, if industrial capacities are known as :

- Fan_{type} = type 1 => $Q \in [Q1;Q3]$;
- Fan_{type} = type 2 => $Q \in [Q2;Q4]$;
- Fan_{type} = type 1 => $Q \in [Q1;Q4]$;

such as Q1 < Q2 < Q3 < Q4.

Then, Q is decomposed in 3 states: [Q1 – Q2], [Q2 – Q3] and [Q3 – Q4] and for each type of fan, probability distribution is defined as uniform on possible states (Figure 3-9).

Fan_type	Q1-Q2	Q2-Q3	Q3-Q4
type 1	50	50	0
type 2	0	50	50
type 3	33.333	33.333	33.333

Figure 3-9. Probability table of node Q

c. characteristic node P_{fan}: P_{fan} depends on both the decision variable Fan_{type} and the flow Q provided by the fan. The characteristic node Q is then created, and linked to Fan_{type} and P_{fan}. The definition of P_{fan} probability table requires more data treatment than Q probability table: performance curves given by the suppliers specifications are used to build a linear regression model:

$$P_{fan} = f(Fan_{type}, Q) \quad (3)$$

The direct introduction of (3) in the definition of the node P_{fan} is possible, but this might introduce an error due to the necessary approximation. In order to address this approximation, we define a Gaussian probability $\mathcal{N}(\mu, \sigma^2)$ in which μ is the expected performance, for example $f(Fan_{type}, Q)$, and σ is the standard deviation, i.e. the average error. This means that, rather than having a deterministic relation represented in (3), we define a probability for P_{fan} to be included in the interval $[f(Fan_{type}, Q)-\sigma ; f(Fan_{type}, Q)+\sigma]$. This fact is concretely represented by this function within the appropriate node:

$$p(P_{fan} | Fan_{type}, Q) = \mathcal{N}(f(Fan_{type}, Q), \sigma^2) \quad (4)$$

d. <u>intermediate calculation node $dP_{system/fan}$</u>: $dP_{system/fan}$ is defined through intermediate calculation of dP_{system} and the fan placement $Fan_{placement}$. Therefore $dP_{System/fan}$ is linked to a new deterministic node dP_{system} that calculates the overall system pressure drop, and the decision variable $Fan_{placement}$. This dependency relation can be modeled as a linear equation varying in function of $Fan_{placement}$. Supposing that $Fan_{placement}$ has 3 different states, the deterministic function defining the node $dP_{system/fan}$ is :

$$\begin{array}{llll} \textbf{if} & Fan_{placement} = \text{st1} & \textbf{then} & dP_{system/fan} = \text{a}^{*} dP_{system} \\ \textbf{else if} & Fan_{placement} = \text{st2} & \textbf{then} & dP_{system/fan} = \text{b}^{*} dP_{system} \\ & & \textbf{else} & dP_{system/fan} = \text{c}^{*} dP_{system} \end{array} \quad (5)$$

4. **Converting relations/equations into probability tables.** Once all nodes are completely defined, the equations are converted into probability tables using the appropriate function of the software Netica.

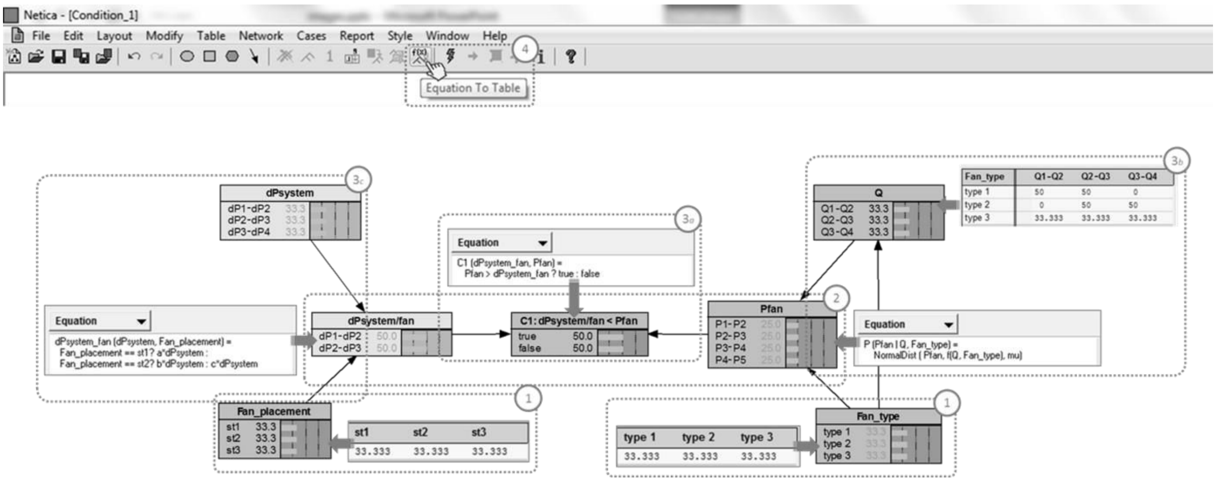


Figure 3-10. Modelling process of Condition 1

3.5.3 Final model and Results

Excluding additional time related to research on the heat exchangers and fan characteristics, modelling the cooling system design problem took about 20 hours. The resulting model is a net comprised of 87 nodes, including:

- 7 decision variable nodes;
- 17 characteristic nodes;
- 30 constraint nodes;
- 33 intermediate calculation nodes or constant nodes.

An overview of the model is presented in Figure 3-11. It is noteworthy that even if the problem contains only seven decision variables, the complexity of the cooling system design is due to the intricate relationships among all of them. It also appears that there is no node with uncertainty, and therefore, no global confidence node. This is due to the fact that designers were experienced and needed to generate concepts based on already well-known technologies.

However, building this network represents a complex task. To obtain the final network, several iterations are necessary. Initial adopted logic for the network development needed to be adjusted. We have observed that starting from the decision nodes and then identifying characteristic and constraint nodes without taking into account the performances, yielded in the inadequate characteristics definition. Therefore, we recommend that firstly required performances be defined in order to identify with precision characteristics node and more importantly appropriate links between these nodes. Identification of all the constraints requires an incremental process: if the resulting solutions are recognised as unfeasible by the experts, then there is a need to enrich the design problem through integration of additional constraints. Finally, filling of all the probability tables can be burdensome. Nevertheless, it is significantly facilitated by using logical relations and mathematical relations for the definition of probability tables.

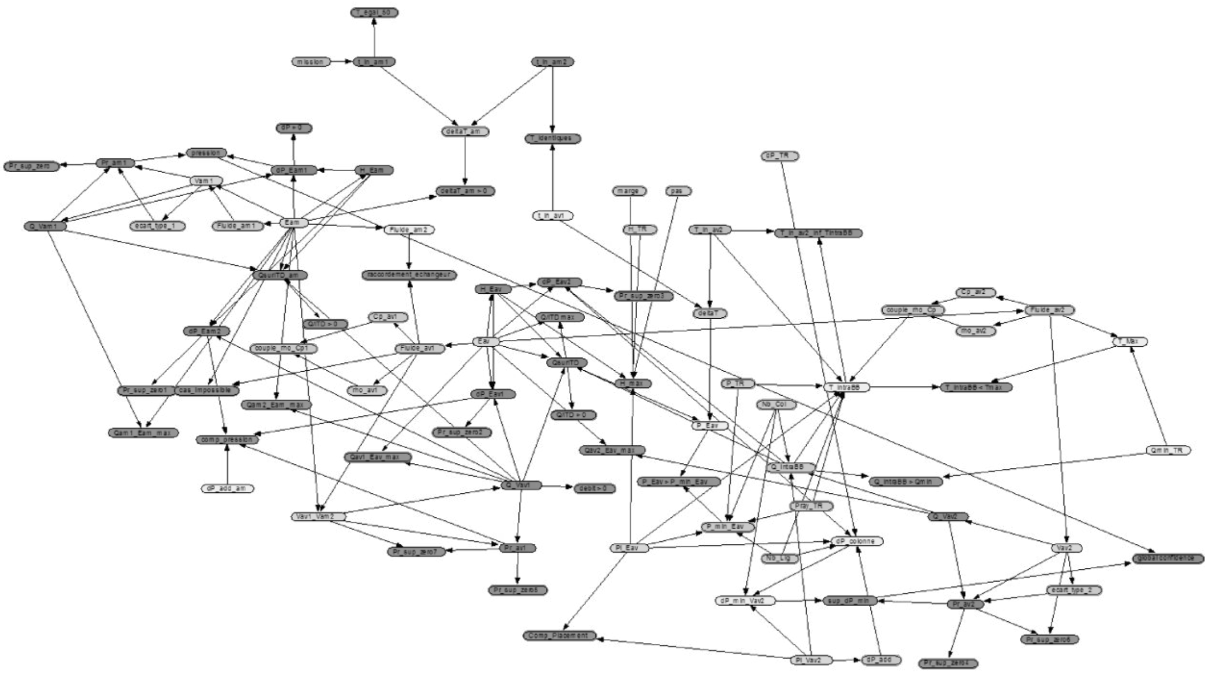


Figure 3-11. BN network for antenna cooling system design

This design problem was initially addressed without any formal method. Experts first elicited 12 global architecture solutions. These architectures were chosen based on expert consensus. Among those, only four solutions were iteratively selected, analysed and abandoned if one of the three feasibility conditions was not fulfilled. The final result of this process was a single solution.

Afterwards, the BN model was used to generate all possible architectures. Among the 5,184 possible combinations for the set of decision variables, four feasible solutions were found in less than 1 second.

Moreover, the proposed algorithm provided an overview of the expected characteristics associated with each solution. For instance, using an axial fan was allowed only if its flow Q and its pressure P_{fan} fell in the range of values indicated for the solution. One must notice that several sets of characteristics may correspond to a same solution, which means that there is still a degree of freedom after the choice of decision variables. As shown in Figure 3-12, out of twelve initial solutions, experts investigated four further in detail. After feasibility study, only one has been considered as feasible. This solution is represented in Figure 12. However, our approach identified three additional solutions that have not been considered by experts.

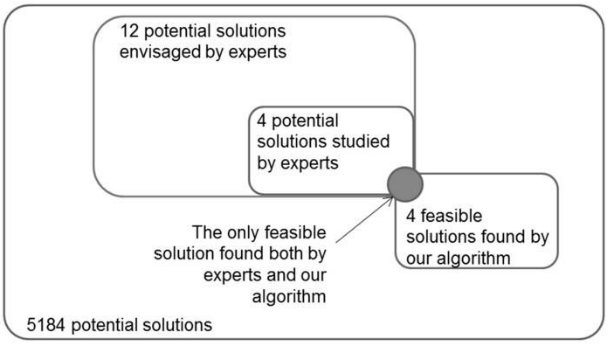


Figure 3-12. Scope of solutions envisaged by experts and our algorithm

Using the BN approach for architecture generation and evaluation for an antenna cooling system yielded several advantages:

- Time: 20 hours (modelling time) and less than 1 second (generation time) for BN vs. 1 month for experts;
- Better understanding of the design problem through a concise representation;
- Exploration of the entire design space with BN (5,184 possibilities) instead of a heuristic and uncertain selection of architectural concepts by experts (12 concepts);
- The consideration of performance requirements in the proposed solutions;
- For each decision variable in each solution, the specification of its characteristics;
- Knowledge capitalisation through the Bayesian network building.

3.6 Discussion

Development of the antenna cooling system showed the potential of our proposed BN model. Even with a large number of nodes (87 nodes), the actual design problem was modelled in drastically less time (20 hours vs. 1 month). Nevertheless, this time does not take into account the time spent for gathering data in order to build linear regression models. Moreover, the relation between several parameters sometimes is not well known and cannot be described by a rough model. In these cases, use of surrogate models could be substituted by importation of Bayesian networks which are automatically built based on existing designs following an approach similar to the one proposed by Matthews (2011).

A strength of this method is its ability to deal with intervals as well as probabilities. Aughenbaugh and Paredis (2006) outlined the sources and role of imprecision in engineering design while emphasising the need of new practicable methods, able to support set-based design approaches. We think that BN method constitutes a potentially interesting answer to this question. The first application to an industrial problem gave encouraging results and also provided insights for future improvements:

- Discretisation of intervals requires attention in order to guarantee the model robustness. At this moment, model robustness is based on expert verification of the equations and node relationships, as well as the simulation of proposed solutions in an external calculation model. Additionally, model robustness is improved when integrating the error induced by surrogate models using Gaussian probability distributions. However, the consistency of generated architectures relies on a convenient discretisation of intervals. Some guidelines and/or tools are required to support this discretisation.
- This industrial case also showed that the complexity of the Bayesian network can grow quickly; we identify two main issues: graphical and combinatorial. Concerning graphical issue, we think that we should examine other types of graphical representations beyond a single, large graph. Concerning the combinatorial issue, a particular attention is required to only focus on decision variables that impact the most architecture solutions and performances. Tractability limits for this method are difficult to estimate because the complexity of inference in a Bayesian network essentially depends on the size of conditional probability tables and on the number and configuration of cycles in the structure (a cycle corresponds to the existence of two different paths starting from a given node and arriving at another node). Therefore it is possible to have very large networks (with thousands of nodes) easy to process, and much smaller ones (say, less than 100 nodes) which are very hard.
- The design constraints in this particular design problem were very strong, thereby yielding a small number of solutions. Only four feasible solutions were found. Therefore, we think that it is necessary to identify the design parameters implying over-constraining and possible trade-offs within the architecture. For the time being, our method does not propose any solution for this type of analysis to support the designers. Therefore, possible future work could focus on the development of appropriate sensitivity analysis techniques related to the impact of parameter and decision variables on the overall architecture performance, as well as the impact of constraints on the set of possible solutions.

The proposed BN based method supports designers in the decision making process by providing discrete confidence level clusters. However, this approach cannot be deemed the most appropriate for system architecture evaluation and selection. As discussed previously, we believe that it is important to integrate designer preferences. Therefore, a more elaborate decision-making support mechanism is required that allows these preferences to be refined.

Finally, as we noted earlier, this design problem involves structural architecture. The relationship between performance and functional requirements is important to ensure that all necessary requirements have been considered. In order to propose a holistic approach towards architecture generation we believe that this part is essential and must be addressed in the future.

3.7 Conclusions

System architecture generation and evaluation remain challenging due to a large number of parameters that must be considered as well as their interdependencies. Moreover, due to innovation integration, several types of uncertainties (uncertainty concerning component characteristic, uncertainties related to component satisfaction of system functions and related system performances, uncertainties concerning component compatibilities) add complexity to the architecture design process.

In this paper, we propose a Bayesian network approach for system architecture generation and evaluation. In this approach, a Bayesian network model represents architecture design problem and designer's knowledge in order to integrate two types of uncertainties: 1) uncertainties concerning component characteristics and 2) uncertainties concerning component compatibility. In addition, we propose an architecture generation and evaluation algorithm enabling the clustering and filtering of different architectures in accordance to a defined overall confidence threshold. This approach is illustrated using the bicycle design problem. Moreover, we verified its usability on a large-scale system, an antenna cooling system design problem. The architecture was developed in collaboration with experts and was implemented in an industrial context.

Some of the advantages of the proposed approach included: reduced time for architecture generation and evaluation, easy performance consideration and integration, considerable knowledge capitalisation, and a better overall understanding of the design problem. However, some limitations have been observed and call for enhancements like integrating designer's preferences in architecture evaluation, enabling the scalability of the problem, i.e. introduce levels of abstraction, and identifying possible trade-offs within the architecture.

3.8 Acknowledgement

This research work has been financed by French government and Thales enterprise. Hereby, we would like to acknowledge Thales for their constant contribution, support and remarks concerning this project. Their feedback on field testing has been a real added value for this project.

3.9 References

Albarello, N. and Welcomme, J.-B. (2012), *"A formal design synthesis and optimisation method for systems architectures"*. Bordeaux, France.

Albers, A., Braun, A., Sadowski, E., Wynn, D., Wyatt, D. and Clarkson, J. (2011). "*System architecture modeling in a software tool based on a contact and channel approach (c&c-a)*", J Mech Design, **133**(10): 8.

Antonsson, E. and Cagan, J. (2001), "*Formal engineering design synthesis*". Cambridge University Press: UK.

Aughenbaugh, J. M. and Paredis, C. (2006), *"Why are intervals and imprecision important in engineering design?"*. Savannah, GA.

Barton, D. N., Saloranta, T., Moe, S. J., Eggestad, H. O. and Kuikka, S. (2008). "*Bayesian belief networks as a meta-modelling tool in integrated river basin management — pros and cons in evaluating nutrient abatement decisions under uncertainty in a norwegian river basin*", Ecological Economics, **66**(1): 91-104.

Bellut, S. (1990). "*La compétitivité par la maîtrise des coûts*", Conception à coût objectif et analyse de la valeur.

Bryant, C., Mcadams, D. A. and Stone, R. B. (2005), *"A computational technique for concept generation - detc2005-85323"*. Long Beach, USA.

Chakrabarti, A. (2002), "*Understanding, approaches and tools*". Springer: London, UK.

Chakrabarti, A. and Bligh, T. P. (1996). "*An approach to functional synthesis of mechanical design concepts: Theory, applications, and emerging research issues*", AI EDAM, **10**(04): 313-331.

Chandrasekaran, B. (1990). "*Design problem solving : A task analysis*", AI Magazine, **11**(4): 12.

Crawley, E., Weck De, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D. and Whitney, D. (2004), "*The influence of architecture in engineering systems*".

Dieter, E. (2000), "*Engineering design: A materials and processing approach*". McGraw-Hill: Singapore.

Gero, J. S. and Kannengiesser, U. (2004). "*The situated function-behaviour-structure framework*", Design Studies, **25**(4): 373-391.

Gupta, S. and Okudan, G. L. E. (2008). "*Computer-aided generation of modularised conceptual designs with assembly and variety considerations*", Journal of Engineering Design, **19**(6): 533-551.

Helms, B. and Shea, K. (2012). "*Computational synthesis of product architectures based on object-oriented graph grammars*", J Mech Design, **134**(2).

Hsu, W. and Liu, B. (2000). "*Conceptual design: Issues and challenges*", Computer Aided Design, **32**(14): 849-850.

Jensen, F. V. and Nielsen, T. D. (2007), "*Bayesian networks and decision graphs*". Springer: New-York.

Kerzhner, A. and Paredis, C. (2009), *"Using domain specific languages to capture design synthesis knowledge for model-based systems engineering - detc2009-87286"*. San Diego, California.

Kreimeyer, M. F. (2009). *"A structural measurement system for engineering design processes"*. Technical University of Munich.

Kurtoglu, T. and Campbell, M. I. (2009). "*Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping*", Journal of Engineering Design, **20**(1): 83-104.

Lindemann, U., Maurer, M. and Braun, T. (2008), "*Structural complexity management: An approach for the field of product design*". Springer-Verlag: Berlin

Matthews, P. C. (2011). "*Challenges to bayesian decision support using morphological matrices for design: Empirical evidence*", Research in Engineering Design, **22**(1): 14.

Moullec, M. L., Jankovic, M., Bouissou, M. and Bocquet, J. C. (2012). *"Product architectures generation under uncertainty: Comparison between two methods"*. In *Proceedings of the IDETC/CIE 2012*. Chicago, IL/USA, ASME.

Roger, J. (1999), *"Antennes - différents types"*.

Rosenstein, D. and Reich, Y. (2011), *"Hierarchical concept generation by sos"*. Copenhagen, Denmark.

Scaravetti, D. (2004). *"Formalisation préalable d'un problème de conceptio, pour l'aide à la décision en conception préliminaire"*. Thèse de doctorat, Ecole Nationale Supérieure d'Arts et Métiers.

Stone, R. B. and Wood, K. L. (2000). "*Development of a functional basis for design*", J Mech Design, **122**(4): 359-370.

Ulrich, K. T. and Eppinger, S. D. (1995), "*Product design and development*". Irwin McGraw-Hill: New York.

Whelton, M., Ballard, G. and Tommelein, I. D. (2002). "*A knowledge management framework for project definition*", ITcon, **7** 197-212.

Wyatt, D., Wynn, D., Jarrett, J. and Clarkson, P. (2012). "*Supporting product architecture design using computational design synthesis with network structure constraints*", Research in Engineering Design, **23**(1): 17-52.

Yannou, B. (2001). *"Préconception de produits"*. Habilitation à diriger les recherches, Ecole Centrale Paris.

Zablit, P. and Zimmer, L. (2001), *"Global aircraft predesign based on constraint propagation and interval analysis"*. Köln, Germany.

3.10 APPENDIX: Discretisation of continuous variables

As mentioned in Section 3, in order to use exact inference algorithms (which is the best option for truly discrete variables), it is necessary to discretise continuous variables. This has a number of consequences on the construction of the BN, and on the precision of the results. For example, let us suppose that we have two continuous variables $X \in [x1, x2]$ and $Y \in [y1, y2]$, and that we want to calculate some deterministic function $Z = f(X, Y)$. This kind of situation is often encountered in our design method when we have to compute the performances of a potential design. In a BN, this will be described with a node Z having the two parents X and Y. The function *f* is in most cases written as a formula, but this information must be transformed into a conditional probability table containing only zeros and ones, before calculations can be performed. The size of the table is the product of the number of states of the node with the numbers of states of each of its parent nodes. So if a node has many states, or many parents, then the table may be very large, and Netica may report that it doesn't have enough memory for the operation. One can alleviate the problem by using coarser discretisations. Finally, a trade off must be made between a fine discretisation that produces precise results, but with a high computing cost and a coarser one that speeds up the calculations, but makes them less precise. Anyway, whatever the chosen discretisation the user must be aware of the fact that the conditional probability table used for inference is the result of a random sampling process, and does not contain exact values.

Here is an example to explain this, with $X \in [2, 3]$, $Y \in [10, 20]$ and $Z = X + Y$. Figure 3-13 represents this case, with 10 discretisation intervals for *X, Y* and *Z*.

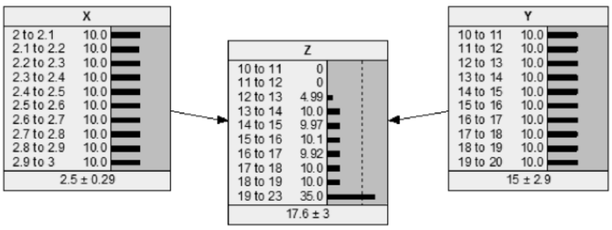


Figure 3-13: Discretisation of continuous variables: example *Z = X + Y*

Supposing that the probability distributions of *X* and *Y* are uniform, when for example $X \in [2.4, 2.5]$, and $Y \in [14, 15]$ Z may be either in [16, 17] or in [17, 18].

To determine, say $p = \Pr(Z \in$[16, 17] | $X \in [2.4, 2.5]$ and $Y \in [14, 15]$ Z), Netica uses a Monte Carlo simulation: it draws at random a sample of values for the couple (*X*, *Y*), supposing that *X* and *Y* are uniformly distributed in their respective discretisation intervals, and computes the value of *Z* for each element of the sample. Then p = the proportion of values of Z falling in the interval [16, 17]. Of course, the larger the size of the sample, the better the precision of the conditional probability table. The advantage of this method is that it is robust and works for any kind of relation between X, Y and Z. It can be deterministic like in the example above, but it can also be probabilistic: Z could be randomly distributed with a distribution whose parameters depend on X and Y.

Considering this process, one can see that there are some traps in the use of continuous variables. For example, there could be intervals with a null estimated probability, like the two first intervals of Z in Figure 3-13. In this example, the reason is quite obvious and it is possible to delete those two intervals, but it may not always be the case. So, in order to get good results, it is necessary to try various discretisations, the objective being to get distributions as "uniform" as possible, in the absence of any observation input in the network. In the example of Figure 2, the spike for the last interval of Z suggests that this interval should be divided into smaller ones.

4

Paper #2. Proposition of combined approach for architecture generation integrating component placement optimisation

Marie-Lise Moullec, Marija Jankovic, Marc Bouissou, Jean-Claude Bocquet, Olivier Maas

This paper has been published in Proceedings of IDETC 2013, under the following reference:

Moullec M.-L., Jankovic, M., Bouissou, M., Bocquet, J.-C., Maas, O., 2013, "Proposition of combined approach for architecture generation integrating component placement optimisation", Proc. IDETC2013, Portland, OR.

Abstract. *Architecture generation and evaluation are critical points in complex systems design. System architecting starts with the exploration of a set of potential solutions that is progressively focused towards the most promising ones. In theory, these solutions are identified through their potential ability to reach system requirements. However, in early design stages, data supporting design choices are fuzzy and uncertain, making difficult the evaluation a priori of the future system architecture performance. In this paper, we propose a method using Bayesian Networks (BN) and Constraint Satisfaction Problem (CSP) to first generate potential system architectures, and then select the best ones regarding system requirements. The association of these two approaches allows enhancing architecture evaluation through integration of component placement optimisation. This approach is demonstrated and implemented in radar antenna architecture generation. In the end, we also discuss some of the limits of the proposed approach as well as future research directions.*

The CSP approach presented in this paper is presented more deeply in Appendix A.

4.1 Introduction

System architecture can be viewed as "an abstract description of the entities of a system and the relationships between those entities" (Crawley, Weck De, et al. 2004). These relationships relate to functions to components mapping and interfaces between components (Ulrich and Eppinger 1995). In early design stages, reasoning about the system architecture supports the designer to create a system with desired behaviours (Crawley, Weck De, et al. 2004) in order to reach system requirements and objectives. When system architecture is fully decided, i.e. when physical components arrangement and main system dimensions are determined (Scaravetti 2004), detailed system design begins. In most of the cases the design decisions made earlier can no longer be changed. This explains the major impact of the system architecture on overall life cycle costs (Whelton, Ballard, et al. 2002b).

Therefore, supporting rigorous design space exploration is essential. This exploration consists of generating several system architectures and exploring their strengths and weaknesses. Addressing these issues has been a research topic for more than 30 years (Antonsson and Cagan 2001) but system synthesis and evaluation remains challenging (Albers, Braun, et al. 2011). However, the major issues met in industrial context are lack of time for system architecture generation, and imprecision and fuzziness in data supporting system architecture evaluation (Hsu and Liu 2000). In order to address these issues, we propose an integrated approach based upon Bayesian Networks (BN) and Constraint Satisfaction Problem (CSP), in order to enhance the evaluation of resulting architectures by integrating component placement optimisation. This approach is applied to the design of a radar active antenna, which is an industrial deployment within Thales Air Systems.

Henceforth, in Section 4.2 we propose to address the relevant state of the art. In Section 4.3, we explain the global methodology of the method. This method is illustrated through the design problem of an active antenna in Section 4.4. Section 4.5 presents discussion of current limits and future developments. The last section summarises our contribution and gives some perspectives.

4.2 State Of The Art

4.2.1 Existing Methods

Two important steps in system architecting are: (1) functions to components mapping while ensuring technical interfaces compatibility, and (2) components parameterization in order to make possible evaluation of potential architecture ability to reach system requirements. Most of existing methods focus on one of the two points, but rarely on both.

Only a few methods are dedicated to supporting system architecture synthesis. These methods vary according to input data used to support system synthesis and proposed system generation techniques. Concerning the data used for system architecture synthesis, existing methods can be classified as follows:

- Function-oriented synthesis methods (Bryant, Mcadams, et al. 2005; Kurtoglu and Campbell 2009; Albarello and Welcomme 2012; Helms and Shea 2012) are focused on defining a functional architecture from which structural architecture can be deduced.
- Component-oriented synthesis methods (Rosenstein and Reich 2011; Wyatt, Wynn, et al. 2012) start with an initial structural architecture and make it vary according design rules based on constraints (Wyatt, Wynn, et al. 2012) or on the qualitative evaluation of performances (Rosenstein and Reich 2011).
- Design problem-oriented synthesis methods (Chenouard 2007; Moullec, Bouissou et al. 2013) propose to represent the design problem using constraints and performance calculations. Afterwards, a solver « reads » this problem and gives all the architecture solutions.

In order to transform these inputs data into structural architectures, these methods generally rely on system generation techniques such as graph grammars rules (Helms and Shea 2012), Design Structure Matrix (DSM) (Bryant, Mcadams, et al. 2005), genetic algorithms (Rosenstein and Reich 2011; Albarello and Welcomme 2012), Bayesian networks (Matthews 2011; Moullec, Bouissou, et al. 2013) or search techniques (Chenouard 2007; Yvars 2008; Wyatt, Wynn, et al. 2012).

Methods that are matrix-based (Bryant, Mcadams, et al. 2005) and using graph grammar rules (Kurtoglu and Campbell 2009; Helms and Shea 2012) are particularly adapted when a new product must be developed: they explore design space by using design repositories and generate consistent structural architectures. Nevertheless, the main weakness of these techniques is that they require the elicitation of each design rules that might be cumbersome in the case of complex system design. Although having a good representation of the system, most of the approaches lack detailed system quantification making impossible the evaluation and comparison of the architectures generated. On the contrary, genetic algorithms use qualitative (Rosenstein and Reich 2011) or quantitative (Albarello and Welcomme 2012) evaluation in order to transform an initial population of architectures into new architectures with better performances. However, these methods slowly converge and necessitate several iterations in order to guarantee a good exploration of the design space. Likewise, the methods based on an initial architecture do not allow a complete exploration of the design space: this implies that starting from two different architectures may give different solutions. The use of constraints in search techniques concern either component compatibilities (Wyatt, Wynn, et al. 2012) or component parameterization (Chenouard 2007). In the latter case, the overall architecture is already fixed. Nevertheless, in both methods, the input must be very precise which is rarely the case in preliminary design. This explains that techniques such as CSP have proved their efficiency mainly in more advanced design stages like layout design. In conceptual design, we believe that using Bayesian Networks is a good alternative because it allows to integrate design uncertainties and designer knowledge, while ensuring that system requirements are fulfilled (Moullec, Bouissou, et al. 2013). Finally, the two last techniques likely appear to be complementary.

4.2.2 Challenges

Previously stated work shows different approaches to address architecture generation. However, one of the major problems concerns uncertainty integration and lack of data in early design. Moreover, the issue of architecture evaluation has only recently started to be addressed. Although there are few works on this subject, we believe that this issue is crucial in order to support the design team in architecture comparison and identification of most promising one. We think that architecture evaluation should include all available information in preliminary design, i.e. designer knowledge and existing solutions. Moreover, as already discussed, architecture evaluation must also deal with the lack and fuzziness of data, and thus with the degree of uncertainty characterising the information (Aughenbaugh and Paredis 2006), in particular when considering new technologies. We argue that uncertainty must be considered as a selection criterion on the same basis as other architecture performances.

In order to address the issue of evaluation in view to the component placement within a given architecture, we propose to enrich already discussed approach using Bayesian Networks (Moullec, Bouissou, et al. 2013) with CSP. The aim is to continue to integrate component and interface uncertainties in order to evaluate architecture using CSP for integrating component placement. Therefore, in Section 3 we detail the proposed integrated approach: enrichment of BN method and definition of CSP for component placement. This approach has been tested and deployed in industry which is discussed in Section 4.4. In Section 4.5 we discuss the advantages and current limitations of this approach. In Section 4.6, we conclude with a synthesis and future work.

4.3 System Architecting

4.3.1 Architecture design problem definition

Scaravetti in his study of the most relevant design process definitions (Scaravetti 2004) shows that although order of the steps can vary, a general design process can be identified. System design begins with system functions specification which leads to the following steps:

1. From system functions specification, system requirements are expressed in terms of objectives and/or constraints.
2. A functional architecture is defined and principle solutions are elicitated.
3. Principle solutions are combined within the concepts.
4. After evaluating and comparing these concepts, the most promising one is represented as structural system architecture regarding global system performances. The main system components are chosen, as well as their physical arrangement and their sizing characteristics.
5. Determination of main system dimensions is the last step of system architecture definition.

For a good solution achievement, this process is reiterated many times.

4.3.2 A Method for System Architecting

Based on this definition, we propose a method that encompasses Step 3 to Step 5. We propose an integrated approach based upon both Bayesian Networks and CSP (Figure 4-1). The main objective of this approach is to integrate uncertainties concerning the system interfaces and component characteristics and to optimise component placement in view to system performances. A Bayesian Network is used for generating feasible architectures. Then, CSP is used in order to check the consistency of BN solutions regarding their ability to reach layout-based performance requirements.

In the beginning, the designer specifies a repertoire of key components as well as their relations. The choice of these key components is motivated by their consistency regarding system functions (the functional architecture is assumed to be already determined). All component characteristics involved in component compatibilities and performance calculation are identified and described through discrete or continuous variables. Afterwards, designers establish the complete set of constraints that ensure the system requirements fulfilment. Our approach requires that performances and constraints are all classified into two types depending on whether they involve component placement considerations or not.

At this point, the designer has elicitated all known information to represent the design problem. A Bayesian Network is used to model this data, as explained in previous articles (Moullec, Bouissou, et al. 2013). A Bayesian Network is a probabilistic graphical model that represents a set of variables and their joint probability distribution. The advantage of this approach is uncertainty integration (Moullec, Bouissou, et al. 2013). In order to design the network, the designer has to identify decision nodes. Appropriate characteristic nodes are specified according to system performances. Afterwards, constraint nodes are identified and defined between decision and characteristic nodes. In the end, in view to the problem, a global confidence node is defined with regard to overall design project risk. An algorithm uses this network to explore the design space and generate architecture solutions. We will address these solutions as "BN solutions" in the sequel.

In most of the cases, system architecting may have ended at this step. However, overall system performances depend not only upon components but also their placement within a given architecture. In view to this issue, we propose to use CSP in order to further investigate the ability to reach layout-based performances requirements. The choice of using CSP is motivated by the fact that we experimentally found that layout problems are often too complex to be resolved using a Bayesian Network. Defining a problem using CSP formalisation consists in defining:

- a set of decision variables $X = \{x_1, x_2,..., x_n\}$;
- the decision variables' domains $D = \{d_1, d_2,..., d_n\}$ such that x_i takes its values in d_i;
- a finite set of constraints $C = \{c_1, c_2,..., c_k\}$ that must be satisfied by decision variables.

Once defined, a CSP is solved using a specific solver such as Choco (2013) or Ilog (2006). The CSP formalism is generic and allows solving problems different in nature. In our method, a possible architecture resulting from BN approach is an input for the CSP step. The objective is that if at least one solution exists, the proposed CSP model suggests several possible layouts or an optimised layout at the choice of designer.

At the end of this process, the architecture solution is completely defined. If there is a need to test another BN solution, it is necessary to change only the inputs corresponding to the size and number of components in CSP. CSP does not require being adapted for each BN solution.

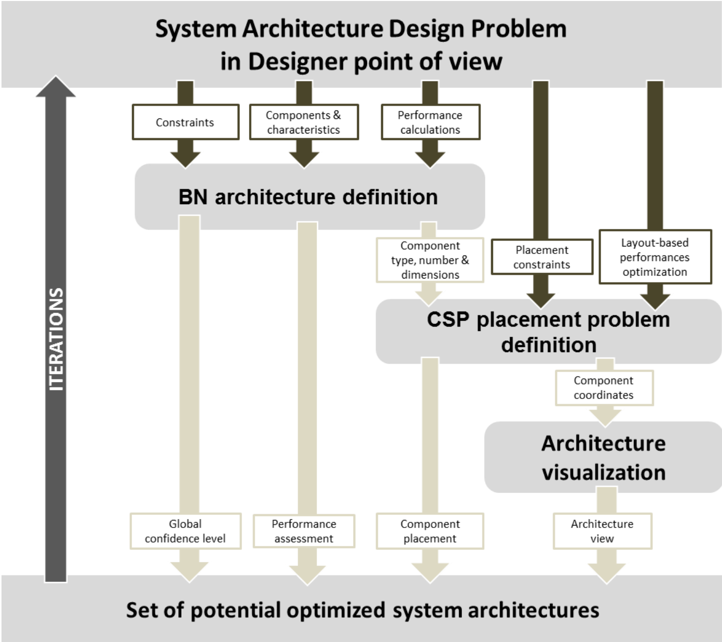


Figure 4-1. Global process of the method

We propose the following process for this integrated approach (Figure 4-1):

- The designer defines the BN and CSP models.
- The BN generation algorithm generates feasible system architectures, i.e. "BN solutions". These solutions take into account system requirements and constraints. They are characterised by a set of components with defined dimensions.
- The designer chooses a solution among these BN solutions, and enters it in the CSP model.
- The CSP model proposes several possible layouts that (1) satisfy placement constraints between components and (2) optimise layout-based performances.
- These layouts are visualised into a CAD viewer.

Finally, the obtained architecture solutions are completely defined by a set of specific components and their relative position. A view of the expected architecture allows designers to quickly understand the solution. Moreover, performances assessments as well as a global architecture confidence level of a given solution are provided.

In the following sections, we describe in details the content definition of both models, as well as their implementation.

4.3.3 Determining potential architectures

The Bayesian Network model has been developed and discussed previously (Moullec, Bouissou, et al. 2013). The templates are given in Appendix A.

4.3.3.1 BN model definition

System functions and potential working principles, represented as key components, are inputs of the BN model.

It is also necessary to identify system performances that are considered as critical on the system level. Performances that do not require any information on component placement are treated within the Bayesian Network. Otherwise, they will be taken into account in the CSP.

Formally, a Bayesian Network is a directed acyclic graph whose nodes represent variables, and edges represent conditional dependencies. A BN is defined by a set of nodes and a set of relationships. For each node, different states are defined representing possible values for one node. Edges represent relationships between nodes defined by conditional probabilities. The proposed BN model defines a design problem through decision variable nodes, characteristic nodes, constraint nodes, performance nodes and a global confidence node (Moullec, Bouissou, et al. 2013). We propose to enrich this model by detailing the elements that need to be in the Bayesian Network so that solutions can be processed further in CSP.

For each system function f, the following elements need to be defined:

- two decision variable nodes (possibly linked together):
  - a node representing f and its possible components. The probability table of this node is uniform.
  - a node that represents the possible numbers of components for f.
- a minimum of three characteristic nodes that represent the dimensions (the width w, the height h and the depth d) of the key component used for the function.

If two functions are possibly accomplished by the same key component, this component appears in both function nodes. However, one must add constraint nodes ensuring that dimensions and component number are the same.

Afterwards, the designer creates performance nodes. In most cases, in order to define these nodes, probability tables are calculated using equations. Finally, constraint nodes are added. The constraints declared in BN model can be classified in three types:

- A compatibility constraint ensures that key components of two different types can be used in a same architecture solution.

  <u>*Example:*</u> *On a bike, a luggage rack cannot be integrated if disc brakes are used.*

- A requirement constraint related to a performance or a characteristic guarantees that value thresholds defined by system requirements are respected.

  Example: The maximal accepted mass of the system is 15kg.

- A sizing constraint indicates a dependency between the dimensions/number of two components, or between a performance and the dimension/number of a component.

  Example: If a 30-speed derailleur is chosen, then there are three chain rings and ten sprockets.

Once all nodes are created and their probability tables are defined, the corresponding BN is ready to be processed by the generation algorithm.

4.3.3.2 BN Implementation and Results

Architectures generation is supported by the algorithm described in (Moullec, Bouissou, et al. 2013). This algorithm explores the entire design space and proposes only the solutions that satisfy given constraints. With the BN model definition described in the previous section, an architecture solution is given as follows:

- Each function f is associated to a type and number of component i.
- Each component i is defined by its main dimensions: width w_i, height h_i and depth d_i.

These characteristics are necessary inputs so that CSP model determines component placement.

However, the previous step could have provided a high number of potential architectures solutions, which requires making a choice between these solutions. This selection is expert-based, or can be considered using MCDA techniques. We believe that this part represents an essential support for the design team and therefore represents our main research objective in further developments.

4.3.4 Managing Component Placement

In order to enrich BN solutions and take into account component placement, we propose a CSP model whose input is one of the BN possible architectures. The given BN solution is then represented in CSP formalism.

4.3.4.1 CSP placement problem definition

In order to refine architecture generation, we propose to integrate placement constraints between components of the different system functions. In accordance with CSP formalism (see Scetion 4.3.2), placement problem has been defined as follows:

- **Decision variables** are at least coordinates x_i, y_i and z_i, as well as orientation o_i of each component of the BN solution. These decision variables are automatically created when the designer creates a function f. Additional decision variables, related to the performances that must be evaluated in the CSP, can be added for each function.
- **Decision variable domains** depend on which decision variable is concerned:

- x_i, y_i and z_i domains are determined by a maximal space that designers are ready to accept. This space is defined with maximum width w_{max}, maximum height h_{max} and depth d_{max} such as:

  $x_i \in [0\ ;\ w_{max}]$

  $y_i \in [0\ ;\ h_{max}]$

  $z_i \in [0\ ;\ d_{max}]$

- o_i has 6 possible orientations, each identified by a number. Thus:

  $o_i \in \{0;1;2;3;4;5\}$

- **Constraints.** Several types of constraints are used as follows :
  - A position constraint (Figure 4-2) on a component indicates that one or several coordinates of the component equals a specific value.

    Example: A car window is situated at 1m in height.

  - An adjacency constraint (Figure 4-2) indicates that a component *i* must be in contact with a component *j*. The possible relative positions (on the right side, on the left side, on the top, on the bottom, in front of, behind) are specified.

    Example: A windscreen wiper is adjacent and in front of the wind shield.

  - Orientation constraint indicates that component *i* must be in a specific orientation (among the six ones possible)

    Example: A side window car is vertical.

  - Distance constraint (Figure 4-2) indicates that distance between component *i* and component *j* must be at a specified distance. Specific thresholds, in place of a fixed value, can also be defined.

    Example: Two windows are distant of at least 1.5m and not more than 2.5m.

  - Performance constraint indicates that a threshold of performance must be reached by the assembly to be considered as a solution. Such constraints concern performances impacted by components placement.

    Example: Total length of the assembly is at most 3m.

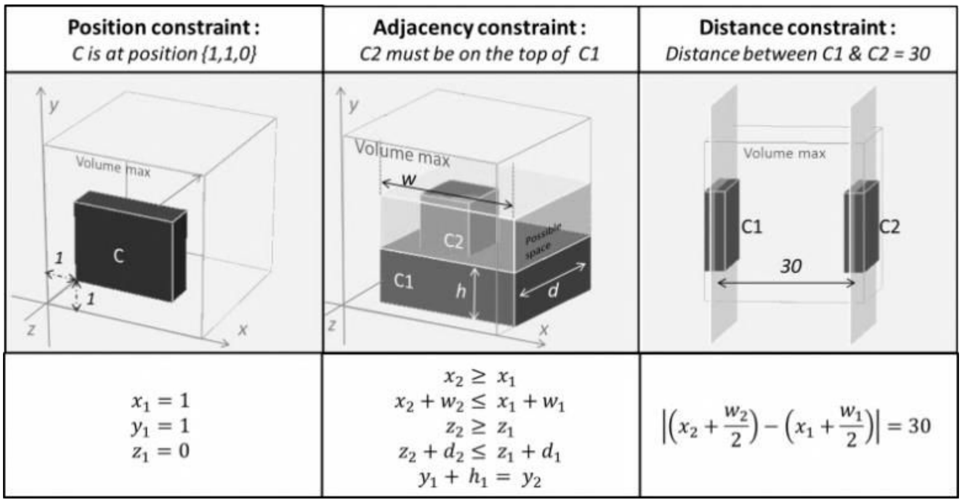


Figure 4-2. Representation of CSP constraints

At this point, the architecture placement problem is defined using the CSP. However, any placement problem requires implicit additional constraints in order to be solved. These constraints are automatically implemented (without any intervention from the designer). The most important one is the non-overlapping constraint which ensures that two different components do not occupy the same space. Other constraints, like symmetry constraint, can also be added in view to resolution time reduction. More information about interest and implementation of these constraints can be found in (Medjoub 1996).

4.3.4.2 CSP Implementation and Results

Once the CSP model is defined and BN solutions are generated, designer selects one of them to be optimised using CSP model. The designer introduces inputs by defining system functions. For each function, there is a need to specify number and dimensions of components. The dimensions of the maximal space for overall system can also be defined. Finally, designer indicates the performance to optimise and the number of desired layouts.

CSP is modelled using a specific CSP solver such as Ilog (2006) or Choco (2013). For this case, we use Choco. Choco solver uses by default a branch & bound algorithm that defines the order of variable assignment according to a ratio that is based on variable domains and the number and strength of constraints to be assigned to each variable. In order to gain time, we propose to add settings so that solver treats constraints in the descending order of weight.

Finding an optimised solution may be time-consuming. However, this allows obtaining an order of magnitude of the best performances that one can expect for a given solution. These performances are defined by creating a new "decision variable" to be maximised or minimised. This decision variable can correspond to a weighted objective function, in which several performances can be optimised. CSP solver explores the entire design space until it determines with certainty that there is no other solution with better performances. In cases in which designer desires x layouts optimising the solution, the solver proposes the x solutions yielding the best performances. However, there may be many solutions with the same performance and the algorithm lists the solutions in the order in which CSP solver finds them. In consequence, resulting solutions may be very similar. As the objective is to explore as much as possible the design space, finding various solutions is essential but may be time consuming in terms of evaluations and exploration. We propose to set the solver so that values variables are randomly selected. Moreover, the order of variables to be implemented is also randomly treated.

This CSP model proposes for each component constituting the explored architecture a position and an orientation. Currently, component representation considers mostly the volume represented as a "box". In order to view in detail different component shapes, there is a need for further development. Moreover, CSP creation is a complex task for designer who is not expert in CSP programming. This requires a specific user interface that is currently under development.

4.3.4.3 Visualising the Architecture Solution

Results for a given architecture, represented as couples of coordinates and orientation, are not easily readable. We propose to display a solution in the form of an assembly of basic boxes in an open source CAD viewer: OpenSCAD (2010). This visualisation feature is fully automatic. Figure 4-3 illustrates the type of system architecture views that are obtained at the end of the method.

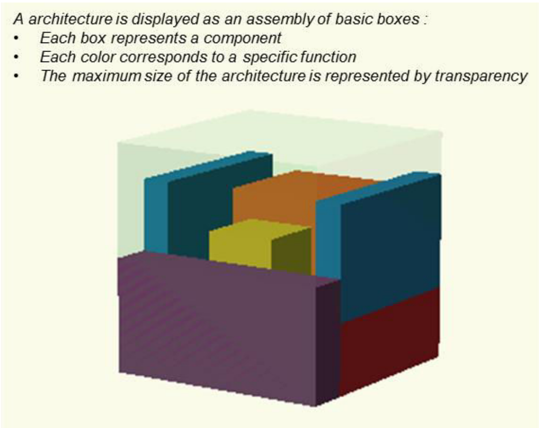


Figure 4-34-3. Visualisation of a final architecture solution

This visualisation allows a better understanding of proposed solutions. Particularly, this speeds up validation and comparison of potential architecture solutions.

4.4 Industrial Case Study

This integrated approach is tested and validated in the case of design of a radar active antenna with regard to the thermal performances. In order to test the approach, this problem has been previously solved without such computational support by a design team in Thales. They found a single feasible solution that they consider « not optimal ». Afterwards, the same problem has been modelled and solved using our approach.

4.4.1 Radar Active Antenna design

A radar antenna function consists of transmitting and receiving electromagnetic signals in order to detect the presence of objects in a given area. These functions are accomplished by electronic devices that are, in this case, grouped in modules called Building Blocks (BBs). Moreover, in order to be ensuring a sufficient range, the transmitted signals must be amplified before being radiated; and the received signals must be amplified before being sent to the calculation system. An active antenna has the particularity of comprising the elements accomplishing signal amplification very close to the radiating elements (Figure 4-4). This particularity makes active antenna design complex and costly because amplifiers, which heat up the most, are located in the antenna itself and require an efficient cooling system. Moreover, another major requirement is that the global volume of the antenna needs to be reduced as much as possible. In conceptual design, the objective is to explore the design space of radar antenna and find the most promising architectures. These architectures are combinations of several design variables that concern:

- Cooling components and fans, which constitute the cooling system of the antenna;
- BB that accomplish the main functions of the antenna.

This design problem has been partially addressed in a previous work described in (Moullec, Bouissou et al. 2012): a BN has been used to generate all possible architectures of the cooling system with given BBs. Four antenna solutions have been identified. However, global volume of the antenna, as well as thermal performance, mainly depends on BB.

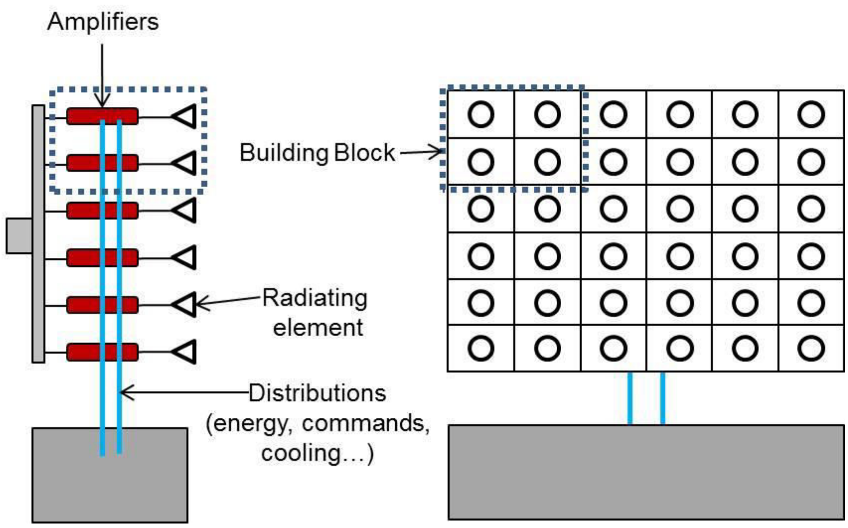


Figure 4-4. Description of a radar active antenna (adapted from (Roger 1999))

4.4.2 Understanding and formalising the problem

Due to confidentiality issues, the modelling that we deal with, in this paper, cannot be presented in detail. However, the approach that we use to model this problem is presented.

Two experts have participated in BB modelling. The first expert is specialised in RF while the second one develops mechanical and thermal solutions. The objective was to determine the modelling hypothesis, the decision variables that are interesting to be modelled as well as the performances to take into account. Four successive interviews were needed in order to define the problem.

From the interviews, 5 main functions of BB have been identified. Functions are RF signals radiation and amplification, RF control signals, power supply and cooling. Concerning the performances to be modelled, they cover RF performances, thermal performances and volume of BB. RF and thermal performances have been modelled in the BN. These performances are necessary to determine the size of components. Based on these hypotheses, experts define the equations for performances, as well as the constraints to be satisfied. The main types of constraints entered in the BN networks are compatibility constraints and sizing constraints. Only a single requirement constraint has been added. Author has supported BN building, because the user interface is not yet sufficiently developed. However, the BN has been approved by experts. Finally, the BN contains 10 decision variable nodes, 35 characteristic nodes, 50 intermediate calculations and 35 constraints. The architecture generation algorithm using this input data yielded 180 possible solutions. Among these 180 solutions, experts chose 3 very different solutions (based on different working principles) that they wanted to investigate with the CSP and explore in detail with regard to placement optimisation. These solutions have also been chosen considering more "qualitative" capabilities such as maintenance or innovation level.

These 3 solutions have been tested in the CSP using a developed user interface. However, constraints and performances to be integrated in the CSP have been beforehand transcribed in Choco. To do this, all potential key components must be considered. The constraints that have been entered are mainly orientation constraints and adjacency constraints. Performances optimised in this problem are the total depth of BB, as well as the RF losses. The maximal number of components that has been tested for a solution is 20 components.

4.4.3 Results

4.4.3.1 BB solutions

Using the BN, 180 solutions are generated in 25 seconds. Among them, experts choose 3 solutions that they input in the CSP. For each solution, one layout is required. Table 1 summarises CSP processing time of component placement optimisation.

Processing time differs according to the BN solution that is treated. For instance, optimising the component placement for BB_1 takes more time than for BB_3. This can be explained by the number of components as well as their type: BB_3 is comprised of 11 components which are subject to stronger placement constraints. Moreover, processing time increases with the number of components, because of the combinatorial explosion of variable number and placement possibilities.

Figure 4-5 shows the resulting layouts of two solutions. The BB solution 1, BB_1 (Figure 4-5), corresponds to the existing BB (BB_0) that designers used in their previous study of the antenna. This should make possible the comparison between the estimated performances of BB_1 and the real performances of BB_0. As component characteristics are expressed in terms of intervals, the performances of BN solutions, like BB_1, are given as value ranges. On the other hand, the performances measured on the existing BB (BB_0) correspond to single point values. Thus, the only way to verify the adequacy of BN performance estimations lies in the investigation that performances of BB_0 can be found within the value ranges estimated for BB_1. This verification confirmed that this is the case.

Table 4–1. CSP performances

Solution	Number of components	Processing time
Solution 1: BB_1	20	146 s
Solution 2: BB_2	6	0.275 s
Solution 3: BB_3	11	0.445 s

In addition, dimensions of both solutions are very similar: width and height are constrained by system requirements and thus identical. Depth is one of the performances to be optimised, and is quite similar (observed error rate: 5%). This means that the component sizing, defined by experts, is adequate, and that CSP solving has been efficient.

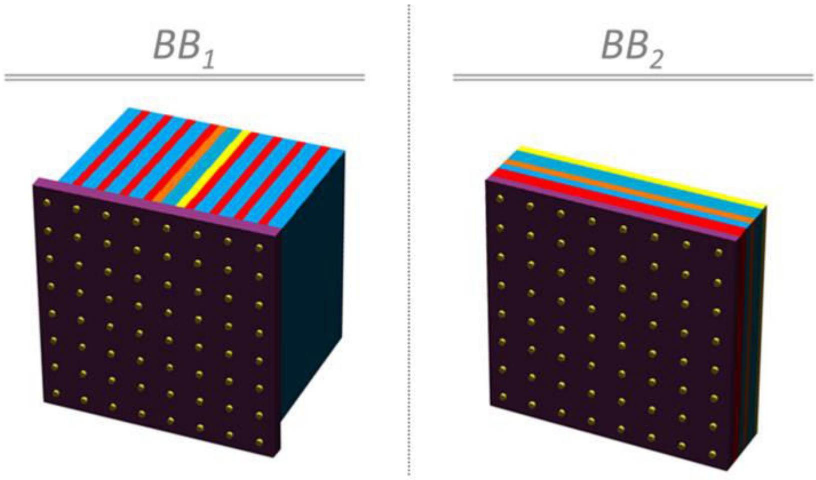


Figure 4-5. Layouts of 2 solutions of building blocks

4.4.3.2 Antenna solutions

The resulting three solutions have been entered as decision variables in the BN in charge of generating new architectures of antenna. In a previous modelling using only the existing BB (BB_0), only 4 solutions have been found. After the introduction of these 3 new solutions of BB, 14 antenna architecture solutions have been proposed:

- 4 antenna architectures using BB_0;
- 10 antenna architectures using BB_1;
- 0 antenna architectures using BB_2;
- 0 antenna architectures using BB_3.

Two interesting points can be discussed in view to results. First of all, as BB_1 is supposed to be equivalent to BB_0, the generated antenna architecture solutions should have been the same. It seems that this is not the case. This is due to the fact that performances are estimated with value ranges: the solution space becomes necessarily larger than the one using point values. Four solutions using BB_0 correspond to four proposed solutions using BB_1, which suggests that calculation models are good. This also shows that, with characteristics that differ from those chosen for BB_0, 10 other antenna solutions may be possible. However, this result must be considered carefully, because the high number of proposed solutions for the use of BB_0 also may be due to the estimation of thermal performances that could be not precise enough.

No solutions for BB_2 or BB_3 antenna architectures have been proposed. The analysis of the BN antenna solutions shows that the thermal performances of BB_2 (Figure 4-5) and BB_3 are not adapted for their integration in the antenna. In the usual expert based design process these types of BB were never studied because the solutions were considered as "unfeasible". This unfeasibility is now characterised by a maximum threshold for thermal performances.

There are two possible ways of continuing the exploration of antenna design space:

1. The experts can add a new constraint on thermal performances in the Bayesian network of BBs, and this will generate new BB solutions. Then, these solutions can be optimised by the CSP and integrated in the antenna BN.
2. The experts can still consider all BB solutions and look for cooling systems able to support such thermal performances. This possibility is motivated by the fact that it can reduce the global depth of the antenna by 66%.

In conclusion, what this case study suggests is that evaluation of system architecture performances provides essential information to steer the design space exploration during preliminary design and to manage trade-offs for systems requirements definition. The proposed approach allows experts to both explore specified design space while evaluating a wide scope of system architecture performances. Although design process still must be reiterated several times to obtain satisfying solutions, the approach makes preliminary design faster and more complete than using no methods. We believe that this point represents strength of the proposed approach because design iterations are inescapable, particularly in the case of complex system architecture generation. Tools that reduce time and costs development due to reiteration are therefore welcome, if not required.

4.5 Discussion

Sections 3 and 4 illustrate the proposed integrated method as a support for system architecture design through exploring the solutions space exhaustively, allowing integration of uncertainties, as well as evaluating and optimising solution performances. However, it also highlights several issues and limitations of the method.

The first issue using this method is gathering data and formalising them. This introduces necessary assumptions and simplifications that may be difficult to obtain, in particular when it comes to calculations usually performed in detailed design. However, feedback from the engineers indicates that building such models makes the design problem more understandable and better understood by designers of different domains, which improves collaborative design. Moreover, formalising the design problem in terms of constraints and objectives, rather than in terms of an overall solution, allows the designers to be more « open » to innovative architecture concepts. Integration of placement constraints particularly addresses a specific need expressed by different designers, and that we were unable to address using only BN model. The introduction of the CSP model completes the BN approach and is considered by designers as a major improvement. Nevertheless, the initial formalisation of the design problem may seem burdensome and requires more effort than in preliminary design. Design problem specification often requires several iterations in order to identify all constraints; and to model them in an appropriate way. However, once both models are defined, solutions are quickly obtained and visualised. The fact that the entire defined design space has been explored and that performances have already been estimated, and even optimised, is a benefit compared to classical methods.

A necessary condition for acceptation of this method is providing a user-friendly interface for both models: this integrated approach aims at supporting designers and no other competency should be required. For the time being, designers have been supported by authors in building BN and CSP models. Future development consists in providing an interface allowing an autonomous use of the method. Moreover, some steps have been made in a "qualitative" manner. This concerns in particular the selection of BN architectures to be processed in the CSP that is ad-hoc expert based. Given the possible high number of solutions, we argue that this selection requires a specific support that is to be developed.

4.6 Conclusion

System architecture design is an essential part of design process, but few methods allow both generating and evaluating system architectures. This paper proposes that a Bayesian Network and a Constraint Satisfaction Problem be used in a complementary way to represent a design problem and to propose new system architecture solutions. The approach has the advantage of evaluating architectures regarding system requirements and objectives, placement constraints and overall architecture uncertainty level. The proposed approach has been tested on an industrial case. However, this work also highlighted some issues, such as the difficulty of the design problem specification. Further work will consider user interface development, and architectures selection method. Others industrial applications are expected.

4.7 Acknowledgments

This research is supported and done in collaboration with Thales group. Hereby we thank them for their constant support and insightful feedback on limitations and needs for further development.

4.8 References

Albarello, N. and Welcomme, J.-B. (2012), *"A formal design synthesis and optimisation method for systems architectures"*. Bordeaux, France.

Albers, A., Braun, A., Sadowski, E., Wynn, D., Wyatt, D. and Clarkson, J. (2011). "*System architecture modeling in a software tool based on a contact and channel approach (c&c-a)*", J Mech Design, **133**(10): 8.

Antonsson, E. and Cagan, J. (2001), "*Formal engineering design synthesis*". Cambridge University Press: UK.

Aughenbaugh, J. M. and Paredis, C. (2006), *"Why are intervals and imprecision important in engineering design?"*. Savannah, GA.

Bryant, C., Mcadams, D. A. and Stone, R. B. (2005), *"A computational technique for concept generation - detc2005-85323"*. Long Beach, USA.

Chenouard, R., (2007), *"Resolution par satisfaction de contraintes appliquee à l'aide à la décision en conception architecturale"*, Bordeaux.

Choco 2. (2013). www.emn.fr/z-info/choco-solver/

Crawley, E., Weck De, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D. and Whitney, D. (2004), "*The influence of architecture in engineering systems*".

Helms, B. and Shea, K. (2012). "*Computational synthesis of product architectures based on object-oriented graph grammars*", J Mech Design, **134**(2).

Hsu, W. and Liu, B. (2000). "*Conceptual design: Issues and challenges*", Computer Aided Design, **32**(14): 849-850.

Ilog (2006). http://www-01.ibm.com/software/commerce/optimization/cplex-cp-optimizer/

Kurtoglu, T. and Campbell, M. I. (2009). "*Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping*", Journal of Engineering Design, **20**(1): 83-104.

Matthews, P. C. (2011). "*Challenges to bayesian decision support using morphological matrices for design: Empirical evidence*", Research in Engineering Design, **22**(1): 14.

Medjoub, B. (1996). *"Méthode de conception fonctionnelle en architecture : Une approche basée sur les contraintes: Archiplan"*. Thèse de doctorat, Ecole Centrale Paris.

Moullec, M. L., Bouissou, M., Jankovic, M. and Bocquet, J. C. (2012), *"Use of a bayesian nets-based method to automatically generate complex product architectures: Application on the cooling system of a radar active antenna"*. Paris, France: Springer-Verlag.

Moullec, M. L., Bouissou, M., Jankovic, M., Bocquet, J. C., Requillard, F., Maas, O. and Forgeot, O. (2013). "*Towards system architecture generation and performances assessment under uncertainty using bayesian networks* ", Journal of Mechanical Design, **135**(4): 13.

Openscad. (2010). http://www.openscad.org/

Rosenstein, D. and Reich, Y. (2011), *"Hierarchical concept generation by sos"*. Copenhagen, Denmark.

Scaravetti, D. (2004). *"Formalisation préalable d'un problème de conceptio, pour l'aide à la décision en conception préliminaire"*. Thèse de doctorat, Ecole Nationale Supérieure d'Arts et Métiers.

Ulrich, K. T. and Eppinger, S. D. (1995), "*Product design and development*". Irwin McGraw-Hill: New York.

Whelton, M., Ballard, G. and Tommelein, I. D. (2002). "*A knowledge management framework for project definition*", ITcon, **7** 197-212.

Wyatt, D., Wynn, D., Jarrett, J. and Clarkson, P. (2012). "*Supporting product architecture design using computational design synthesis with network structure constraints*", Research in Engineering Design, **23**(1): 17-52.

Yvars, P.-A. (2008). "*Using constraint satisfaction for designing mechanical systems*", International Journal of Interactive Design and Manufacturing, **2**(3): 161-167.

4.9 Annex A: Bayesian Network Templates

The method based on Bayesian Networks (for complex architecture generation and evaluation) provides some templates in order to model a design problem in an appropriate way. Below, Figure 4-6 summarises the main templates used in this method:

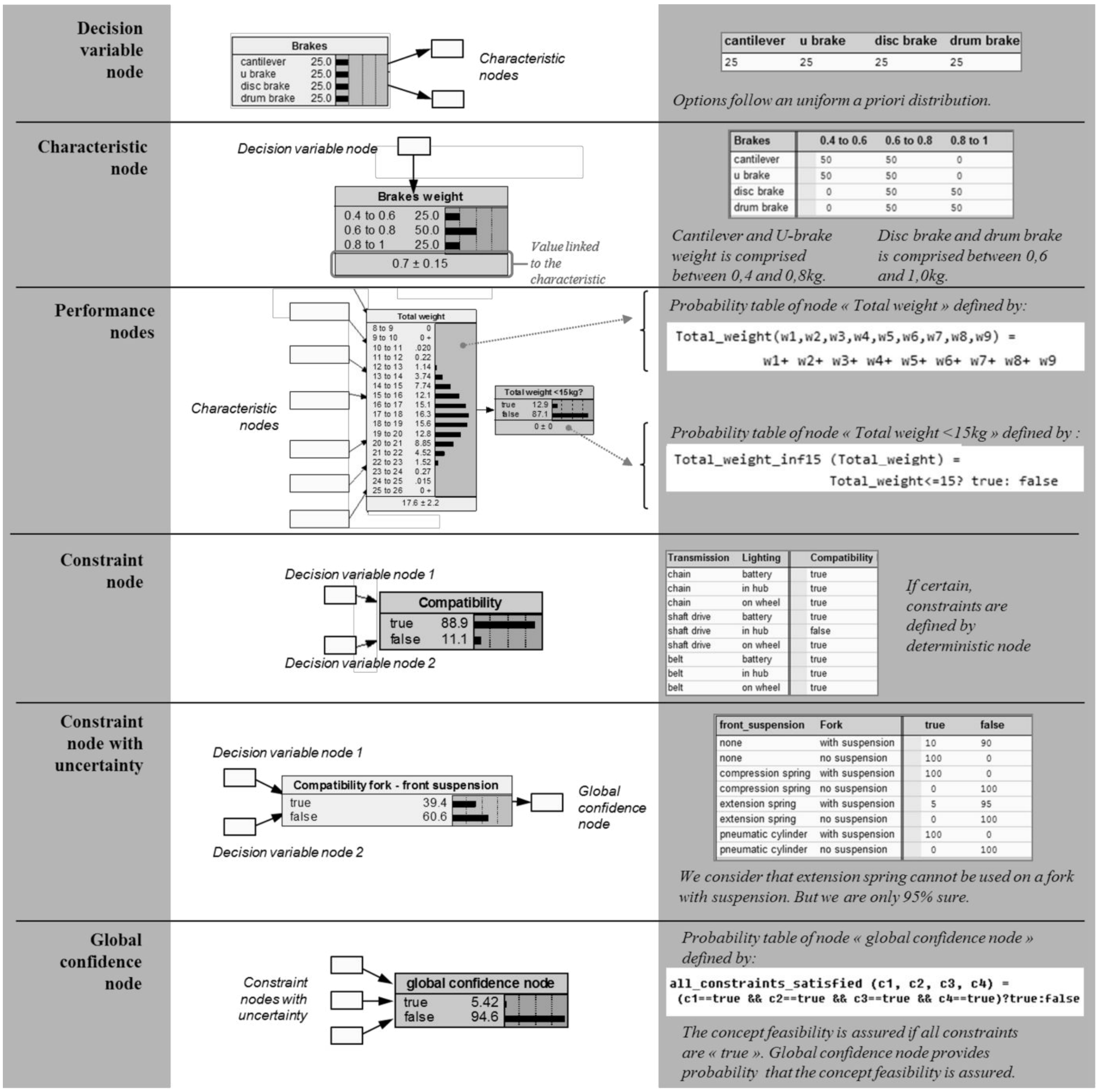


Figure 4-6. Templates for modelling a design problem using Bayesian networks

5

Paper #3. Investigating the Importance of Criteria Selection in System Architecture: Observations from Industrial Experiment

Marie-Lise Moullec, Marija Jankovic, Claudia Eckert

This paper will be submitted in Research in Engineering Design in 2014.

Abstract. *Decisions impacting the most overall lifecycle costs are those taken very early in the design process when information is still incomplete, fuzzy and uncertain. In particular, decisions related to system architecture are difficult because system architecture relates to every stage of system life-cycle which might have conflicting objectives. Most decision-making support tools and methods proposed in both operational research and engineering design fields assume that decision criteria are known and well-defined. In view to test the usability of one of this tool, a workshop was organised in industry: the objective was to choose 5 system architectures amongst 800 generated using an automated method. In the workshop the first step, the identification of selection criteria, proved to be the greatest challenge. As a result, designers selected system architectures that did not satisfy them in the end, without being able to explain what went wrong in their selection process. The objective of this study is investigating the role of criteria in the process of system architecture selection. The recordings of the workshop were transcribed and analysed in order to identify the difficulties related to the definition and the use of criteria. To this aim, a timeline was built in order to illustrate the evolution of criteria use during the workshop, and relations between criteria are shown. The process of designers reasoning is also detailed. The analysis highlights the fact that the inherent interdisciplinarity of system architecture makes criteria interrelated; and lack of information makes impossible to define an exhaustive set of criteria. These particular two points question the usability of existing decision support methods. In addition, several types of criteria were identified in view to their role on architecture selection. Finally, this study provides insights and recommendations for future decision making support tool dedicated to system architecture design.*

Keywords. *Systems design, design methodology, selection criteria, design decision making*

5.1 Introduction

Good decisions arise from the right problem definition. System development processes are shaped by different decisions starting from the choice of working principles to the parameterization of a detailed design. Numerous research studies point out that decisions most impacting the overall lifecycle costs are those taken very early in the design process (Berliner and Brimson 1988; Okudan and Tauhid 2009; Martinsuo and Poskela 2011). And because these decisions initiate the detailed development of the system, they are often irreversible. System architecture design occurs during the early stages of design process, when information is incomplete, fuzzy and uncertain (Chapman and Pinfold 1999). It is an interdisciplinary activity and relates to every stage of system life-cycle which might have conflicting objectives. Methods coming from both Operational Research and Engineering Design support designers in this process. However, most of them rely on a set of selection criteria that are already defined. These criteria are based on system requirements or generic considerations used in product development; they are directly defined by designers. Criteria support system architectures evaluation and can lead to inappropriate solutions if they are incorrectly defined, in particular when using computer aided methods, which often generate a high number of concepts among which identification of the most promising ones may be difficult.

We conducted a one-day workshop in Thales to investigate the use of multicriteria decision aid methods to select system architectures. Four experts in different system or engineering domains had to select an active antenna subsystem. We analysed the selection process as well as strengths and weaknesses related to criteria definition and architectures selection. In the workshop the first step, the identification of selection criteria, proved to be the greatest challenge. As result, designers selected system architectures that did not satisfy them in the end, without being able to explain what went wrong in their selection process. The objective of this study is investigating the role of criteria definition in the process of system architecture selection.

In the following section, an overview of multicriteria decision making methods as well as concept selection methods in product development is given to determine which criteria are customarily used in concept selection. Section 5.3 explains the study context as well as the protocol. Section 5.4 summarises the main insights emerging from the workshop. Section 5.5 analyses the potential role of criteria within the selection process while Section 5.6 discusses issues related to criteria definition. Section 5.7 provides insights regarding the requirements for future decision support system suitable for selection of complex system architectures.

5.2 System architecture selection in the literature

5.2.1 Definition of a criterion

According to Oxford dictionary, a criterion is “a principle or a standard by which something may be judged or decided”. In decision-making field, a criterion is “a function that associates each action with a number indicating its desirability according to consequences related to the same point of view" (Roy and Bouyssou 1991). A criterion may also be considered as an "attribute", an "objective" or a "goal" as emphasised by Henig and Buchanan (Henig and Buchanan 1996) or (Ullman 2002). In this study, a "criterion" is deliberately viewed in its broadest sense: it may refer to an attribute, a performance or a point of view.

A set of criteria is considered as the basis for any rational decision making. If not already given, the choice of selection criteria appears when structuring the decision problem, which is recognised as one of the critical steps in problem solving. Saaty (Saaty 2008) underlines the need "*to know the problem, the need and purpose of the selection, the alternatives as well as the criteria, their order of priorities regarding the stakeholders involved in the decision process*". He also reminds us that criteria may be intangible and sometimes have no measurements to guide the ranking of alternatives or defining priorities. Such cases occur when the selection problem is complex and ill-structured. The methods provided by MDCA and problem structuring endeavour the clarification of the decision problem by proposing tools and models allowing the representation of the problem understanding and the decision makers preferences. This process is considered to be iterative (Simon 1956; Roy and Bouyssou 1991).

5.2.2 Existing research on criteria selection and multi-criteria decision methods

The research field that focuses on criteria is Decision-Making, in particular Multi Criteria Decision Aid (MCDA) and problem structuring. MCDA mainly focuses on modelling the preferences of decision makers (Pidd 2003) and helping them find adequate solutions. Problem structuring on the other hand aims at conceptualising the decision-making process in terms of criteria, alternatives (Raiffa and Keeney 1976; Corner, Buchanan et al. 2001; Saaty 2008), and sometimes attributes (Henig and Buchanan 1996). As techniques and decision methods proposed by the two domains are numerous, Table 5–1 provides a synthetic overview of the best known methods, showing their objectives and particularities, without intending to be entirely exhaustive. A more detailed review can be found for MCDA in (Figueira, Greco et al. 2005) and (Belton and Stewart 2002); and for problem structuring in (Mingers and Rosenhead 2004).

Table 5–1. Overview of methods and tools of MCDA and problem structuring

Domain	Method	Reference	Working principle
MCDA	ELECTRE Methods	(Roy 1991)	Construction of outranking relations to compare each pair of alternatives and provide recommendations for choosing, ranking or sorting alternatives.
	PROMETHEE Methods	(Brans, Vincke et al. 1986)	Construction of valued outranking relations representing preference intensity.
	MAUT – Multiattribute Utility Theory	(Raiffa and Keeney 1976)	Elicitation of multi-attribute utility or value functions; uses identification of the most preferred or ranking of alternatives
	AHP	(Saaty 2008)	Decomposition of the decision problem into a hierarchy of sub-problems. Their relative importance is converted to numerical values to calculate a score for each alternative
Problem structuring	Strategic Options Development Analysis (SODA)	(Eden and Ackermann 2001)	SODA uses interview and cognitive mapping to capture individual views of an issue. They are used to facilitate negotiation about value/goal systems, key strategic issues, and option portfolios.
	Soft Systems Methodology (SSM)	(Checkland 1999)	The original version of SSM as a seven-stage methodology that relies on tools to express the problem situation, build conceptual models in order to finally define changes that are desirable and feasible.

Criteria definition is considered to be an important point in both MCDA and problem structuring. In problem structuring, there are two approaches (Montibeller, Franco et al. 2009) for structuring criteria: in alternative-focused thinking (AFT) criteria are defined from the characteristics that help to distinguish the alternatives. In value-focused thinking (VFT), the evaluation of criteria should reflect objectives. By contrast, the main purpose of MCDA methods is modelling decision makers' preferences with regard to a family of pre-defined criteria, which according to Bouyssou (Bouyssou 1990) must satisfy the following properties:

- Exhaustiveness: the family should contain every important point of view. This condition implies that if the evaluation of two alternatives is equal regarding all the criteria in the family, then these two alternatives must be considered as not different in terms of preference;
- Monotonicity: the partial preferences that are modelled by each criterion have to be consistent with the global preferences expressed on the alternatives. This condition implies that if *a* is judged to be better than *b* taking into account all the points of view, the same judgment will hold for an alternative *c* that is judged at least as good as *a* on every criterion;
- Non-redundancy: the family must not include unnecessary criteria, i.e. which suppression will lead to a family still satisfying the first two conditions.

If these conditions are not satisfied, decision makers' preferences are not adequately modelled and may lead to the selection of inappropriate alternatives with regard to the initial problem.

5.2.3 Concepts selection in product design

Because all "*future activity focused on the chosen alternative, uses time, money and other resource and excludes any effort on the alternatives rejected*" (Ullman 2001), selection criteria used in design decision making must be carefully chosen. To our knowledge, most of studies focus on the decision making process when a set of selection criteria is already given (Yang and Sen 1997; Girod, Elliott et al. 2003; Kihlander 2011) while very few studies are dedicated to the definition of evaluation and selection criteria in product development process, particularly in preliminary design. This phase is extremely challenging due to the inherent fuzziness and lack of available information (Yeo, Mak et al. 2004; Olausson and Berggren 2010). The resulting uncertainty makes concept selection in early design stage an ill-structured problem (Guindon 1990), where new criteria and new requirements appear as information in the process is increasing. However, uncertainty needs to be considered as much as possible, otherwise it may lead to overall project failure (Saari and Sieberg 2004).

Design prescriptive models do not attach importance to criteria definition: in systematic design, Pahl et al. (2007) emphasise the fact that evaluation criteria must be derived from product requirements in order to ensure product feasibility. Afterwards feasible concepts are selected according to "*technical, economic and safety criteria at the same time*". They also specify a number of important points in the selection and product embodiment definition, such as assembly, transport, maintenance, etc. The authors said that these considerations depend on the available information which is growing as the design choices are made. They must be integrated as early as possible through detailed studies, but it is not indicated how they are integrated as selection criteria within the development process. In this respect, Ullman (2002), when discussing the ideal engineering decision making support, suggests that a comprehensive tool "*should manage incomplete alternatives and criteria generation; and allow their addition throughout the decision-making*".

Okudan and Tauhid (2009) have identified several decision-making methods that are used in concept selection and classified them in 6 categories which are given in Table 5–2.. When a concept selection method (CSM) allows the choice of selection criteria, the set of criteria is supposed to be known and defined by decision makers. However, in previous research it is stated that criteria used in early phases often lack clarity compared to later stages where more explicit criteria can be defined (Schmidt, Sarangee et al. 2009). A number of CSM also require preferential independent criteria (Okudan and Tauhid 2009). In addition, it must be noticed that whereas product architecture design requires considering the physical properties of components, most of concept selection methods remain based only on functional considerations ignoring the potential form of components, and thus excluding related criteria. Some of them are also far too complex to be implemented in industry, particularly those taking into account uncertainty (Okudan and Tauhid 2009). In any case, little time is spent on discussing the importance of different evaluation criteria (Girod, Elliott, et al. 2003).

Table 5–2. Classification of Concept Selection Methods (CSMs) based on (Okudan and Tauhid 2009)

CSMs based on…	Principle	Reference methods
Decision Matrices	A matrix in which concepts (in columns) are qualitatively evaluated regarding criteria (in rows).	Pugh Matrix (Pugh 1991), House Of Quality (Terharr, Clausing et al. 1993)
AHP	Goal is broken into hierarchy of criteria. These criteria are ranked in order of importance using pairwise comparisons. Then pairwise comparisons of alternatives regarding each criterion provide basis for an overall selection.	AHP (Saaty 2008)
Uncertainty modelling	Concept selection methods integrating uncertainty using non-classical mathematics, probabilistic and fuzzy clustering.	(Biltgen and Mavris 2007) (Scott 2007) (Ayag 2005)
MAUT/Economic Models	An utility function is used to evaluate and rank the alternatives regarding different criteria.	(Raiffa and Keeney 1976)
Optimisation	Identifying the solutions that are not dominated regarding multiple criteria (by selecting Pareto optimal solutions)	(Mattson and Messac 2003)
Heuristics	Methods that used qualitative techniques to select some designs from billions possible.	(Buonanno and Mavris 2004)

5.2.4 Solution selection in product synthesis systems

Design synthesis, as a research field, aims at developing guidelines, methods and tools for supporting the creation of products, helping designer to explore the design space or automating design tasks (Chakrabarti, Shea et al. 2011). As the objective of some synthesis methods is to generate or select optimal concepts through fitness/objective functions (or equivalent), it is of interest to investigate what are the criteria used in this process and how corresponding alternative evaluations are made.

Product selection or optimisation in generative systems can be classified in three categories. The first category contains the methods that use a single criterion to select an optimal concept. This criterion is in general cost (Agarwal, Cagan et al. 1999). The second category is methods using multiple generic metrics. For example, Wyatt et al. (2012) generate product architectures and evaluate them according to 3 complexity metrics, representing complexity of designing a product (D-complexity), the complexity of manufacture (M-complexity) and the complexity of making a change to the product (C-complexity). These metrics are generic and can be applied on any type of product. Yan et al. (2006) propose a method that generates a number of design options and selects the best ones using a fuzzy C-means algorithm: with regard to several functional, commercial and marketing criteria, design options are evaluated in terms of importance ratings. These criteria are evaluated subjectively. Scaravetti (2004) conducted an extensive literature review with a detailed list of 28 functional, structural, economic and company or marketing-related criteria useful for product embodiment definition. Each criterion is defined by a generic formula and used as a constraint in a Constraint Satisfaction Problem. The third category of product selection method comprises the methods in which criteria are chosen by the designers. Vico et al. (1999) use genetic algorithms to generate optimal design concepts. Designers define their own criteria. Each design is evaluated subjectively by designers and these evaluations are used to improve the fitness function using a neural network. This method also works with objective criteria. Rosenstein and Reich (2011) proposes the HSoS method, an extension of SoS method (Ziv-Av and Reich 2005), in which a genetic algorithm is used to generate optimal concepts (viewed as assembly of building-blocks). Designers evaluate each building-block with regard to objective and subjective criteria, also hierarchically defined by designers. The criteria related to customer requirements and manufacturer objectives are aggregated into a weighted objective function used by the genetic algorithm. Campbell et al. (1999) propose the A-Design method, an agent-based synthesis method that transforms functions into physical configurations. These configurations are evaluated on the basis of input design specifications and sorted using a Pareto optimisation. Criteria used in this method are therefore product specific.

The literature does not point to a standard way to conduct product evaluation and selection in design synthesis. Evaluation criteria rely on product specifications, on general considerations or focus on specific product performances such as complexity or cost. The work of Scaravetti (2004) highlights the variety of points of view (Table 5–3) according to which a concept may be evaluated and the amount of information that is needed to select product architecture properly.

Table 5–3. Main criteria to define and evaluate product architectures (Scaravetti 2004)

Expected performances	Robustness	Production cost	Sustainable development
Regulation	Autonomy	Ownership cost	Environmental impact
Minimal life expectancy	Mass	Maintenance cost	Project duration
Reliability	Encumbrance	Recycling cost	Risks: design, costs, time
Availability	Resistance to the environment	Objective cost	Startup?
Maintainability	Complexity	Investment capacity	Nuisances
Security	Industrialisation	Recyclability	User perception

When criteria are chosen product evaluation can be objective using equations or be subjective. In architecture selection often a Pareto optimisation or an overall weighted function is used. In this case, Antonnson and Cagan (2001) emphasised the difficulty of capturing subtleties and complexities of practical designs in terms of constraints and objectives functions.

As previously discussed, there are not many methods that support criteria identification in concept generation; in addition most of the methods presented in Section 5.2.3 and 5.2.4 address design of simple products. However, system architecture design methods concern complex systems of which the inherent interdependencies and multidisciplinarity (Crawley, Weck De, et al. 2004) may make the definition of selection criteria more difficult. In this sense, our study was motivated with two objectives: to understand the system architecture design process currently applied in industry, but also if tools are needed to support this process. For example the method (Moullec, Jankovic et al. 2013) used in this case study generated 800 potential architectures: selecting amongst some 800 complex system architectures cannot be considered to be a trivial issue. Therefore, we propose a protocol to investigate how complex system architectures are selected in the industrial setting; as well as what criteria are used during this process.

5.3 Empirical Study

This paper is based on a one-day workshop conducted in Thales Air Systems, that develops radars for military and civil applications (Figure 5-1). These products require several years of development and have an intended life span of several decades. Military applications require cutting edge technology and therefore constant innovation. The choice of specific technologies occurs very early in the design process and may require significant investments. To provide highly reliable and safe systems and compete in the market, innovative solutions need to be considered, which are more risky than those established in the company for a long time. Radars are incremental products. To assess the impact of introducing innovations, different solution alternatives are studied at the very early design stage. Such investigations are time consuming and require a multidisciplinary investigation in order to consider interactions related to different domains such as thermal effects on electronic devices. As result, only a few architecture alternatives are studied. At the end of this process a specific system architecture is selected considering all system requirements. This decision becomes nearly irrevocable.

Figure 5-1. Radar antennas developed by Thales Air Systems

In Thales Air Systems, technical solution evaluation and selection usually takes place through peer review workshops. Typically these workshops mainly concerns subsystems and are focused on one discipline such as thermal issues or radiofrequency. The engineers therefore are familiar with the requirements related to their field of expertise. However, system architecture evaluation and selection is inherently interdisciplinary: all domains must be considered at the same time and traded-off against each other. The choice of the set of selection criteria therefore depends on performances parameters that can be assessed despite the inherent lack of information, This assessment is in addition made difficult by the complexity of system to be designed.

5.3.1 Context of the study

This paper reports on research carried out as part of doctoral research on system architecture generation and evaluation. Research study presented in this work is organised as an action based research where one of the authors has been working in the company for three years as part of an industry funded PhD research project. The aim of this work is to support engineers in system architecture generation and selection. We developed a method to automate the generation and the evaluation of system architectures (Moullec, Jankovic, et al. 2013), where architectures are automatically evaluated for performances previously specified by engineers and their component placement is optimised.

The first step of the method is modelling the design problem while considering uncertainty about performances and interfaces as a Bayesian network model for architecture generation (Moullec, Bouissou, et al. 2013). This Bayesian network model represents a design problem in terms of design decisions, constraints and performances. Design decisions relate to technologies, components and number of components that will form the final architecture. Each design decision is defined by a set of design alternatives. For example, a cooling component could be a heat exchanger, a radiator or a cooling unit. Design alternatives are described by a set of characteristics that they all have in common, like cooling efficiency in our example, and that will be used to calculate desired system architecture performances. These calculations are defined within the Bayesian network by designers. In addition, designers have the possibility to introduce constraints, such as compatibility between two technologies, or requirements on a performance. All data in the Bayesian network are expressed using probability distributions, which make possible integration of uncertainties related to component compatibilities and performance estimations. As a result, the performances of the generated architectures are represented as probability distributions. In the second step, component placement optimisation of each architecture is processed regarding performances that are related to this aspect, for example system dimensions (Moullec, Jankovic, et al. 2013). The performances optimised in this second step use data expressed as value points.

A final step consists in supporting designers in the evaluation of performances that could not be automatically estimated and in the selection of the best generated architectures, for which adequate selection criteria need to be identified.

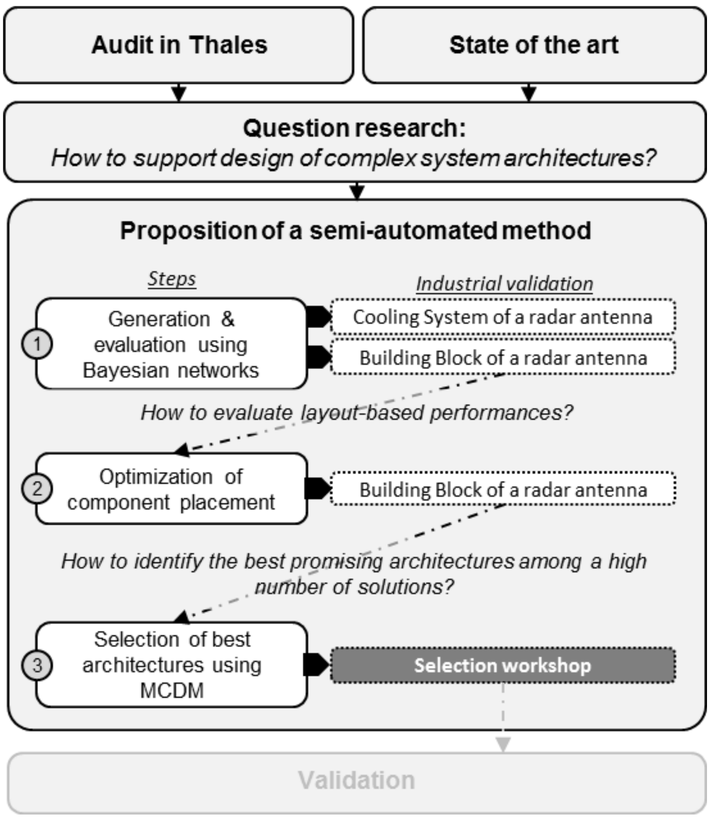
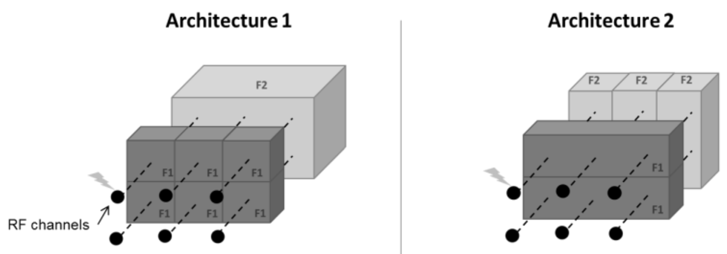


Figure 5-2. Process of overall research study

As shown in Figure 5-2, several studies have been conducted in industry setting (Moullec, Bouissou, et al. 2013; Moullec, Jankovic, et al. 2013). The present case study relates to a building block of radar antenna but focuses on innovative technologies with the objective to identify the architecture for a new generation. This system is described more deeply in Section 5.3.2. Eight hundred feasible solutions have been identified amongst 50176 possible architectures. These architectures are mainly varying according to the technologies that are used as well as different physical arrangements of parts, each with advantages and disadvantages regarding system architecture performances.

5.3.2 The complex system: radar active antennas

The basic functions of a radar antenna are transmitting and receiving electromagnetic signals to detect the presence of objects in a given area. However, to be usable, the transmitted signals must be amplified before being radiated and the received signals must be amplified before being processed. Active antennas amplify the signal in close proximity to the radiating elements within one integrated building block which incorporates cutting edge technology to carry out some of the core functions of the antenna. In this empirical study, such a building block is considered in the architecture selection process. A building block is always connected to a fixed number of radiating elements, called radio frequency (RF) channels whose the spacing is determined by the chosen signal frequency. Depending on the technology that is used, each component inside the building block may cover one or several RF channels, making the component size and number dependent on the number of RF channels (Figure 5-3). In this study, depending on the technology that is used, the building block solutions can be classified in 3 families in which the architectures vary according to decomposition patterns.



Architecture 1	Number of components	Channels covered by each component
F1	3	1
F2	1	6

Architecture 2	Number of components	Channels covered by each component
F1	2	3
F2	3	2

Figure 5-3. Variations of component size and number according to RF channels covered

This decomposition has a strong impact on reliability and maintainability costs. As to the technologies inside the building block, they mainly impact the manufacturing costs given that they represent around 60% of the total cost of the antenna; they have a considerable importance in the selection process. This is even more critical as radar antennas are generally manufactured in small series. Due to the cost, this building block is designed to be reused in other products of the same family and therefore its architecture needs to allow a certain degree of customisation in particular related to software. However because related to too many parameters cost estimation is usually not available when building block system architecture. Therefore the selection of such a building block implies several types of considerations related to technical choices, cost, and functions-components mapping to meet system requirements and company objectives. These considerations must be analysed in detail to determine a set of criteria for building block architecture selection.

Eight hundred new architectures of building-block have been generated using the method described in Section 5.3.1. In the workshop analysed for this paper, industry experts had to identify the five best architectures from automatically generated building block architectures.

5.3.3 Architecture selection workshop

5.3.3.1 Workshop objectives and organisation

This workshop had two objectives: 1) observe how engineers proceed when confronted with a large number of new solutions and criteria and 2) identify the relevance of MCDM methods in this process. The goal for the experts was to identify five architectures to study more in depth amongst the 800 architectures that have been generated using the method described in the previous section. Four engineers took part in this workshop (Table 5–4). They have been invited to participate because of their domain expertise and involvement in the overall project. The main relevant areas of expertise were covered by these four experts.

Table 5–4. Role of Thales engineers involved in the workshop

	Domain	Role	Involvement in the whole project
S1	Antenna architecture	Technology and innovation engineer in the area of antenna concepts	Yes
S2	Mechanical integration of antenna	Technical expert in charge of antenna integration, very familiar with thermal and mechanical problems encountered in recent years	Yes
S3	RF studies	Expert in radio frequency (RF) studies, in charge of feasibility study concerning RF components	No
S4	Radar architecture	Radar architect, expert in functional architecture of the radar, and thus, interfaces between the antenna and the rest of the radar	Partial

For architecture selection two expert groups were set up to select architectures separately (one using a MCDM, the other without MCDM), who were brought together to compare and rank the whole set of selected architectures. This allowed evaluating the efficiency of the MCDM used by the first group while studying the selection process used by the second group. Therefore, the workshop was organised in four different phases (Figure 5-4).

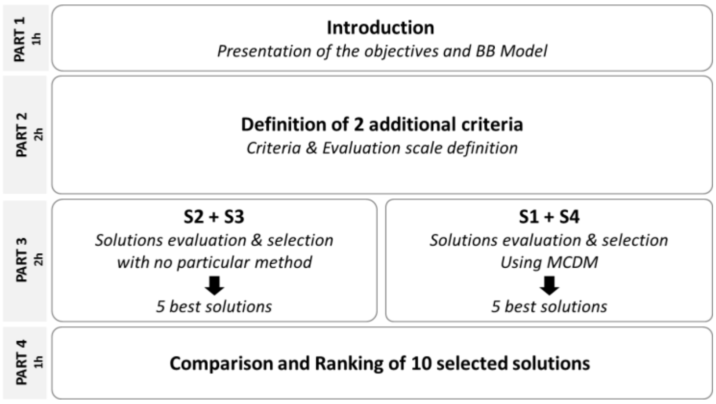


Figure 5-4. Workshop program

The introductory session explained the workshop objectives, showed the software for system architecture visualisation, and allowed time for questions. As the MCDM requires selection criteria to be defined, in the second part of the workshop the idea was to define this set of criteria. Four overall criteria were estimated by the BN method (temperature, pressure loss, volume and mass). All participants decided that more criteria were needed. Due to time constraints a decision was taken to look for two additional criteria. In the third part of the workshop, the experts were divided into two groups: one group evaluating and selecting architectures without any method, and one using the MCDM. These groups were put together according to two considerations: one member of each team should be acquainted with the methodology and the BB model, and the two members should have competencies as complementary as possible. Thus, S1 worked with S4, and S2 with S3. Each team had to propose what they considered the five best architecture solutions. In the last part of the workshop, the objective was to rank all ten in view to their preferences.

5.3.3.2 Available data during the workshop

Possible architecture solutions were presented within the software specifically developed for the company. In addition a spread sheet with all performances estimation has been made available to experts. For all possible system architecture, the following information was available (Figure 5-5):

1. The elements of the architecture were shown in a schematic view as customarily used by Thales engineers.

2. Second view presents the performances of the architecture. Performances estimated by the Bayesian Network were given as probability distributions, while performances depending upon component placement optimisation were given as a single value point (see Section 3.2). For example, volume was optimised for a given architecture, whereas the pressure loss and the air temperature inside the building block were probability distributions (Figure 5-5).

3. Last view shows a 3D-visualisation of the solution following the configuration proposed by the optimisation system. However, optimising the component placement of 800 solutions, even fully automatic, takes time. As consequence, some of the solutions did not have an optimised placement, and some others (60%) had no visualisation at all.

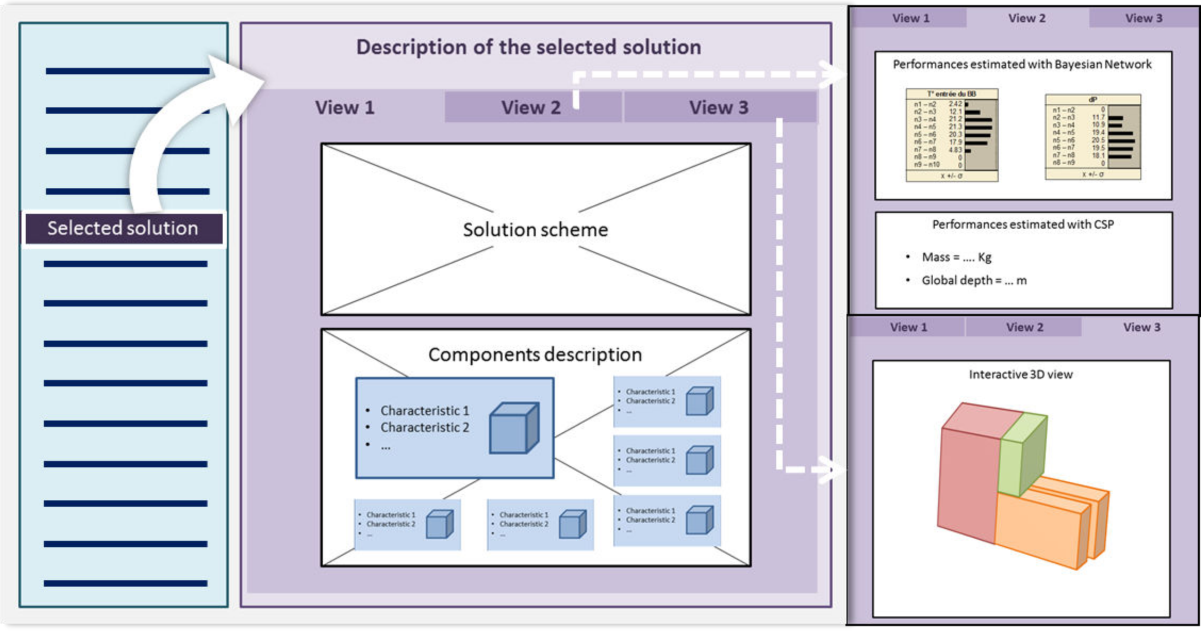


Figure 5-5. Developed user interface for system architecture visualisation

5.3.3.3 Workshop observation data gathering and research methodology

The entire workshop has been video recorded. Besides the session using Promethee, all sessions have been transcribed using Sonal (Sonal 2009). Following the workshop, the authors organised 5 half-day sessions to listen and discuss workshop. In order to diminish the bias in data coding, two authors read through the transcript to code the occurrence of criteria. After the first analysis two interviews were conducted with engineers in order to clarify some points or to better understand underlying issues. The overall aim of the analyses was to identify how and which criteria were used during selection process. The first analysis concerns the architecture selection process and the order in which selection criteria appear and are considered in the discussion. Then, an analysis of the number of occurrences of given criteria and their interrelatedness have been visually represented using Gephi (2008). Finally, a selection process has been abstracted.

Figure 5-6 explains the process by which the data was analysed.

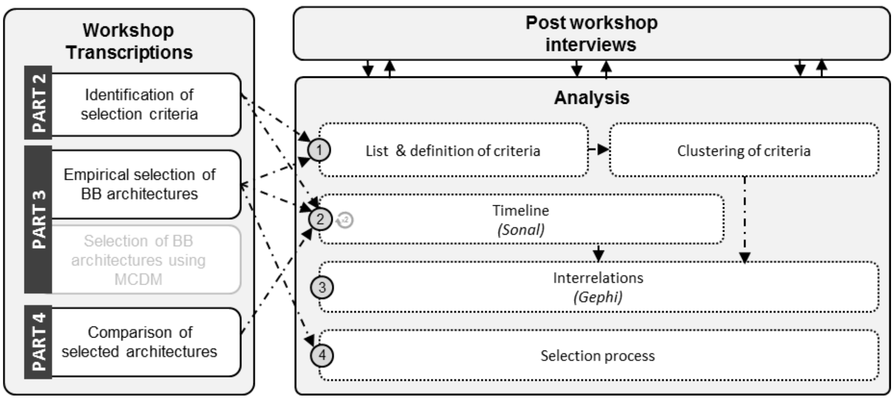


Figure 5-6. Data analysis process

5.4 Architecture selection criteria in the workshop

The first analysis concerns the process of criteria identification and the order in which they appeared. The definition of two additional criteria is then discussed. A list of all criteria mentioned in the workshop is given with a short definition. Finally, the architecture selection process adopted by the experts is explained.

5.4.1 Process of criteria identification

Based on the transcripts of the workshop, a timeline (Figure 5-7) representing the evolution of the criteria that have been mentioned during the identification stage of the two new selection criteria has been drawn. On the timeline the vertical line shows the order in which the criteria like cost, number of elements or manufacturing came up. Figure 5-7 shows first 20 minutes where experts discussed possible criteria before entering in the definition of complexity. The video recording has been divided in several extracts according to three categories according to the information the experts are referring to:

- *Example* refers to hypothetical case is given to explain a criterion or a relation between two criteria.
- *Concept* refers to Thales engineers reasoning about these criteria and their mutual effects.
- P*ast experience* refers to the times in which past products are remembered in order to identify real reference points.

A dot in the matrix represents a reference to the corresponding criterion. Several dots in a single column mean that several criteria were addressed at the same time.

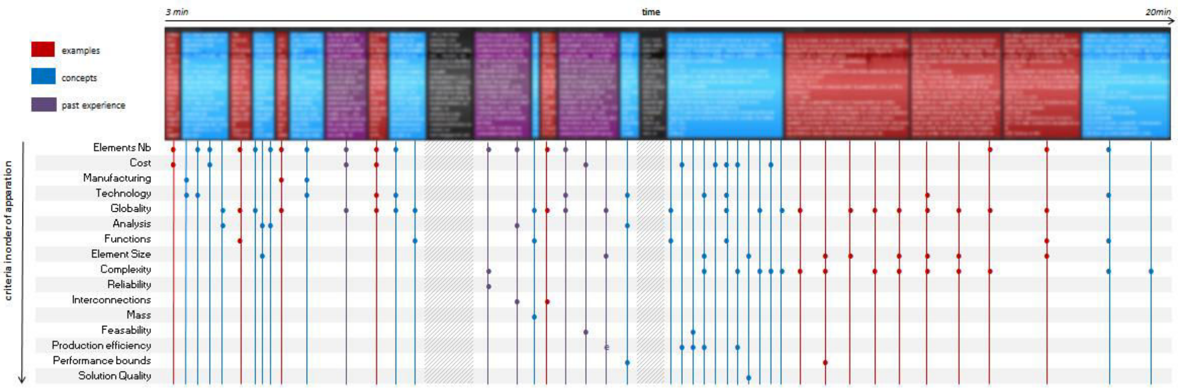


Figure 5-7. Timeline of identification of the new criteria

5.4.2 The choice of two new criteria

The information on the proposed architectures mainly concerned their composition, in terms of type and number of components, as well as four performance factors (mass, temperature, pressure losses, depth). The choice of the new selection criteria has been, above all, driven by cost considerations. However, the difficulty faced by experts was that sufficient information was not available to evaluate cost. To compensate for missing information the experts started by analysing the way the alternatives of architectures varied. A component generally realises 1 or 2 functions depending on the technology that is used and the number of components is declined in several patterns. From this the notion of "globalité" appeared. Based on the experts' definition, "globalité" refers to the physical and functional decomposition of the architecture and has two aspects:

- A functional aspect which means that a same component may comprise several functions, and addresses the notion of function sharing. For radars, this is very dependent on the technology that is used.
- A component aspect which means that a function for the whole set of RF channels may be realised by several components. The number of components is limited by the technology that is used in the architecture and the size of the element will depend on the number of RF channels (see Section 5.3.2).

In order to link the criterion "globalité" with the criterion "cost", the experts defined the criterion "complexity", to show how cost is evolving through variation in both the functional and the component aspect of "globalité". "Complexity" reflects the difficulties related to the manufacturing of each component depending on the number of functions and the number of channels. The assumption made by the experts is that cost increases with complexity. To be objective they did not want to express a personal opinion, which led them to define an evaluation function.

The development of the equation took about 1h30. At first, experts examined a number of typical examples of architecture based on the number of functions integrated in the technologies of the architecture, as well as on components number. Two typical examples were 1) an architecture using a technology integrating all functions and composed of 2 components; and 2) an architecture using technologies integrating only one function and composed of 140 components. Finally, complexity C was defined with the following equation (1).

$$C = \sum_{t \in T} \alpha_t * Nb^{t}_{channels/component} \qquad (1)$$

with **T the set of technologies used in the architecture;**
$Nb_{channels/elements}$, the number of channels covered by one component;
α_t the coefficient of complexity related to the technology.

For a technology t used in the building block, α_t is a coefficient representing the complexity of the technology used in one type of component (more functions are integrated in the technology, more the component will be complex to manufacture). For example, α_t is higher for a technology integrating 5 functions than another technology integrating 2 functions. In addition the component may be replicated in order to cover the required number of building block channels: the experts considered that more channels were covered by a single component, more difficult this compoenent was to manufacture. Therefore, the whole complexity of one component is estimated by the product of α_t with the number of channels covered by the component. The complexity of the architecture consists of the addition of each technology in the architecture. Then, the experts adjusted the coefficients α_t by testing several values on the whole set of architectures. Once coefficients α_t are set for the different technologies, the experts used the spreadsheet to calculate the complexity values of every architecture. At the end, the value range for the criterion complexity spanned from 18 to 448.

In addition, the experts chose total "number of elements" as a second criterion. It also represents the impacts of the number of elements on the cost. However, "number of elements" and "complexity" are conflicting. The timeline (Figure 5-8) shows the process of mapping and convergence towards the 3 criteria which are "complexity", "globalité" and "number of elements".

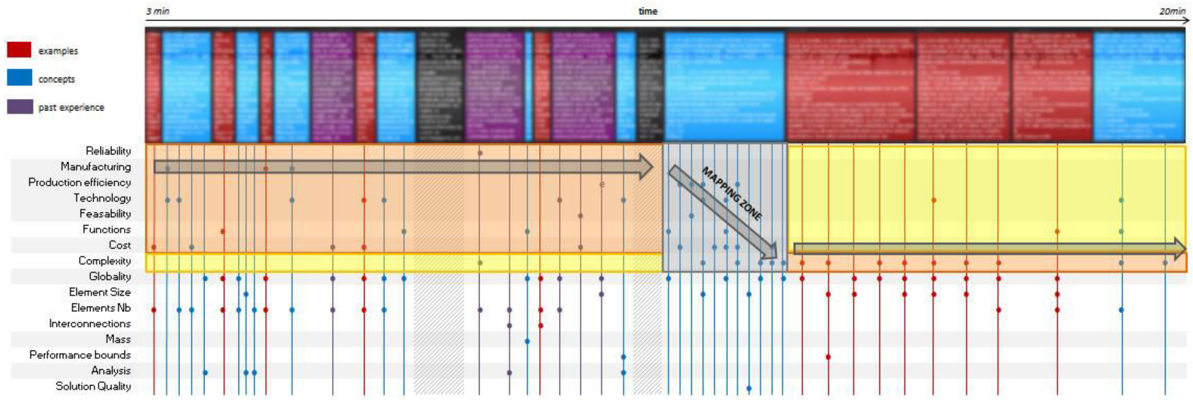


Figure 5-8: Convergence process towards the new selection criteria during criteria definition

5.4.3 The overall set of criteria

Throughout the workshop, i.e. Part 2 and Part 3, a total of 36 terms have been used by experts to describe criteria selecting an architecture. Some of them are synonymous or only slightly differ in their meaning. We grouped the terms considered as synonyms. The appendix in Section 5.10 lists all the terms used as criteria, shows how they were grouped and gives a quick description for each group of criteria. All the terms used in the workshop were in French. We translated them as precisely as possible, however some French concepts were difficult to translate in English. For example, "globalité" has no equivalent in English. This term has been chosen and defined by the engineers in order to designate a very specific concept. Even in French, one cannot fully understand the underlying concept of "globalité" without the explanation surrounding it. We have tried different words like "globalisation" or "inclusiveness" without finding the one that expresses its whole signification. We have therefore decided to keep the French term.

5.4.4 Architectures selection and comparison

The experts had to choose the 5 "best" architectures amongst 800 in about in two hours. The set of criteria for architectures selection was derived from the performances estimated by the generative system, i.e.

"mass", "global depth", "temperature", "pressure loss", and of two additional criteria chosen by the experts, being "complexity" and "number of elements". In addition, the experts wanted to select architectures belonging to the three families. This requirement was represented by the criterion "diversity of solutions".

The architecture selection was carried out in two phases (Figure 5-9). First, a pre-selection based on the criteria "mass" and "temperature" resulted in 100 potential architectures. These criteria were primarily used because of their selectivity and relative easiness to define thresholds given that they refer to system requirements. Nevertheless the criteria thresholds had to be revised several times in order to ensure solution diversity. This stage lasted about 40 minutes. At that time, the retained architectures had very similar performances in terms of "global depth" and "pressure loss": these performances were very dependent on architecture families and no differentiation was possible unless going beyond the criterion "diversity of solutions". Therefore, the experts decided to filter them according to the criterion "complexity". However, they had experienced difficulties determining a threshold value perceived by them as "completely subjective". Finally, the median of complexity values made by retained architectures was adopted as filtering threshold. At this step, each architecture family was represented in the set of retained architectures, but one family was represented by only one architecture, making it definitively selected. During the remaining hour, the experts focused on determining the 4 other architectures, mainly based on their configuration (physical decomposition). However, this second selection stage involved a number of difficulties that raised issues like the relative importance of the criteria "complexity" and "number of elements" as well as the regrettable absence of cost evaluation. The criteria "reliability" and "feasibility" came up again during the final moments of the selection.

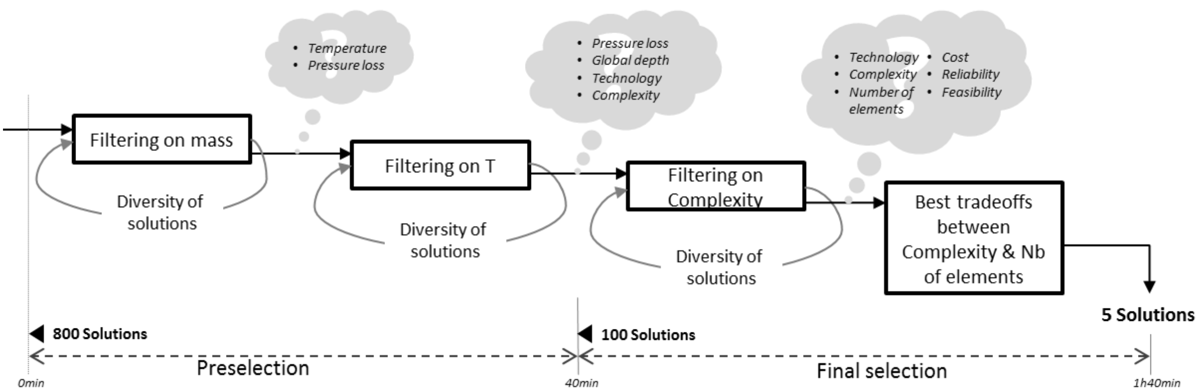


Figure 5-9: Two stage architecture selection process and related criteria

5.5 Analysis

5.5.1 Different purposes and scope

The criteria mentioned by the experts resulted from discussions and debates that occurred throughout and sometimes before the workshop. This collective reasoning is an alternation between references, hypothetical examples and past experiences as well as moments of conceptualisation (Figure 5-7). As result, experts defined the set of criteria referring to various aspects of the architecture and constituting the scope of criteria definition (Figure 5-10). They reflect different points of view to be considered, but also relate to attributes and performances of a possible architecture, or conditions on the selection process. Five areas were considered:

- configuration: all the components (and their characteristics) that constitute the architecture;
- quality of information: it mainly refers to the quality of data used in the generative model;
- properties: all the performances estimated by the model;
- objectives: all considerations coming from the context, e.g. company objectives, system requirements, specific points of view;
- set of final architectures: all the constraints like "diversity of solutions" that the set of chosen solutions must meet.

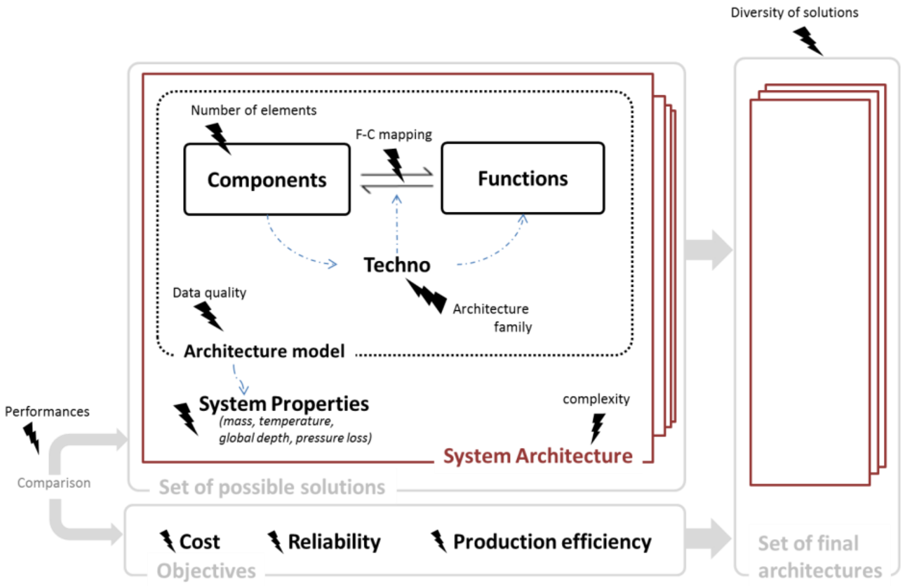


Figure 5-10. Scope of criteria definition

During the workshop, engineers mentioned them at any time, without any particular order or strategy. As the discussion went on, various criteria appear and then disappear without a discussion or consensus to determine their role in the selection process. Some issues like lack of information and inaccuracy in the performance estimations are approached but never resolved in a definitive way. This did not necessarily supported engineers in the convergence towards wanted solutions (see Section 5.5.2).

5.5.2 Interrelations of criteria

The selection process is made difficult to structure because experts often addressed a criterion in association with another one. A total of 56 relations between criteria has been mentioned by the experts, out of which some conflicting trends emerged. Based on the transcripts, the relations identified by Thales engineers are illustrated in Figure 5-11: the importance of criteria and their interdependences are shown by highlighting relations that express consistent or conflicting objectives. These criteria interdependences increase the difficulty for experts to express their preferences. In addition, experts mentioned the fact that concessions on cost could be made provided that gains regarding another criterion were sufficient. This directly violates the MCDM postulate of independence. The other postulates of MCDM are criteria non-redundancy and exhaustivity. Cost assessment is not possible, which make the postulate of exhaustivity impossible to meet. The criteria finding process shows that criteria non-redundancy is likely to be difficult to satisfy in an industrial setting.

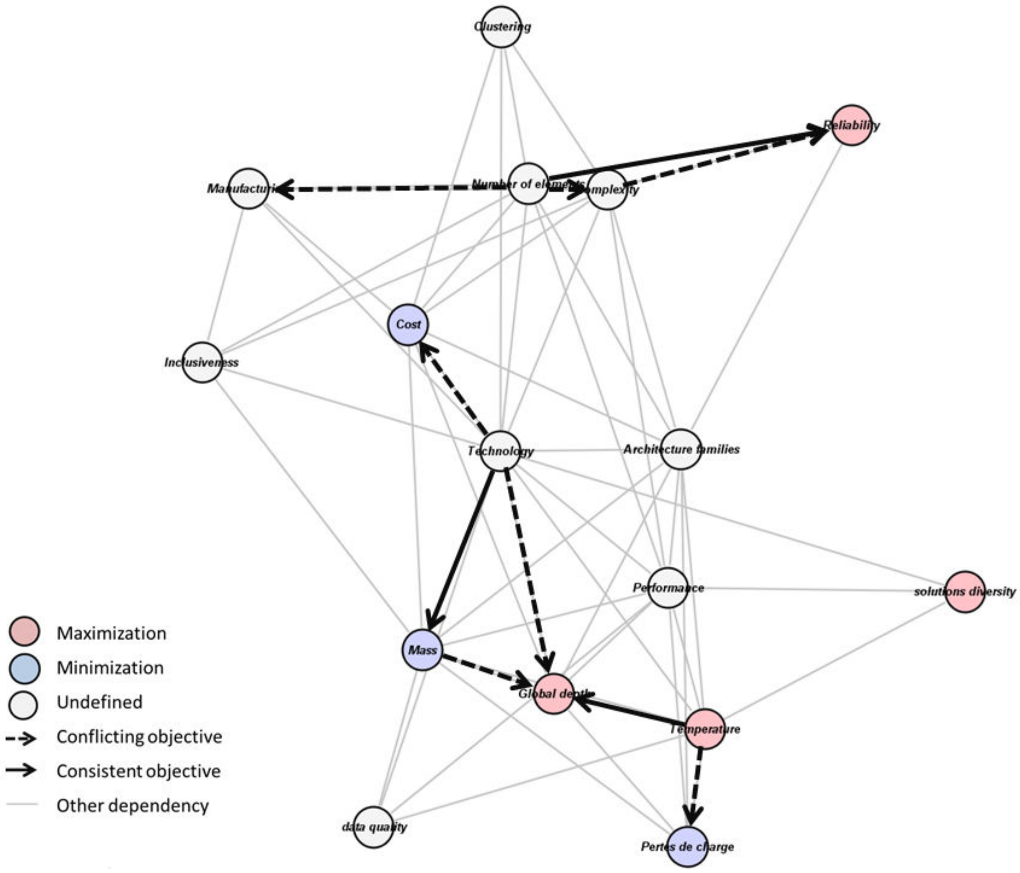


Figure 5-11: Criteria interdependence analysis

5.5.3 Architecture selection strategies

5.5.3.1 Criteria utilisation pattern

One of initial objectives of the workshop was to observe the architecture selection process "as is". A particular point that was analysed is utilisation of criteria. There is a clear pattern observed during this workshop (Figure 5-12). In the first step, a criterion is considered with regard to the performance ranges of considered architecture solutions. But this choice also depends on the relative importance of the criterion, as well as the number of solutions that it can remove. Given that the initial number of architectures is very high the experts tried to reduce the number of potential architectures as quickly as possible and therefore preferred very selective criteria. Once a criterion was chosen, ranking (for quantified values) or classification (for categories) was made depending on criterion type. Then, engineers determined admissible categories/performance thresholds. In the case of quantified values, they choose either to use a filtering threshold, or to retain the *n* best ranked solutions. The resulting set of solutions is then analysed. If the set of possible architectures globally seems to be satisfying requirements and target performances, the selection process is continued. Otherwise, the previous filtering was cancelled, and another threshold was tested or another selection criterion is looked into.

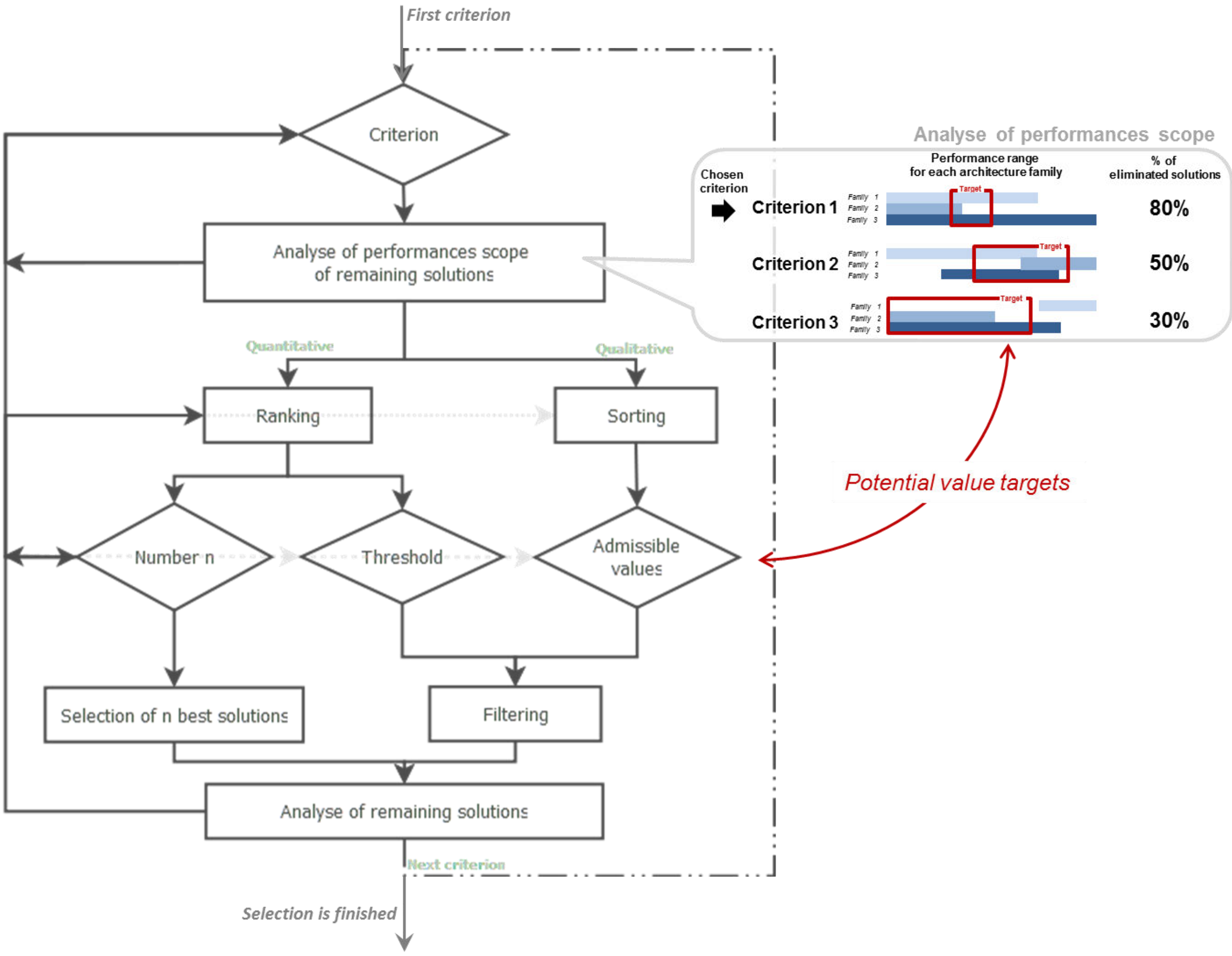


Figure 5-12. Abstracted process of criteria utilisation

5.5.3.2 Difficulties encountered

At the very end of the workshop, experts talked about their difficulties in architecture selection. As shown in Figure 5-9, each of selection steps was preceded by indecision phases as to the order and the weights to be given to criteria.

Difficulties in selecting the right criteria

While they have been chosen as the two additional criteria, the criteria "complexity" and "number of elements" have been used only at the end of the architectures selection process, making it difficult to determine trade-offs and raising doubts over their suitability for architecture selection. Figure 5-13 shows the frequency distribution of criteria use during the workshop: during the definition of additional criteria, it is observed that the experts progressively focused on "complexity" and "number of elements". This suggests that "complexity" and "number of elements" would have a major role to play in the selection process. But Figure 5-13 also shows that these specific criteria were not used more than the other criteria during the selection. Contrarily, all the criteria discussed at the beginning of the workshop reappeared. This is reflecting the difficulty of choosing the final architectures and the doubts over the criteria previously defined.

Figure 5-13. Frequency distribution of criteria during the workshop

Difficulties in using criteria

Once a criterion was chosen the experts struggled with setting filtering thresholds for all criteria. They faced three main problems:

- Conflicting criteria: for a given criterion, although they knew a reference value to apply as a filtering threshold, they could not use it without violating the criterion "diversity of solutions", which required that all architecture families are represented in the set of selected architectures. However, the families had very different performance ranges (Figure 5-132), making an acceptability thresholds hard to define. The question of whether solutions should be divided according to architecture families arose. The experts finally chose to evaluate all the solutions at the same time rather than making three distinct selection (one per family), which caused numerous iterations.
- Lack of preferences: when the experts they did not know what value to choose, i.e. for "complexity" or "number of elements" they finally opted to retain solutions around median values, which led them to select final architectures that have very similar performances.
- Criterion with low weight: Sometimes, the considered criterion was not significant enough to remove architectures using a crisp threshold, and thus to risk removing architectures that could be better regarding other criteria. In these cases, Thales selected the *n* best architectures regarding the criterion.

The transcripts indicate that the choice of a particular threshold is also influenced by criterion selectivity. This is mainly due to time constraints as well as the essence of the selection problem which consists in identifying the 5 solutions out of a total set of 800 potential solutions.

Difficulties in using the right information

An interesting point that was observed concerns data that were used for architecture selection. Due to the number of generated architectures, engineers reasoned solely on performances (using the spreadsheet), while they had the possibility to visualise architectures. As a result, they "discovered" the appearance of the architectures that they had selected only during Part 4 (when discussing and comparing chosen architectures). At this time, they seemed disappointed saying that those were not necessarily the directions that they would have taken in the current architecture design. The doubt was confirmed by the fact that the experts have been unable to rank or express preferences among the selected solutions (with and without MCDA), including for those belonging to the same family. Therefore, architecture visualisation seemed to hold more information and more meaning that a table summarising architecture descriptions and performances. Two hours were necessary to determine which architectures to select, while two seconds of looking at the visualisation were sufficient for the engineers to say that these architectures were finally not as good as they thought.

5.5.4 Classes of criteria

Even if only a subset of criteria has been effectively used in the selection process, all criteria discussed in the Part 2 of the workshop have had a specific role in the selection process: peripheral, key or meta.

5.5.4.1 Peripheral criteria

Criteria like *manufacturing* or *reliability* have been mentioned only a few times during the workshop, and were not involved during the selection process. This is due to the fact that they may be represented by or included in other criteria (e.g. manufacturing in the cost) or just not being considered as important as other ones. However, these criteria arise from experts' experience and represent objectives initially addressed at the beginning of the workshop. They constitute an initial basis to identify a complete set of "key criteria".

5.5.4.2 Key criteria

The previous section emphasises the fact that only "mass" and "temperature" are used easily in the selection. These values are related directly to system requirements and are at the centre of concerns in engineers' daily work. Moreover, they are estimated during architecture generation, which means that the Thales engineers had identified these values as particularly interesting for architecture selection when building the generation model of building blocks. When asked why these criteria are important, engineers point to larger considerations:

- "Mass" and "global depth" impact the system deployment;
- "Temperature" relates to system deployment and reliability;
- "Pressure loss" refers to difficulties encountered in past developments.

These properties of the system reflect larger considerations than the value itself acting as proxies that link system architecture with architecture goals and allow anchoring selection process in objectivity.

5.5.4.3 Metacriteria

The engineers chose to deal with criteria that at least fulfil the following conditions, thus introducing criteria on the criteria. These metacriteria are:

- Assessability: the criterion "cost" has not been chosen because the experts did not know how to estimate it. By constrast, "mass" and "temperature" were easily calculated during the generation process of the architectures, because the engineers had sufficient data to evaluate them.
- Measurability: the experts insisted to define a formula in order to quantify "complexity", and refused to scale it numerically, which was important to maintain objectivity.

In addition, the criterion "diversity of solutions" is also considered as a metacriterion due to its impact on the whole selection process; as well as the criterion "cost" which drove the choice of the two additional criteria.

5.5.5 Conclusions of analysis

Observations from the architecture selection process have underlined several difficulties. The first one is related to the definition of criteria. They are all interrelated and thus interdependent. Moreover, we have identified three "types" of criteria. Their role in the architecture selection process is important and therefore can have impacted the organisation of this process. The notion of solution diversity and families is underlying this importance. Another fact seems to be obvious "a posteriori" but has a significant importance: once the architecture selection process has been conducted, the criteria definition process seems to be able to begin. Without confronting all this criteria, and having the possibility of exploring them through trial and error, it is not possible to have a sensible architecture selection process. Metacriteria have a significant importance in this. The existence of metacriteria was not obvious at the beginning, but can have a significant impact onto the process.

5.6 Discussion

5.6.1 Applicability of PSM and MCDM

The experts used four criteria to select architectures in a sequential way, which gives a high importance to the order in which the criteria are involved within the selection: when sequential, the weight of criteria varies according to their order in which they appear in the selection process. This is made more critical by the fact that filtering thresholds are not chosen only according feasibility requirements, but in relation to the performance range of the retained architectures (Section 5.5.3.1).

The selection process followed by the experts has disadvantages that could be avoided using a multicriteria decision method (MCDM). Indeed, most of MCDMs require determining in advance the weight of criteria and allow using as many as criteria as needed. But using MCDM implies that the set of selection criteria meet the conditions of exhaustiveness, monotonicity and non-redundancy. While it might be possible to come up with a family of criteria that are not redundant, it is difficult to imagine how the exhaustiveness and monotonicity conditions (see Section 5.2.2) could be satisfied when applied to system architecture selection. The impacts of system architecture seem too large to be qualified in an exhaustive way. In addition, the lack of information inherent in the system architecture definition generally does not allow the assessment of the impact of each variable. In our workshop, system architecture was only described by 4 performances and 7 design decisions each described by at most 5 characteristics (Figure 5-14). The lack of information limits reliable estimation of reliability or cost. To choose of 5 out of 800 architectures, the best solution may have been to anticipate and define selection criteria before architecture generation, or reconsider the BB model in order to gather data that is missing for solution evaluation with regard to these criteria. Problem Structuring Methods (PSM) that support definition of stakeholders who take part in the decision as well as to structure a problem in terms of criteria, attributes and alternatives could be a manner to clarify as much as possible the problem.

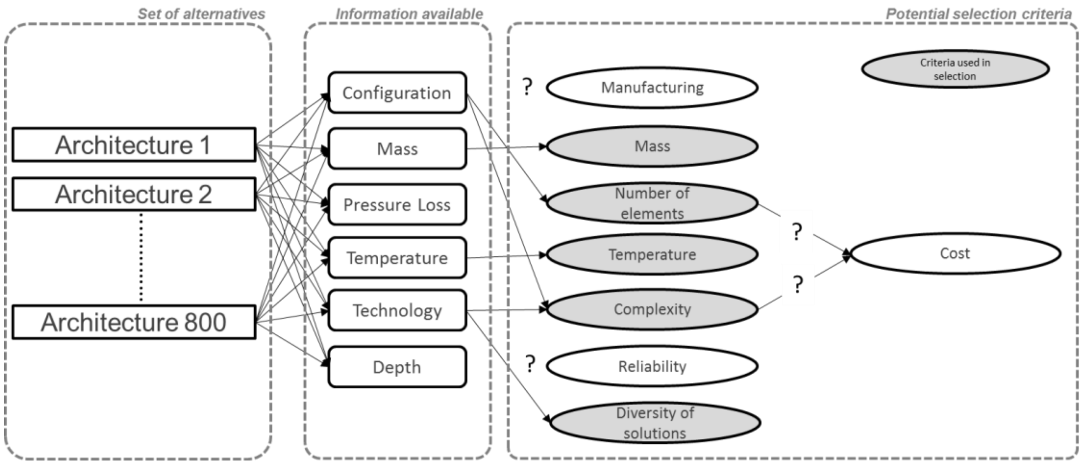


Figure 5-14. Elements involved in the selection of building blocks architectures

5.6.2 Use of generic criteria

Research in engineering design proposes sets of criteria on which the architecture selection process should be based. These criteria range from general criteria like cost or number of components (Pahl, Beitz, et al. 2007) to very specific criteria that depend on the lifecycle that have to be taken into account (Scaravetti 2004). However, the lack of information does not always permit the evaluation of these criteria; and sometimes the criteria themselves are not appropriate. For example, many complexity metrics have been proposed in design field (Summers and Shah 2010). In this workshop, the experts defined a complexity measure that decreases with the increasing number of elements (Equation (1)): when defining the criterion "complexity", the experts had in mind issues of industrial feasibility and cost, and therefore considered the internal complexity of the architecture components, rather than the complexity of the architecture itself. This is very specific to the electronic application and runs counter some other complexity metrics (Bashir and Thomson 1999) which increase with the number of elements. Therefore, the complexity metric defined by Thales experts cannot be extended to every system. Likewise, air temperature would have never appeared in a set of generic criteria. However, in the case of the building block, the temperature has to be a criterion because reliability strongly depends on it.

In view to this, contextualisation seems necessary to identify criteria, especially when information is lacking, like in system architecture design. The experience of engineers and previous designs point to issues that played a significant role in the selection or rejection of specific architectures, and aided the identification of the main elements that merits consideration. Remembering major complications due to the choice of a particular architecture serves as a reminder of new constraints or preferences. However, as one of the Thales engineers explained after the workshop, a major part of the shared information is implicit. This may lead to different interpretations from experts and examples are critical to ensure a common understanding within the experts group. Hypothetical or real examples of the considered system allow building a specific evaluation for each chosen criterion according to the information that is available. This way to build and share a common understanding of the selection problem is often part of PSM and MCDM and is strongly recommended.

5.6.3 Handling criteria

When setting the appropriate criterion, one also needs to choose a scale according to which the criterion will be evaluated. This choice are of importance since it could compromise the effective use of the criterion: as instance, when defining "complexity", Thales engineers refused to give personal opinions in order to maintain objectivity (Section 5.4.2) and preferred to develop a formula quantifying complexity, but finally never used this evaluation because it was *"making no sense"* and defining a complexity threshold "*was totally subjective"*.

While it is true that quantitative criteria present various advantages such as allowing optimisation, ranking and statistical analysis, it is not necessary the most suitable way to handle fuzzy and conceptual criteria like "complexity". An ordinal classification, for example *[too high; high; medium; low; too low]*, may have been easier to handle in that context since it would have prevented the experts from wondering whether a difference of 5, for example, is important or not when comparing two architectures. Unlike equations which depict only the explicit elements identified by the experts, ordinal classification present the other advantage of keeping the overall tacit content of the criterion. In addition, both formula and ordinal classification are subjective because both were developed by the experts in a subjective manner. However, preferring a formula rather than a classification arises from the experts needing to evaluate 800 architectures. Using a formula bypasses the questions of number of evaluators and the weight attributed to each of them (if they are specialist or not) by establishing a consensus on the evaluation of criteria.

5.6.4 Bias introduced in the workshop

The issue related to the evaluation of a high number of architectures is very specific to our study, as the use of computer aided method for design space exploration resulted in a problem that would not exist in usual system architecture selection. In general, designers mainly aim at finding one or a few "satisficing" architectures. The concept of "satisficing" has been introduced by Hebert Simon (1956) and illustrates the fact that people are often ready to accept a "sufficiently good solution" instead of an optimal solution if it costs too much time or too many effort. This means that designers generally choose one of the first solutions that meet all requirements. The purpose of the workshop was rather to identify the 5 best architectures to be studied further in depth.

This workshop was "only" an exercise. It is not sure that the experts would have been so inclined to avoid conflicts in a real situation of system architecture selection, when their own responsibilities come in line. Perhaps they would have more insisted on imposing their own point of view. In such cases, contextualisation would play a major role in determination of selection criteria, and the time devoted to this selection would probably increase considerably. In return, the short duration of this workshop might have biased the engineers towards hasty choices of selection criteria and/or use of evaluation formula in order to quickly sort the architectures and save time. The number of criteria would have not been limited and the selection may have been improved.

However, we believe that the situations observed in this study are still representative and even magnified in real circumstances. This exercise revealed the difficulty of choosing system architecture, and the complexity of the reasons that motivate the choice of a particular architecture. System architecture selection needs specific support that cannot be entirely covered by current problem structuring and multicriteria decision methods. We propose in the next section requirements for such a selection method.

5.7 Implications

5.7.1 Desired characteristics of criteria

The observations and the analysis stemming from this workshop suggest a set of desired characteristics for selection criteria. Regarding the difficulties and the facilities encountered by the engineers during the workshop, an "ideal" criterion for system architecture selection should be a property or an attribute of the system architecture which is, if possible, representative of a single objective. If it is integrated or related in several objectives, the latter must not be conflicting. In this sense, a preference (maximisation or minimization) would be clearly identified, and would remain consistent in case of multiple objectives. The criterion must also be measurable in a way that makes sense for the experts. Indeed, measures that have been built for fuzzy criteria evaluation, like "complexity", were not used because they lacked of meaning. In addition, the criterion should also be assessable, which means that one must ensure that the information required to evaluate the alternatives regarding this criterion is available. Finally, the set of criteria should contain enough criteria to address of all the interesting aspects of the problem.

5.7.2 Implications for criteria definition

In view to the discussions in previous sections, it becomes clear that finding criteria that satisfy these characteristics is not easy. First of all, the architecture selection problem must be understood in its entirety, which is not facilitated by the wide impacts of system architecture (Fixson 2005). In general, one cannot just evaluate system architecture according to a list of generic criteria because they turn out to be inappropriate, intangible or even impossible to be assessed in view to the information available for the considered system. Moreover, these criteria are in general fuzzy and ambiguous, which leads to different interpretations from experts. At least, selection criteria need to be customised according to the system being evaluated. For that purpose, the process of alternation between moments referring to "past experience", "conceptualisation" and "examples" is particularly appropriate. References to past experiences allow identifying key points to remember. Problem definition may also be sparked by the list of generic criteria found in literature which will ensure that no critical aspect of the problem is forgotten. Visualising is a way to share a common understanding of the problem (Kihlander 2011). Cognitive maps, in general used in Problem Structuring Methods, provide a visualisation of the problem and of the interdependences between the criteria. In view to the amount of considerations involved in system architecture selection, a prioritisation seems necessary and must be done regarding main objective, but also available information. For that purpose, a difference and a link between the criteria and the attributes describing the architectures must be done. This mapping is subjective (Henig and Buchanan 1996), but allows visualising how criteria may be assessed and what additional information should be eventually gathered. We propose to choose architecture attributes as selection criteria, given that they are measurable and assessable. However, they have to be carefully chosen in order to reduce the number of interdependencies and being usable. Keeney (2005), when looking into a general decision-making process, provides advices on the nature of criteria to be chosen (natural, proxy or constructed), and proposes a method that help the experts to define usable criteria. This process is iterative and aims at improving the understanding of the architecture selection problem.

5.7.3 Implications for architecture selection

System architecture selection mainly depends on the system that is studied, the information that is known, the selection criteria that are used as well as the constraints applied on the selection process. In a general manner, anticipating the selection problem before generating the architectures would allow integrating criteria from the very beginning of the generation process; and reducing the lack of information as well as the number of iterations during the selection. The uncertainty underlying the system architecture selection tends to encourage the development of interactive selection method in which the designers can add or remove alternatives and selection criteria. This is useful to gradually manage the interdependencies between criteria.

Architecture selection should integrate methods coming from project portfolio selection problems. None of the CSMs inventoried by (Okudan and Tauhid 2009) take into account this kind of constraint, given that they mainly aim at finding an optimal system. However, determining a set of projects to be further developed is becoming more frequent with the development of set-based design approaches. In our workshop, integrating such considerations (by satisfying the criterion "diversity of solutions") induced many problems and iterations during the selection process, because the experts did not know how to apply it. Finally, the introduction of computer aided method require additional features like clustering and Pareto optimisation in order to handle the big amount of generated alternatives. Moreover, when using computer aided methods, the role of 3D visualisations of proposed architectures needs to be further investigated: during the workshop, the experts spent 2 hours discussing the performances of architectures in order to determine which architectures were the best. At the end of the workshop, when visualising the architectures that they had selected, less than 2 seconds were needed to let them say that the selected architecture were not good. This complies with Okudan's conclusions, when she emphasised the need for concept selection methods that more form-oriented.

5.8 Conclusions and further work

In this paper, we highlighted the difficulty to identify the right selection criteria when it comes to complex system architectures selection. Complex system architecture impact many stages of the whole system life-cycle, which makes identification of selection criteria difficult:

- objectives are conflicting and sometimes interdependent;
- architecture attributes are all related;
- crucial information, such as cost, is missing and such performances may not be assessable.

As result, the experts can be lost. They do not know which criteria to take into account, which criteria should be treated in priority and how to deal with multiple criteria. Selection process is therefore becoming more and more confused as it progresses. In the end, the experts do not know if the selected architectures are better that the removed ones. Theirs reactions when visualising the chosen architectures in 3D suggest that it was not the case.

Because the solution is only as good as the criteria by which it is selected, a methodology to support the identification of criteria is needed. Indeed, no method to support the choice of criteria have been noted in the field of engineering design, despite the existence of many concept selection methods based on evaluation criteria. Interrelations between attributes and objectives will always exist when discussing system architecture to be selected. For this reason, existing methodologies in MCDM and concept selection need to be tailored to designers' needs and be more interactive. Further work is now required to research more detailed questions, such as:

- What should be considered when selecting system architectures to be further developed?
- How to handle interdependent criteria?
- How to deal with inherent uncertainty when selecting system architectures?
- Which selection methods are the most appropriate for a high number of alternatives?

5.9 References

Agarwal, M., Cagan, J. and Constantine, K. G. (1999). "*Influencing generative design through continuous evaluation: Associating costs with the coffeemaker shape grammar*", AI EDAM, **13**(4): 253-275.

Antonsson, E. and Cagan, J. (2001), "*Formal engineering design synthesis*". Cambridge University Press: UK.

Bashir, H. A. and Thomson, V. (1999). "*Estimating design complexity*", Journal of Engineering Design, **10**(3): 247-257.

Belton, V. and Stewart, T. J. (2002), "*Multiple criteria decision analysis: An integrated approach*". Springer.ISBN 079237505X.

Berliner, C. and Brimson, J. A. (1988), "*Cost management for today's advanced manufacturing: The cam-i conceptual design*". Harvard Business School Pr.ISBN 087584197X.

Biltgen, P. T. and Mavris, D. N. (2007), *"A technique for interactive probabilistic multiple attribute decision making"*.

Bouyssou, D. (1990). *"Building criteria: A prerequisite for mcda"*. In *Readings in multiple criteria decision aid.* Springer, p. 58-80.

Brans, J.-P., Vincke, P. and Mareschal, B. (1986). "*How to select and how to rank projects: The promethee method*", European Journal of Operational Research, **24**(2): 228-238.

Buonanno, M. A. and Mavris, D. N. (2004). "*Aerospace vehicle concept selection using parallel, variable fidelity genetic algorithms*", AIAA Paper, **4432** 2004.

Campbell, M. I., Cagan, J. and Kotovsky, K. (1999). "*A-design: An agent-based approach to conceptual design in a dynamic environment*", Research in Engineering Design, **11**(3): 172-192.

Chakrabarti, A., Shea, K., Stone, R. B., Cagan, J., Campbell, M., Wood, K. L. and Hernandez, N. V. (2011). "*Computer-based design synthesis research: An overview*", Journal of Computing and Information Science in Engineering, **11** 021003-021001.

Chapman, C. B. and Pinfold, M. (1999). "*Design engineering: A need to rethink the solution using knowledge based engineering*", Knowledge-Based Systems, **12**(5): 257-267.

Checkland, P. (1999). "*Systems thinking, systems practice: Includes a 30-year retrospective*".

Corner, J., Buchanan, J. and Henig, M. (2001). "*Dynamic decision problem structuring*", Journal of Multi-Criteria Decision Analysis, **10**(3): 129-141.

Crawley, E., Weck De, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D. and Whitney, D. (2004), "*The influence of architecture in engineering systems*".

Eden, C. and Ackermann, F. (2001). "*Strategic options development and analysis: The principles*".

Figueira, J. E., Greco, S. and Ehrgott, M. (2005), "*Multiple criteria decision analysis: State of the art surveys*". Springer, vol. 78.ISBN 038723067X.

Friend, J. and Hickling, A. (2005), "*Planning under pressure*". Routledge.ISBN 0750663731.

Gephi. (2008). http://gephi.org

Girod, M., Elliott, A., Burns, N. D. and Wright, I. (2003). "*Decision making in conceptual engineering design: An empirical investigation*", Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, **217**(9): 1215-1228.

Guindon, R. (1990). "*Designing the design process: Exploiting opportunistic thoughts*", Human-Computer Interaction, **5**(2): 305-344.

Henig, M. I. and Buchanan, J. T. (1996). "*Solving mcdm problems: Process concepts*", Journal of Multi-Criteria Decision Analysis, **5**(1): 3-21.

Keeney, R. L. and Gregory, R. S. (2005). "*Selecting attributes to measure the achievement of objectives*", Operations Research, **53**(1): 1-11.

Kihlander, I. (2011). *"Managing concept decision making in product development practice"*. KTH.

Martinsuo, M. and Poskela, J. (2011). "*Use of evaluation criteria and innovation performance in the front end of innovation*", Journal of Product Innovation Management, **28**(6): 896-914.

Mattson, C. A. and Messac, A. (2003). "*Concept selection using s-pareto frontiers*", AIAA journal, **41**(6): 1190-1198.

Mingers, J. and Rosenhead, J. (2004). "*Problem structuring methods in action*", European Journal of Operational Research, **152**(3): 530-554.

Montibeller, G., Franco, L. A., Lord, E. and Iglesias, A. (2009). "*Structuring resource allocation decisions: A framework for building multi-criteria portfolio models with area-grouped options*", European Journal of Operational Research, **199**(3): 846-856.

Moullec, M. L., Bouissou, M., Jankovic, M., Bocquet, J. C., Requillard, F., Maas, O. and Forgeot, O. (2013). "*Towards system architecture generation and performances assessment under uncertainty using bayesian networks* ", Journal of Mechanical Design, **135**(4): 13.

Moullec, M. L., Jankovic, M., Bouissou, M. and Bocquet, J. C. (2013). *"Proposition of combined approach for architecture generation integrating component placement optimization"*. In *Proceedings of the IDETC/CIE 2013* Portland, OR/USA, 10.

Okudan, G. E. and Tauhid, S. (2009). "*Concept selection methods a literature review from 1980 to 2008*", International Journal of Design Engineering, **1**(3): 243-277.

Olausson, D. and Berggren, C. (2010). "*Managing uncertain, complex product development in high tech firms: In search of controlled flexibility*", R&d Management, **40**(4): 383-399.

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), "*Engineering design: A systematic approach*". Springer-Verlag: London, vol. 157, p. 617.ISBN 1846283191.

Pugh, S. (1991), "*Total design: Integrated methods for successful product engineering*". Addison-Wesley Wokingham.ISBN 0201416395.

Raiffa, H. and Keeney, R. (1976). "*Decisions with multiple objectives: Preferences and value tradeoffs*", Decisions with multiple objectives: Preferences and Value Tradeoffs.

Rosenstein, D. and Reich, Y. (2011), *"Hierarchical concept generation by sos"*. Copenhagen, Denmark.

Roy, B. (1991). "*The outranking approach and the foundations of electre methods*", Theory and decision, **31**(1): 49-73.

Roy, B. and Bouyssou, D. (1991), "*Decision-aid: An elementary introduction with emphasis on multiple criteria*". Université de Paris Dauphine-Laboratoire d'analyse et modéisation de systèmes pour l'aide à la décision.

Saari, D. G. and Sieberg, K. K. (2004). "*Are partwise comparisons reliable?*", Research in Engineering Design, **15**(1): 62-71.

Saaty, T. L. (2008). "*Decision making with the analytic hierarchy process*", International Journal of Services Sciences, **1**(1): 83-98.

Scaravetti, D. (2004). *"Formalisation préalable d'un problème de conceptio, pour l'aide à la décision en conception préliminaire"*. Thèse de doctorat, Ecole Nationale Supérieure d'Arts et Métiers.

Schmidt, J. B., Sarangee, K. R. and Montoya, M. M. (2009). "*Exploring new product development project review practices**", Journal of Product Innovation Management, **26**(5): 520-535.

Scott, M. (2007). "*Quantifying uncertainty in multicriteria concept selection methods*", Research in Engineering Design, **17**(4): 175-187.

Simon, H. A. (1956). "*Rational choice and the structure of the environment*", Psychological review, **63**(2): 129.

Sonal. (2009). http://www.sonal-info.com/

Summers, J. D. and Shah, J. J. (2010). "*Mechanical engineering design complexity metrics: Size, coupling, and solvability*", Journal of Mechanical Design, **132** 021004.

Terharr, S., Clausing, D. and Eppinger, S. (1993). "*Integration of quality function deployment and the design structure matrix*", Cambridge, Mass.: Massachusetts Institute of Technology.

Ullman, D. G. (2001). "*Robust decision-making for engineering design*", Journal of Engineering Design, **12**(1): 3-13.

Ullman, D. G. (2002), *"The ideal engineering decision support system"*.

Vico, F. J., Veredas, F. J., Bravo, J. M. and Almaraz, J. (1999). "*Automatic design synthesis with artificial intelligence techniques*", Artificial Intelligence in Engineering, **13**(3): 251-256.

Wyatt, D., Wynn, D., Jarrett, J. and Clarkson, P. (2012). "*Supporting product architecture design using computational design synthesis with network structure constraints*", Research in Engineering Design, **23**(1): 17-52.

Yan, W., Chen, C.-H. and Shieh, M.-D. (2006). "*Product concept generation and selection using sorting technique and fuzzy< i> c</i>-means algorithm*", Computers & Industrial Engineering, **50**(3): 273-285.

Yang, J.-B. and Sen, P. (1997). "*Multiple attribute design evaluation of complex engineering products using the evidential reasoning approach*", Journal of Engeering Design, **8**(3): 211-230.

Yeo, S., Mak, M. and Balon, S. (2004). "*Analysis of decision-making methodologies for desirability score of conceptual design*", Journal of Engineering Design, **15**(2): 195-208.

Ziv-Av, A. and Reich, Y. (2005). "*Sos – subjective objective system for generating optimal product concepts*", Design Studies, **26**(5): 509-533.

5.10 Appendix

The overall set of criteria discussed in the workshop is presented in Table 5–5.

Table 5–5. List of selection criteria

Criterion	Designation	Definition
Number of elements	Number of elements Element Size Interconnections Decomposition	It is the number of components that constitute a solution. It determines the size of each element (Section **Erreur ! Source du renvoi introuvable.**) and drives the number of interconnections in the architecture.
Cost	Cost Cost/grouping surface Cost/Element size	Here cost refers to the cost of the technology that is used in the solution, which includes the electronic items but also the fabrication process.
Manufacturing	Manufacturing Feasibility Production efficiency Fabrication efficiency Efficiency	This addresses the questions linked to the manufacturability of the product. Depending on the size of the components and the technology, manufacturing efficiency consider profitability regarding investments for development and production means.
Technology	Technology	This is electronic technology that performs a top level function.
Function	Function	Top level function of the building block.
“Globalité”	Function Sharing Mix of technologies Integration Clustering	Function Sharing, mix of technologies and integration refers to the functional aspect of “globalité”. Clustering corresponds to the number of components performing the same function.
Complexity	Complexity	It is the internal complexity of a component which represents the difficulty of fabrication. It depends on the number of functions performed by the component and on the number of RF channels covered by the component.
Reliability	Reliability Redundancy	In the context of this workshop, considering the reliability means examining the number of components, how they are related, as well as their redundancy in order to ensure an acceptable mean time between failures (MTBF).
Mass	Mass	It includes all the components modelled in the architecture.
Temperature	Temperature	This is the air temperature required to avoid any deterioration of electronic components.
Pressure drop	Pressure drop	It is the expected loss of pressure produced by the system solution. It generally impacts the cooling system design at the antenna scale.
Global depth	Global depth	The building block has a fixed length and height. The solutions vary only on the depth. Otherwise, the whole volume of the solution would have been considered.
Diversity of solutions	Diversity of solutions	This criterion does not apply to a single solution, but to the set of solutions that are selected. Indeed, the experts would like to retain solutions that are diverse in terms of technology or/and performances.
Architecture families	Architecture configuration Solutions clusters Similarity of solutions	The particular technology used in a solution has a strong impact on its physical layout and falls into particular patterns that are called “architecture families” by the experts. This concept is especially linked to the diversity of solutions, but this time it applies to a single solution.
Performance	Performance boundaries	

	Performance thresholds	This relates to the thresholds imposed by the experts for a given performance. In the same way, the range of possible values for the whole solutions or only a subset falls also into this category.
	performances relaxation	
Data quality	Data quality	This takes into account the quality of the data available for selection. The estimation of some performances may be of lower-quality than others. The assumptions and the bias induced by the model must also be considered.
	Knowledge	
	Uncertainty	
	Architecture consistency	

6
Conclusions and perspectives

The present chapter outlines contributions, limits and perspectives of propositions and industrial implementations included in this thesis. The topic of this research study is complex system architecture design when considering integration of innovative solutions. The impacts resulting from the choice of system architecture make system architecture design process decisive. However, current system architecture frameworks provide tools useful for the description of system architectures, but not adapted to their design. The latter requires dedicated tools (Yannou, 2001) in particular for architecture generation and selection.

This PhD thesis has been conducted in collaboration with Thales Air Systems in view to propose a concrete help to the early design stages of complex systems. More particularly, the objective was to provide a method integrating innovation and supporting the definition of system architecture. Based on industrial problems and state of the art, two research questions have been defined:

1. How to integrate innovation in early design stages and to support design space exploration in complex system architectures design?
2. How to compare and select complex system architectures when objectives are still ill-defined and information on system architecture alternatives is incomplete, uncertain and fuzzy?

A response to the first research question is proposed with a method automatically generating system architectures and estimating their performances. This method, using Bayesian networks, relies on designer knowledge and integrates the uncertainties related to component characterisation, component compatibilities and performance estimation (Chapter 3). A further step, based on CSP, aims at optimising component placement of generated architectures (Chapter 4). Finally, in order to answer the second research question, criteria for system architectures selection are investigated through the initiation and the analysis of an architecture selection workshop that took place in industry. It provided insights and recommendations for future system architecture selection method. This process and the related contributions are illustrated in Figure 6-1.

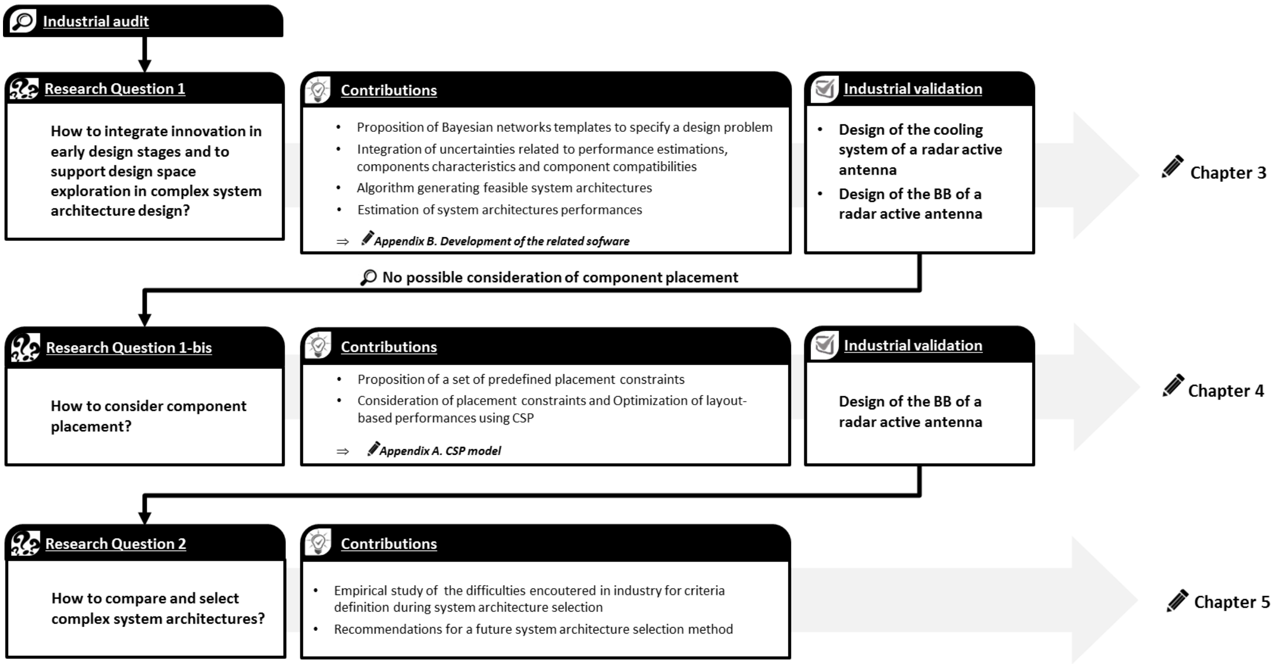


Figure 6-1. Contributions overview

The next paragraphs present contributions, limits and perspectives respectively related to the method provided to generate system architecture, the study of criteria for system architecture selection and the industrial implementations.

6.1 A method to generate feasible complex system architectures

In view to address the first research question, we propose a method using Bayesian networks combined with Constraint Satisfaction Problems in order to generate feasible system architectures that integrate innovative technologies. This generation is based on the information chosen by designers. Thus, only important design choices as well as main performances are considered: the information is formalised as decision variables, constraints and performance calculations within a Bayesian network in order to generate system architectures, whose components placement is optimised using CSP.

6.1.1 Contributions

The state of the art presented in Chapter 3 and Chapter 4 underlined lacks in the current design synthesis methods. In particular, most methods cannot handle complex system problems because they require exhaustive elicitation of design variables and rules, which is very difficult and fastidious for complex systems. Anyhow, the resulting design space becomes too large to be entirely explored. In addition, such methods use design repositories which lack information about data used in the synthesis. This is all the more the case when innovative solutions must be considered. As a result, system architecture evaluation and selection are impossible. Otherwise, the methods that are based on designers' knowledge sometimes propose system architecture evaluation, but this evaluation is either qualitative, or based on metrics that do not consider satisfaction of system requirements.

The method that we propose addresses each of these problems by also attempting to give a pragmatic and practical answer. In this method, designers play a prominent role in the generation of system architectures: they choose which decision variables are important and have to be modelled; they also decide which performances are evaluated and how this evaluation is made. Defining the solution space in this way permits an exhaustive exploration, by focusing only on variables which have been judged interesting by designers. The generic character of Bayesian networks offers a flexible modelling framework by not restricting designers to model a specific type of decision variables which may be a technology, a function, a component or even a placement configuration. Moreover, Bayesian networks allow taking into account several types of uncertainty: uncertainty on compatibilities between variables, imprecision in the data and uncertainty in the evaluation models. This is particularly useful to integrate innovations when information is lacking or not sure. The resulting confidence level can be then considered as any other performance of generated architectures and architecture selection can be made in better knowledge of risk. This is all the more important in early design stages. Besides this, this method estimates architecture performances, which makes possible the selection of architectures regarding system requirements, including those related to component placement. As for component placement, layout optimisation provides simplified 3D visualisations. This feature has been particularly appreciated by designers, because it makes the architecture solutions more meaningful and easier to apprehend. In addition, each architecture solution is provided with working value ranges for every characteristic of decision variables, which may constitute initial design specifications for the chosen architecture. Finally, designers can decide to change or add one hypothesis, one decision variable, one constraint or one performance in their own way. The time required for generating the solutions on the two industrial cases varied between 10 seconds and 3 minutes, which seems adapted to work in an industrial context. This makes this method a practical tool to progressively explore the solutions space while focusing on the most interesting solutions.

6.1.2 Limits

The implementation of Thales use cases allowed us identifying a number of weaknesses that need to be addressed in the future.

The first ones concern the modelling approach. First, defining decision variables that hold on technical solutions that might be very different but that must be described by the same characteristics may be arduous. A second weakness relies on the expression of uncertainty about compatibilities between components: this uncertainty reflects the opinion of designer; and several designers may disagree on the level of this uncertainty. The approach, as it is proposed, requires that designers reach agreement on the level of uncertainty. Other limitations are mainly technical. A limitation affects the size of design problem. In cases where relations between variables are particularly intricate, troubles due to Bayesian inference may occur: complexity of inference in a Bayesian network essentially depends on the size of conditional probability tables and on the number and configuration of cycles in the network structure (a cycle corresponds to the existence of two different paths starting from a given node and arriving at another node). Therefore it is possible to have very large networks (with thousands of nodes) easy to process, and much smaller ones (say, less than 100 nodes) which are very hard. For the latter, a lot of RAM may be required. In particular, this limitation has been a handicap to optimise component placement. Instead, we finally chose to use CSP for component placement optimisation. However, CSP manipulate crisp values: dimensions of components are given as intervals by the Bayesian networks, but only one value (the mid-value of the most probable interval) is input in the CSP. The ability to deal with uncertainty is therefore lost for layout-based performances. Bayesian network modelling had been chosen because it was a graphical approach; this was estimated to ease the problem representation. But modelling a complex system creates a complex graph and does not permit an instantaneous understanding of the design problem. Additional visualisation features are therefore needed to ease the graph understanding. A second difficulty in the use of Bayesian networks is the consideration of continuous variables which need to be discretised. This discretisation is currently made manually. However, a wrong discretisation, in particular the specification of minimum and maximum values, may lead to wrong estimations of performances and thus to wrong results (cf. Section 3.10). At this time, this limitation is circumvented by informing designers of the problem.

6.1.3 Research perspectives

Most of the limits that we identified are inherent to the use of Bayesian networks. In the short term, the problem of discretisation of continuous variables could be solved by proposing an automatic discretisation feature that would be based on min-max and probability distribution analyses. Further information on discretisation techniques is available in (Catlett 1991; Liu, Hussain et al. 2002; Kurgan and Cios 2004). In the longer term, problems related to the size of Bayesian network could be avoided by developing an object oriented approach. There are several modelling frameworks that can be considered as an extension of Bayesian networks. In particular, object oriented probabilistic relational models (OOPRM) (Torti, Wuillemin, et al. 2010) may suit our application: OOPRM could solve problems of network size by allowing calculation and navigation in optimally determined sub-models (Torti 2012). The use of OOPRM would both simplify the construction of models and ensure the scalability of the approach. However, no tool adapted to industrial context is currently available and OOPRM still need further development.

This research study also underlines new perspectives addressing the improvement of the method. New templates may be proposed in particular to address the integration of technical solutions that do not share the same characteristics. The consideration of divergent designer opinions concerning the uncertainty should also be addressed: Ullman (2006) proposed a Bayesian approach aiming at aggregating opinions from different people. This may constitute a first answer. Another improvement of the method consists in providing tools for refinement of the design problem by analysis of the corresponding Bayesian network. For example, it would be useful to provide indicators that identify the constraints that strongly reduce solution space in order to enable trade-offs analysis and constraint relaxation. Likewise, identifying which decision variables have a strong impact on a specified performance could enhance the understanding of the design problem, and allow focusing design efforts on specific points of the design problem. Also, enrichment of the Bayesian network model is still possible by integrating other performances, including system safety and reliability that are often studied using Bayesian networks. Finally, in order to perfectly suit designers' needs, CSP modelling needs more development like proposition of additional placement constraints, management of shapes other than parallelepipeds as well as new resolution algorithms.

6.2 Investigation of the importance of criteria definition in the selection process

Following the implementation of the two case studies, a one-day workshop was organised in Thales in order to select the best architectures amongst those generated using the method. The objective was to observe the selection process adopted by designers as well as to investigate the usefulness of Prométhée which is a multicriteria decision method. At the end of the workshop, 10 architectures were selected regarding their performances. However, the designers were not fully convinced with choices that they had made and were unable to compare and rank these architectures. An in-depth analysis showed their difficulties in defining the right selection criteria due to lack of information and dependencies between criteria.

6.2.1 Contributions

Our main contribution related to architecture selection question is to have highlighted the non-intuitive character of the definition of selection criteria. Indeed, most of concept selection methods (which could be used for system architecture selection) assume that criteria are already defined by designers or give recommendations as to what to consider when selecting a concept. However, system complexity, lack of information and the interdependencies between criteria make them difficult to be applied to system architecture selection. As a result, the selection criteria were ill-defined, ever-changing and designers were neither satisfied nor confident with regard to chosen architectures. This workshop mainly showed that designers still need a method supporting architecture selection. Indeed, current multicriteria decision methods are not suited to system architecture selection given that their postulates – exhaustiveness of criteria family, monotonicity and non-redundancy of criteria (Bouyssou 1990) – cannot be satisfied. The second contribution of this work is that it gave some insights on the features that a system architecture selection method should include. In particular, it should manage the uncertainty inherent to the definition phase of system architecture, as well as the interdependencies between criteria. Current selection methods do not manage these two issues. With regard to these needs, some potential solutions have been suggested and require further reflection and development.

6.2.2 Limits

Our analysis relies on observations coming from a single workshop and cannot be generalised. Moreover, this workshop included a number of biases that must be kept in mind when interpreting the results.

First of all, this workshop was an exercise where designers' responsibility was not engaged. It is not certain that the experts would have been so inclined to avoid conflicts if their own responsibilities had come in line. Perhaps they would have insisted more on imposing their own point of view, leading to more debates. Moreover, time and industrial constraints led us to define a limited number of criteria and a limited number of decision-makers, which may have affected the choice of architecture solution. The little time spent on this workshop likely induced some biased behaviours such as hasty choices of selection criteria in order to quickly sort the architectures and save time. But, the fact that not all experts are attending such a meeting and/or that the meeting has a limited duration is a situation usually encountered in industry. This should be taken in consideration when developing future architecture selection methods. Finally, because the candidate architectures were generated with an automatic tool, the number of potential architectures was very high, which is not usually the case in current industrial practises. Nevertheless, the trend is to find optimised concepts rather than "satisficing" concepts (Simon 1956). Likewise, the resort to automated/intelligent methods tends to increase and design methodologies should evolve accordingly.

6.2.3 Research perspectives

Pursuing this work should encompass several steps necessary to propose an adequate and generic architecture selection method. First of all, similar workshops in other industrial contexts should be organised in order to identify common practise and recurring difficulties. In addition, the effects of the biases addressed in the previous sections should be analysed in order to measure the impacts of each of them. The role of visualisation features also needs to be investigated: during the workshop, 3D visualisations seemed to carry much more information than descriptive spreadsheets or diagrams. However, they have been used only at the end of the selection process as a confirmation that the right choices were made. Introduction and structuring of adequate data media (3D visualisation or others) within system architecture selection method is one important action in view to propose a new concept selection method.

More generally, this work opened up new questions specific to system architecture selection issue. Indeed one cannot anticipate and consider all the impacts of potential system architecture. Thus, what should be considered at this design stage in view to select realistic architectures? Likewise, how to alleviate the recurring problem of lack of information, in particular for cost evaluation? Moreover, system architecture as a result of its interdisciplinarity induces interdependent selection criteria. We propose to manage them by setting up a multistep selection process in which a way of articulating criteria needs to be defined. Finally, the use of automated methods results in a high number of solutions. This last point opens the way towards induction methods exploiting experts' opinions for architecture evaluation; as well as selection methods more adapted to high number of alternatives.

6.3 Industrial implementations

This thesis was conducted in Thales and implemented on two case studies. In view of a future use in the company, software has been developed to ease design problem modelling as well as visualisation of solutions.

6.3.1 Contributions

The two implementations address current problems encountered in Thales, but only on very specific cases. However, the method that we proposed allows modifications in the design problem model. The models corresponding to the building-block and the cooling system can therefore be adapted to new projects by changing system requirements or adding new decision variables and/or constraints.

Concerning the specific use of this method, it has been shown that it resulted in a better and quicker exploration of design space (Chapters 3 and 4). Moreover, the fact that the constraints and performances of all domains are entered within a same model enhances common problem understanding from designers, which may result in the reduction of conflicts and iterations during system architectures development, and thus improve concurrent design. Finally, system architectures are all evaluated according to the same performances, which objectifies the choice of a specific solution.

Software developed during this thesis allows an easy modelling of design problem and frees designers from the need to specifically know Bayesian networks and CSP. It also provides 3D visualisation of system architectures solutions. Moreover, "library" features permits knowledge capitalisation on data characterising technical solutions, on performance estimation models as well as on justification of specific architecture choices.

6.3.2 Limits

The information on which the two industrial implementations are built was provided by Thales engineers. However, some hypotheses have been made when information was not available. Therefore, these pieces of information need refinement and require confirmation in order to ensure the validity of performance estimations. Moreover, the industrial cases were modelled in close collaboration with Thales designers, but they do not use the approach by themselves. This way of modelling a design problem is very different from what designers usually do, which results in the fact that this method requires a learning phase. Finally, the software developed to ease the design problem modelling only addresses the Bayesian network construction. The CSP part should be in turn treated in the same manner in order to develop the whole potential of the approach. In addition, software is at a very early stage and requires further tests and improvements.

6.3.3 Development perspectives

From Thales point of view, the method has entered in a test phase, where some engineers are using it. If the method contributions are judged sufficient, it will be implemented and used for real projects. In such a case, CSP development will become necessary and software will need professional developments.

Finally, using this method in the design process may result in changes in Thales design process. The time saved by the automation of the solution space exploration will be available for the development of better surrogate models and more efforts on data gathering.

Personal Publications

Papers in bold are those selected to constitute chapters of this dissertation.

Journal papers

- **Moullec, M. L., Bouissou, M., Jankovic, M., Bocquet, J. C., Requillard, F., Maas, O. and Forgeot, O. (2013). "*Towards system architecture generation and performances assessment under uncertainty using Bayesian networks* ", Journal of Mechanical Design, 135(4): 13.**
- **Moullec, M. L., Jankovic, M. and Eckert, C. (2013). "*Investigating the importance of criteria selection in system architecture: Observations from industrial experiment*", submitted to Research in Engineering Design.**

Conference papers

- **Moullec, M. L., Jankovic, M., Bouissou, M. and Bocquet, J. C. (2013). *"Proposition of combined approach for architecture generation integrating component placement optimisation"*. In *Proceedings of the IDETC/CIE 2013.* Portland, OR/USA, 10.**
- Moullec, M. L., Bouissou, M., Jankovic, M. and Bocquet, J. C. (2012), *"Use of a bayesian nets-based method to automatically generate complex product architectures: Application on the cooling system of a radar active antenna"*. Paris, France: Springer-Verlag.
- Moullec, M. L., Jankovic, M., Bouissou, M. and Bocquet, J. C. (2012). *"Product architectures generation under uncertainty: Comparison between two methods"*. In *Proceedings of the IDETC/CIE 2012*. Chicago, IL/USA.
- Moullec, M. L., Bouissou, M., Jankovic, M. and Bocquet, J. C. (2012). *"Product architecture generation and exploration using bayesian networks"*. In *Proceedings of the 12th International Design Conference DESIGN 2012*. Dubrovnik, Croatia, 1761-1770.

Bibliography

Agarwal, M., Cagan, J. and Constantine, K. G. (1999). "*Influencing generative design through continuous evaluation: Associating costs with the coffeemaker shape grammar*", AI EDAM, **13**(4): 253-275.

Albarello, N. and Welcomme, J.-B. (2012), *"A formal design synthesis and optimization method for systems architectures"*. Bordeaux, France.

Albers, A., Braun, A., Sadowski, E., Wynn, D., Wyatt, D. and Clarkson, J. (2011). "*System architecture modeling in a software tool based on a contact and channel approach (c&c-a)*", J Mech Design, **133**(10): 8.

Antonsson, E. and Cagan, J. (2001), "*Formal engineering design synthesis*". Cambridge University Press: UK.

Aughenbaugh, J. M. and Paredis, C. (2006), *"Why are intervals and imprecision important in engineering design?"*. Savannah, GA.

Ayag, Z. (2005). "*An integrated approach to evaluating conceptual design alternatives in a new product development environment*", International Journal of Production Research, **43**(4): 687-713.

Ayyub, B. M. (2010), "*Elicitation of expert opinions for uncertainty and risks*". CRC press.ISBN 0849310873.

Barton, D. N., Saloranta, T., Moe, S. J., Eggestad, H. O. and Kuikka, S. (2008). "*Bayesian belief networks as a meta-modelling tool in integrated river basin management — pros and cons in evaluating nutrient abatement decisions under uncertainty in a norwegian river basin*", Ecological Economics, **66**(1): 91-104.

Bashir, H. A. and Thomson, V. (1999). "*Estimating design complexity*", Journal of Engineering Design, **10**(3): 247-257.

Bellut, S. (1990), "*La compétitivité par la maîtrise des coûts. Conception à coût objectif et analyse de la valeur*".

Belton, V. and Stewart, T. J. (2002), "*Multiple criteria decision analysis: An integrated approach*". Springer.ISBN 079237505X.

Bénabès, J. (2011). *"Optimisation intégrée et interactive pour l'agencement d'espace"*. Thèse de doctorat, Ecole Centrale de Nantes.

Berliner, C. and Brimson, J. A. (1988), "*Cost management for today's advanced manufacturing: The cam-i conceptual design*". Harvard Business School Pr.ISBN 087584197X.

Biltgen, P. T. and Mavris, D. N. (2007), *"A technique for interactive probabilistic multiple attribute decision making"*.

Bouyssou, D. (1990). *"Building criteria: A prerequisite for mcda"*. In *Readings in multiple criteria decision aid.* Springer, p. 58-80.

Brans, J.-P., Vincke, P. and Mareschal, B. (1986). "*How to select and how to rank projects: The promethee method*", European Journal of Operational Research, **24**(2): 228-238.

Bryant, C., Mcadams, D. A. and Stone, R. B. (2005). *"A computational technique for concept generation"*. In *Proceedings of the ASME Internationa Design Engineering Technical Conference IDETC 2005*. Long Beach, USA.

Buonanno, M. A. and Mavris, D. N. (2004). "*Aerospace vehicle concept selection using parallel, variable fidelity genetic algorithms*", AIAA Paper, **4432** 2004.

Cagan, J., Campbell, M. I., Finger, S. and Tomiyama, T. (2005). "*A framework for computational design synthesis: Model and applications*", J Comput Inf Sci Eng, **5**(3): 171-181.

Cagan, J., Shimada, K. and Yin, S. (2002). "*A survey of computational approaches to three-dimensional layout problems*", Computer-Aided Design, **34**(8): 597-611.

Campbell, M. I., Cagan, J. and Kotovsky, K. (1999). "*A-design: An agent-based approach to conceptual design in a dynamic environment*", Research in Engineering Design, **11**(3): 172-192.

Catlett, J. (1991), *"On changing continuous attributes into ordered discrete attributes"*. Springer, p. 164-178.

Chakrabarti, A. (2002), "*Understanding, approaches and tools*". Springer: London, UK.

Chakrabarti, A. and Bligh, T. P. (1996). "*An approach to functional synthesis of mechanical design concepts: Theory, applications, and emerging research issues*", AI EDAM, **10**(04): 313-331.

Chakrabarti, A., Bligh, T. P. and Holden, T. (1992). "*Towards a decision-support framework for the embodiment phase of mechanical design*", Artificial Intelligence in Engineering, **7**(1): 21-36.

Chakrabarti, A., Shea, K., Stone, R. B., Cagan, J., Campbell, M., Wood, K. L. and Hernandez, N. V. (2011). "*Computer-based design synthesis research: An overview*", Journal of Computing and Information Science in Engineering, **11** 021003-021001.

Chandrasekaran, B. (1990). "*Design problem solving: A task analysis*", AI Magazine, **11**(4): 13.

Chapman, C. B. and Pinfold, M. (1999). "*Design engineering: A need to rethink the solution using knowledge based engineering*", Knowledge-Based Systems, **12**(5): 257-267.

Checkland, P. (1999). "*Systems thinking, systems practice: Includes a 30-year retrospective*".

Chenouard, R., (2007), *"Resolution par satisfaction de contraintes appliquee à l'aide à la décision en conception architecturale"*, Bordeaux.

Choco 2. (2013). www.emn.fr/z-info/choco-solver/

Cooper, R. G. (1994). "*Third-generation new product processes*", Journal of Product Innovation Management, **11**(1): 3-14.

Corner, J., Buchanan, J. and Henig, M. (2001). "*Dynamic decision problem structuring*", Journal of Multi-Criteria Decision Analysis, **10**(3): 129-141.

Crawley, E., Weck De, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D. and Whitney, D. (2004), "*The influence of architecture in engineering systems*".

Cross, N. (2000), "*Engineering design methods: Strategies for product design*". Wiley Chichester, vol. 58.

Davies, A. and Hobday, M. (2005), "*The business of projects: Managing innovation in complex products and systems*". Cambridge University Press.ISBN 0521843286.

De Weck, O., Eckert, C. and Clarkson, J. (2007), "*A classification of uncertainty for early product and system design*". vol. 7.

Dieter, E. (2000), "*Engineering design: A materials and processing approach*". McGraw-Hill: Singapore.

Eden, C. and Ackermann, F. (2001). "*Strategic options development and analysis: The principles*".

Figueira, J. E., Greco, S. and Ehrgott, M. (2005), "*Multiple criteria decision analysis: State of the art surveys*". Springer, vol. 78.ISBN 038723067X.

Fixson, S. K. (2005). "*Product architecture assessment: A tool to link product, process, and supply chain design decisions*", Journal of Operations Management, **23**(3): 345-369.

Gephi. (2008). http://gephi.org

Gero, J. S. and Kannengiesser, U. (2004). "*The situated function-behaviour-structure framework*", Design Studies, **25**(4): 373-391.

Girod, M., Elliott, A., Burns, N. D. and Wright, I. (2003). "*Decision making in conceptual engineering design: An empirical investigation*", Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, **217**(9): 1215-1228.

Goldberg, D. E. and Holland, J. H. (1988). "*Genetic algorithms and machine learning*", Machine learning, **3**(2): 95-99.

Guindon, R. (1990). "*Designing the design process: Exploiting opportunistic thoughts*", Human-Computer Interaction, **5**(2): 305-344.

Gupta, S. and Okudan, G. E. (2008). "*Computer-aided generation of modularised conceptual designs with assembly and variety considerations*", Journal of Engineering Design, **19**(6): 533-551.

Haykin, S. (1994), "*Neural networks: A comprehensive foundation*". Prentice Hall PTR.ISBN 0023527617.

Helms, B. and Shea, K. (2012). "*Computational synthesis of product architectures based on object-oriented graph grammars*", J Mech Design, **134**(2).

Henig, M. I. and Buchanan, J. T. (1996). "*Solving mcdm problems: Process concepts*", Journal of Multi-Criteria Decision Analysis, **5**(1): 3-21.

Hsu, W. and Liu, B. (2000). "*Conceptual design: Issues and challenges*", Computer Aided Design, **32**(14): 849-850.

Hubka, V. and Eder, W. E. (2001). *"Design science"*.

Ilog (2006). http://www-01.ibm.com/software/commerce/optimization/cplex-cp-optimizer/

Jankovic, M. (2013). *"Inverstigating the impact of system architecture onto overall system design"*. Habilitation à diriger les recherches, Ecole Centrale Paris.

Jensen, F. V. and Nielsen, T. D. (2007), "*Bayesian networks and decision graphs*". Springer: New-York.

Keeney, R. L. and Gregory, R. S. (2005). "*Selecting attributes to measure the achievement of objectives*", Operations Research, **53**(1): 1-11.

Kerzhner, A. and Paredis, C. (2009), *"Using domain specific languages to capture design synthesis knowledge for model-based systems engineering - detc2009-87286"*. San Diego, California.

Kihlander, I. (2011). *"Managing concept decision making in product development practice"*. KTH.

Kreimeyer, M. F. (2009). *"A structural measurement system for engineering design processes"*. Technical University of Munich.

Krob, D. (2009). "*Eléments d'architecture des systèmes complexes*", Gestion de la Complexité et de l'information Dans les Grands Systèmes Critiques, 179-207.

Kumar, V. (1992). "*Algorithms for constraint-satisfaction problems: A survey*", AI Magazine, **13**(1): 32.

Kurgan, L. A. and Cios, K. J. (2004). "*Caim discretization algorithm*", Knowledge and Data Engineering, IEEE Transactions on, **16**(2): 145-153.

Kurtoglu, T. and Campbell, M. I. (2009). "*Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping*", Journal of Engineering Design, **20**(1): 83-104.

Levis, A. H. (2011). *"Systems architecture"*. In *Handbook of systems engineering and management*. Wiley. com.

Lindemann, U., Maurer, M. and Braun, T. (2008), "*Structural complexity management: An approach for the field of product design*". Springer-Verlag: Berlin

Liu, H., Hussain, F., Tan, C. L. and Dash, M. (2002). "*Discretization: An enabling technique*", Data mining and knowledge discovery, **6**(4): 393-423.

Maier, M. W. and Rechtin, E. (2000), "*The art of systems architecting*". CRC press.ISBN 0849304407.

Martinsuo, M. and Poskela, J. (2011). "*Use of evaluation criteria and innovation performance in the front end of innovation*", Journal of Product Innovation Management, **28**(6): 896-914.

Matthews, P. C. (2011). "*Challenges to bayesian decision support using morphological matrices for design: Empirical evidence*", Research in Engineering Design, **22**(1): 14.

Mattson, C. A. and Messac, A. (2003). "*Concept selection using s-pareto frontiers*", AIAA journal, **41**(6): 1190-1198.

Maurer, M. S. (2007), "*Structural awareness in complex product design*". Verlag Dr. Hut.ISBN 3899636325.

Medjoub, B. (1996). *"Méthode de conception fonctionnelle en architecture : Une approche basée sur les contraintes: Archiplan"*. Thèse de doctorat, Ecole Centrale Paris.

Meyer, J. (2000). "*Qualitative research in health care: Using qualitative methods in health related action research*", BMJ: British Medical Journal, **320**(7228): 178.

Mingers, J. and Rosenhead, J. (2004). "*Problem structuring methods in action*", European Journal of Operational Research, **152**(3): 530-554.

Montibeller, G., Franco, L. A., Lord, E. and Iglesias, A. (2009). "*Structuring resource allocation decisions: A framework for building multi-criteria portfolio models with area-grouped options*", European Journal of Operational Research, **199**(3): 846-856.

Moullec, M. L., Bouissou, M., Jankovic, M. and Bocquet, J. C. (2012), *"Use of a bayesian nets-based method to automatically generate complex product architectures: Application on the cooling system of a radar active antenna"*. Paris, France: Springer-Verlag.

Moullec, M. L., Bouissou, M., Jankovic, M., Bocquet, J. C., Requillard, F., Maas, O. and Forgeot, O. (2013). "*Towards system architecture generation and performances assessment under uncertainty using bayesian networks* ", Journal of Mechanical Design, **135**(4): 13.

Moullec, M. L., Jankovic, M., Bouissou, M. and Bocquet, J. C. (2013). *"Proposition of combined approach for architecture generation integrating component placement optimization"*. In *Proceedings of the IDETC/CIE 2013* Portland, OR/USA, 10.

Nordqvist, T., Edberg, M. A., Bystrom, M. H. and Gustafsson, M. M.-L. (2006). "*Systems engineering, architecture frameworks and modelling & simulation*".

O'leary, C., Rao, S. and Perry, C. (2004). "*Improving customer relationship management through database/internet marketing: A theory-building action research project*", European Journal of Marketing, **38**(3/4): 338-354.

Okudan, G. E. and Tauhid, S. (2009). "*Concept selection methods a literature review from 1980 to 2008*", International Journal of Design Engineering, **1**(3): 243-277.

Olausson, D. and Berggren, C. (2010). "*Managing uncertain, complex product development in high tech firms: In search of controlled flexibility*", R&d Management, **40**(4): 383-399.

Openscad. (2010). http://www.openscad.org/

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), "*Engineering design: A systematic approach*". Springer-Verlag: London, vol. 157, p. 617.ISBN 1846283191.

Pidd, M. (2003), "*Tools for thinking*". Wiley Chichester.ISBN 0470847956.

Piller, F. T., Moeslein, K. and Stotko, C. M. (2004). "*Does mass customization pay? An economic approach to evaluate customer integration*", Production planning & control, **15**(4): 435-444.

Pugh, S. (1991), "*Total design: Integrated methods for successful product engineering*". Addison-Wesley Wokingham.ISBN 0201416395.

Raiffa, H. and Keeney, R. (1976). "*Decisions with multiple objectives: Preferences and value tradeoffs*", Decisions with multiple objectives: Preferences and Value Tradeoffs.

Roger, J. (1999), "*Antennes - différents types*".

Rosenstein, D. and Reich, Y. (2011), "*Hierarchical concept generation by sos*". Copenhagen, Denmark.

Roy, B. (1991). "*The outranking approach and the foundations of electre methods*", Theory and decision, **31**(1): 49-73.

Roy, B. and Bouyssou, D. (1991), "*Decision-aid: An elementary introduction with emphasis on multiple criteria*". Université de Paris Dauphine-Laboratoire d'analyse et modéisation de systèmes pour l'aide à la décision.

Saari, D. G. and Sieberg, K. K. (2004). "*Are partwise comparisons reliable?*", Research in Engineering Design, **15**(1): 62-71.

Saaty, T. L. (2008). "*Decision making with the analytic hierarchy process*", International Journal of Services Sciences, **1**(1): 83-98.

Scaravetti, D. (2004). "*Formalisation préalable d'un problème de conceptio, pour l'aide à la décision en conception préliminaire*". Ecole Nationale Supérieure d'Arts et Métiers.

Schmidt, J. B., Sarangee, K. R. and Montoya, M. M. (2009). "*Exploring new product development project review practices**", Journal of Product Innovation Management, **26**(5): 520-535.

Scott, M. (2007). "*Quantifying uncertainty in multicriteria concept selection methods*", Research in Engineering Design, **17**(4): 175-187.

Simon, H. A. (1956). "*Rational choice and the structure of the environment*", Psychological review, **63**(2): 129.

Sonal. (2009). http://www.sonal-info.com/

Stone, R. B. and Wood, K. L. (2000). "*Development of a functional basis for design*", J Mech Design, **122**(4): 359-370.

Summers, J. D. and Shah, J. J. (2010). "*Mechanical engineering design complexity metrics: Size, coupling, and solvability*", Journal of Mechanical Design, **132** 021004.

Terharr, S., Clausing, D. and Eppinger, S. (1993). "*Integration of quality function deployment and the design structure matrix*", Cambridge, Mass.: Massachusetts Institute of Technology.

Torti, L. (2012). *"Structured probabilistic inference in object-oriented bayesian networks"*. Paris 6.

Torti, L., Wuillemin, P.-H. and Gonzales, C. (2010), *"Reinforcing the object-oriented aspect of probabilistic relational models"*. p. 273-280.

Tsang, E. (1993), "*Foundations of constraint satisfaction*". Academic press London, vol. 289.

Ullman, D. G. (2001). "*Robust decision-making for engineering design*", Journal of Engineering Design, **12**(1): 3-13.

Ullman, D. G. (2002), *"The ideal engineering decision support system"*.

Ullman, D. G. (2006), "*Making robust decisions: Decision management for technical, business and service teams*". Trafford Publishing.ISBN 142510956X.

Ulrich, K. T. and Eppinger, S. D. (1995), "*Product design and development*". Irwin McGraw-Hill: New York.

Vico, F. J., Veredas, F. J., Bravo, J. M. and Almaraz, J. (1999). "*Automatic design synthesis with artificial intelligence techniques*", Artificial Intelligence in Engineering, **13**(3): 251-256.

Whelton, M., Ballard, G. and Tommelein, I. D. (2002a). "*A knowledge management framework for project definition*", ITcon, **7**(ICT for Knowledge Management in Construction): 197-212.

Whelton, M., Ballard, G. and Tommelein, I. D. (2002b). "*A knowledge management framework for project definition*", ITcon, **7** 197-212.

Whitney, D. E. (2004), "*Mechanical assemblies: Their design, manufacture, and role in product development*". Oxford university press New York.ISBN 0195157826.

Wyatt, D., Wynn, D., Jarrett, J. and Clarkson, P. (2012). "*Supporting product architecture design using computational design synthesis with network structure constraints*", Research in Engineering Design, **23**(1): 17-52.

Yan, W., Chen, C.-H. and Shieh, M.-D. (2006). "*Product concept generation and selection using sorting technique and fuzzy< i> c</i>-means algorithm*", Computers & Industrial Engineering, **50**(3): 273-285.

Yang, J.-B. and Sen, P. (1997). "*Multiple attribute design evaluation of complex engineering products using the evidential reasoning approach*", Journal of Engeering Design, **8**(3): 211-230.

Yannou, B. (2001). "*Préconception de produits*". Habilitation à diriger les recherches, Ecole Centrale Paris.

Yannou, B. and Petiot, J.-F. O. (2011). "*A view of design (and jmd): The french perspective*", Journal of Mechanical Design, **133**(5): 050301-050301.

Yeo, S., Mak, M. and Balon, S. (2004). "*Analysis of decision-making methodologies for desirability score of conceptual design*", Journal of Engineering Design, **15**(2): 195-208.

Yvars, P.-A. (2008). "*Using constraint satisfaction for designing mechanical systems*", International Journal of Interactive Design and Manufacturing, **2**(3): 161-167.

Zablit, P. and Zimmer, L. (2001), "*Global aircraft predesign based on constraint propagation and interval analysis*". Köln, Germany.

Zadeh, L. A. (1988). "*Fuzzy logic*", Computer, **21**(4): 83-93.

Ziv-Av, A. and Reich, Y. (2005). "*Sos – subjective objective system for generating optimal product concepts*", Design Studies, **26**(5): 509-533.

Appendices

A

Component placement optimisation

This chapter presents the object-oriented CSP approach used to optimise components placement of system architectures. The different classes of objects are defined as well as the constraints easing the modelling of the placement problem.

A. 1 Constraint Satisfaction Problem

A. 1.1 Definition

A constraint satisfaction problem (CSP) is characterised by a set of variables, each defined on a finite domain, and that must satisfy a number of constraints. Formally, a CSP is defined by a triple $\{X, D, C\}$ where:

- $X = \{x_1, x_2, \ldots, x_n\}$ are decision variables;
- $D = \{d_1, d_2, \ldots, d_n\}$ are the domains of decision variables such that x_i takes its values in d_i;
- $C = \{c_1, c_2, \ldots, c_k\}$ are a finite set of constraints.

A CSP can be discrete or continuous. If CSP is discrete, d_i is a list of all possible assignments for x_i and CSP resolution is based on listing and filtering. For CSPs with real variables on continuous domains, constraints are propagated over intervals.

A. 1.2 Resolution algorithm

There exists numerous ways of solving a CSP (Kumar 1992). The simplest algorithms is "generate-and-test" algorithm in which each possible combination of the variables is systematically generated and then tested to see if it satisfies all the constraints. In backtracking algorithm, variables are instantiated sequentially. As soon as all the variables relevant to a constraint are instantiated, the validity of the constraint is checked. If a partial instantiation violates any of the constraints, backtracking is performed to the most recently instantiated variable that still has alternatives available. This algorithm has been improved using dynamic approach and some heuristics, like arc-consistency, are in general applied to reduce processing time.

A. 2 Classes used in the object-oriented CSP

In order to optimise the component placement of the system architectures generated by the Bayesian networks, we chose to address this problem using an object-oriented approach. This consists in defining several classes of objects and defining placement constraints as methods. For the modelling, we used the CSP solver *Choco 2*.

A. 2.1 The class "Space"

This object represents the maximum volume that the assembly of architecture components can occupy. It is described by four attributes:

- Name: the name of the space;
- Dimensions: the dimensions of the volume expressed in the tuple {length, width, depth}.

In this way, the space represents the 3D referential S in which components can be placed using x, y and z coordinates.

A. 2.2 The class "Function"

Each function that the architecture must accomplish is defined with the following attributes:

- Name : the name of the function;
- Technology : the technology used in the function;
- Number: the number of identical components that accomplish the function;
- Dimensions: the width, the height and the depth of the component;
- Colour: the colour expressed in a tuple $\{R, G, B\}$ used for 3D visualisation.

All these attributes are known and constitute inputs for the placement problem.

A. 2.3 The class "Component"

A component accomplishes a specific function of the architecture. This is the entity that must be placed. Its attributes are:

- Function: the function that the component accomplishes;
- Index: the rank of the component in the whole set of components;
- Functional Index: the rank of the component in the set of components accomplishing the same function;
- Orientation: the orientation of the component;
- Dimensions: the width w, the height h, and the depth d which correspond to the functional dimensions but expressed in the referential S. As we need to share a common orientation

referential, all components are at first set in a same reference orientation defined such as: $w^0 \leq h^0 \leq d^0$;

- Coordinates: x, y and z that define the position of the component within the space.

A. 3 Definition of the CSP

A. 3.1 Inputs

Each solution to be optimised is generated by a Bayesian network using a list of functions described by their 5 attributes (name, technology, number of components, dimensions and colour). They constitute the inputs for a given architecture so that all the instances of components to be placed can be automatically generated.

A. 3.2 Decision variables

The attributes constituting the set of decision variables X of the CSP are the orientation and the coordinates of every component. For each component c, a domain D is defined following these rules:

- coordinate x: $D_x^c = [0, space_{width} - \min(w, h, d)]$
- coordinate y: $D_y^c = [0, space_{height} - \min(w, h, d)]$
- coordinate z: $D_z^c = [0, space_{depth} - \min(w, h, d)]$
- orientation:
  - if c has 3 different dimensions: $D_{or}^c = \{0,1,2,3,4,5\}$
  - if c has width and height identical: $D_{or}^c = \{1,4,6\}$
  - if c has width and depth identical: $D_{or}^c = \{0,1,2\}$
  - if c has depth and height identical: $D_{or}^c = \{2,3,6\}$
  - if c has 3 identical dimensions: $D_{or}^c = \{0\}$

with each number corresponding to a specific orientation (Figure A- 1).

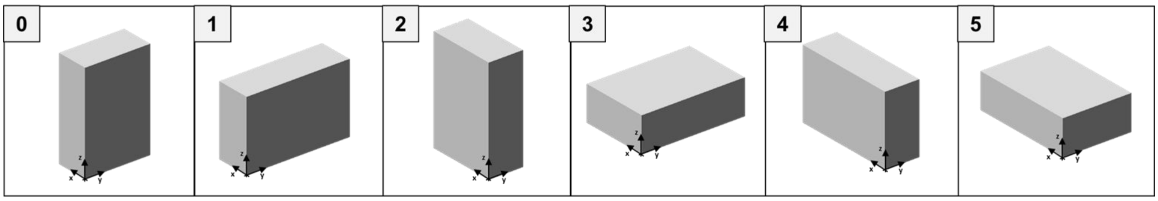


Figure A- 1. Possible orientations of a component

A. 3.3 Constraints

The second step of CSP definition consists in elicitation of all placement constraints to be satisfied. This is the most difficult step given that one must consider all cases, notably in terms of technology, so that any architecture solution can be optimised. In this section, we propose a set of predefined constraints in order to ease this modelling. However, it is not intended to be exhaustive.

A.3.3.1. Implicit constraints

First of all, some implicit constraints shared by any placement problem must be satisfied so that components do not overlap. These constraints are the following:

$$y_i^2 \le y_i^1 + h_i^1 \ \textbf{and}\ y_i^2 \ge y_i^1 - h_i^1 \ \textbf{and}\ z_i^2 \le z_i^1 + d_i^1 \ \textbf{and}\ z_i^2 \ge z_i^1 - d_i^1 \ \Rightarrow x_i^2 \le x_i^1 - w_i^1 \ \textbf{or}\ \ x_i^2 \ge x_i^1 - w_i^1$$

$$x_i^2 \le x_i^1 + w_i^1 \ \textbf{and}\ x_i^2 \ge x_i^1 - w_i^1 \ \textbf{and}\ z_i^2 \le z_i^1 + d_i^1 \ \textbf{and}\ z_i^2 \ge z_i^1 - d_i^1 \ \Rightarrow y_i^2 \le y_i^1 - h_i^1 \ \textbf{or}\ \ y_i^2 \ge y_i^1 - h_i^1$$

$$x_i^2 \le x_i^1 + w_i^1 \ \textbf{and}\ x_i^2 \ge x_i^1 - w_i^1 \ \textbf{and}\ y_i^2 \le y_i^1 + h_i^1 \ \textbf{and}\ y_i^2 \ge y_i^1 - h_i^1 \ \Rightarrow z_i^2 \le z_i^1 - d_i^1 \ \textbf{or}\ \ z_i^2 \ge z_i^1 - d_i^1$$

Also, other constraints are used to define the dimensions of each component i according to its orientation O_i and its dimensions in reference orientation $w_i^0, h_i^0\ and\ z_i^0$:

$$O_i = 0 \Rightarrow w_i = w_i^0;\ h_i = h_i^0; d_i = d_i^0$$

$$O_i = 1 \Rightarrow w_i = w_i^0;\ h_i = d_i^0; d_i = h_i^0$$

$$O_i = 2 \Rightarrow w_i = h_i^0;\ h_i = w_i^0; d_i = d_i^0$$

$$O_i = 3 \Rightarrow w_i = h_i^0;\ h_i = d_i^0; d_i = w_i^0$$

$$O_i = 4 \Rightarrow w_i = d_i^0;\ h_i = w_i^0; d_i = h_i^0$$

$$O_i = 5 \Rightarrow w_i = d_i^0;\ h_i = h_i^0; d_i = w_i^0$$

A.3.3.2. Position constraint

The constraints presented in Table A- 1 force the component to be adjacent to one of the sides of the assembly.

Table A- 1. Position constraints

On left	$x = 0$
On right	$x + w = \max\limits_{0 \le i \le n}(x_i + w_i)$
On bottom	$y = 0$
On top	$y + h = \max\limits_{0 \le i \le n}(y_i + h_i)$
On front	$z = 0$
On back	$z + d = \max\limits_{0 \le i \le n}(z_i + d_i)$

A.3.3.3. Orientation constraint

This constraint forces a component to be in the specific orientation O chosen amongst the 6 possible orientations defined in the previous section:

$$c_{or} = O$$

A.3.3.4. Adjacency constraints

The set of constraints illustrated in Figure A- 2 force the two components C1 and C2 to be adjacent according to a specific side (assuming that the dimensions of C2 in the orthogonal axes are lower or equal to the dimensions of C1).

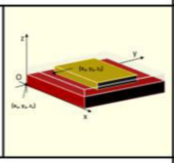
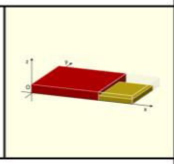
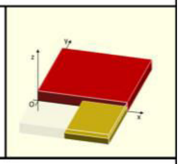
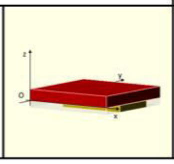
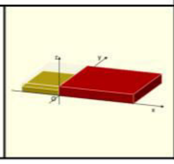
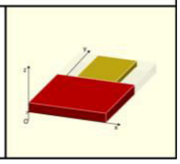
C2 on the top of C1		C2 on the right of C1		C2 on the front of C1	
$x_2 \geq x_1$ $x_2 + w_2 \leq x_1 + w_1$ $y_2 \geq y_1$ $y_2 + h_2 \leq y_1 + h_1$ $z_1 + d_1 = z_2$		$y_2 \geq y_1$ $y_2 + h_2 \leq y_1 + h_1$ $z_2 \geq z_1$ $z_2 + d_2 \leq z_1 + d_1$ $x_1 + w_1 = x_2$		$z_2 \geq z_1$ $z_2 + d_2 \leq z_1 + d_1$ $x_2 \geq x_1$ $x_2 + w_2 \leq x_1 + w_1$ $y_1 + h_1 = y_2$	
C2 on the bottom of C1		C2 on the left of C1		C2 on the back of C1	
$x_2 \geq x_1$ $x_2 + w_2 \leq x_1 + w_1$ $y_2 \geq y_1$ $y_2 + h_2 \leq y_1 + h_1$ $z_2 + d_2 = z_1$		$y_2 \geq y_1$ $y_2 + h_2 \leq y_1 + h_1$ $z_2 \geq z_1$ $z_2 + d_2 \leq z_1 + d_1$ $x_2 + w_2 = x_1$		$z_2 \geq z_1$ $z_2 + d_2 \leq z_1 + d_1$ $x_2 \geq x_1$ $x_2 + w_2 \leq x_1 + w_1$ $y_2 + h_2 = y_1$	

Figure A- 2. Adjacency constraints

A.3.3.5. Distance constraints

This constraint forces two components c_1 and c_2 to be at least D-distant according to a specific axis (Ox, Oy or Oz):

- $Ox: |x_1 - x_2| \geq D$
- $O_y: |y_1 - y_2| \geq D$
- $O_z: |z_1 - z_2| \geq D$

A. 4 Solver

Several solvers (http://openjvm.jvmhost.net/CPSolvers) are available and propose numerous resolution techniques and heuristics to express and resolve a CSP that can be discrete, continuous or mixed. In this study, we have used the solver *Choco 2* in order to optimise the placement of components.

With *Choco 2*, defining a resolution algorithm mainly consists in choosing or defining a branching strategy. For this study, we opted for the default branching strategy. It uses the standard approach in constraint programming that develops the enumeration tree in a Depth First Search manner (Kumar 1992). This solver can be used to 1) enumerate all or a given number of possible solutions or 2) find an optimised solution.

The purpose of this resolution is to find one placement optimised in terms of depth. However, there may be several optimised placements that it could be interesting to compare. To this aim, we propose a resolution in 2 steps:

1. Launch the solver in the configuration 2 in order to find the optimised value for the depth;

2. Introduce the constraint so that the depth is equal to the optimised value found in the precedent step. Launch again the solver in the configuration 1 in order to find a given number of solutions, five for example.

In this way, five optimised placements are proposed for one BN solution input in the CSP. In order to improve the diversity of proposed placements, the solver was set to iterate on variables and values in a random way during the search. However, this leads to a slight deterioration of resolution time.

A. 5 Solutions

A. 5.1 Description

A solution is presented under the form of a list of components, each characterised by the function accomplished, the set of chosen coordinates $\{x, y, z\}$ as well as a tuple of dimensions $\{w, h, d\}$ that are representative of the chosen orientations. For instance, if we consider an architecture of 4 components of which the two first ones have the same function, the solution is defined as in Table A- 2.

Table A- 2. Type of solutions given by the CSP solver

Component	Function	x	y	z	w	h	d
C1	F1	0	5	10	40	40	20
C2	F1	60	5	10	40	40	20
C3	F2	40	15	0	20	20	20
C4	F3	0	0	0	100	50	10

A. 5.2 Visualisation

The solutions generated by the CSP in their "table" form are difficult to apprehend. In order to aid the comprehension of such solutions, we developed a code that allows visualising them in the 3D viewer *OpenSCAD* (Openscad 2010). The solution presented in Table A- 2 is graphically represented in Figure A- 3.

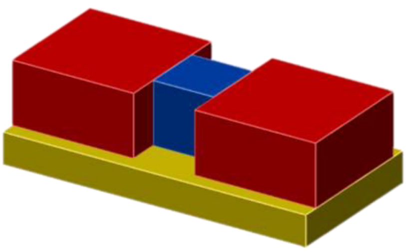


Figure A- 3. Architecture visualisation using OpenSCAD

B

Software

Not all engineers have knowledge in Bayesian networks and CSP. However, our method for system architecture generation is expected to be used by all types of engineers. For this reason, it requires to be supported by dedicated software which allows modelling design problem within a Bayesian Network and the associated CSP as well as visualising the generated architectures. Two interfaces have been developed: an interface to define the Bayesian network and an interface to visualise solution. The interface to define CSP still requires development.

B. 1 Software for problem modelling

This software has been developed with the help of a computer science student in view to provide an interface easing the definition of a design problem. This interface is designed using Java and Netica APIs. The main function aims at defining:

- decision variables and theirs characteristics;
- constraints with uncertainty or not;
- performances;

Decision variables as well as performance calculations can be recorded in two libraries in order to be reused in other models. The design problem that is being modelled is represented with a graph. The overall interface is shown in Figure B- 1.

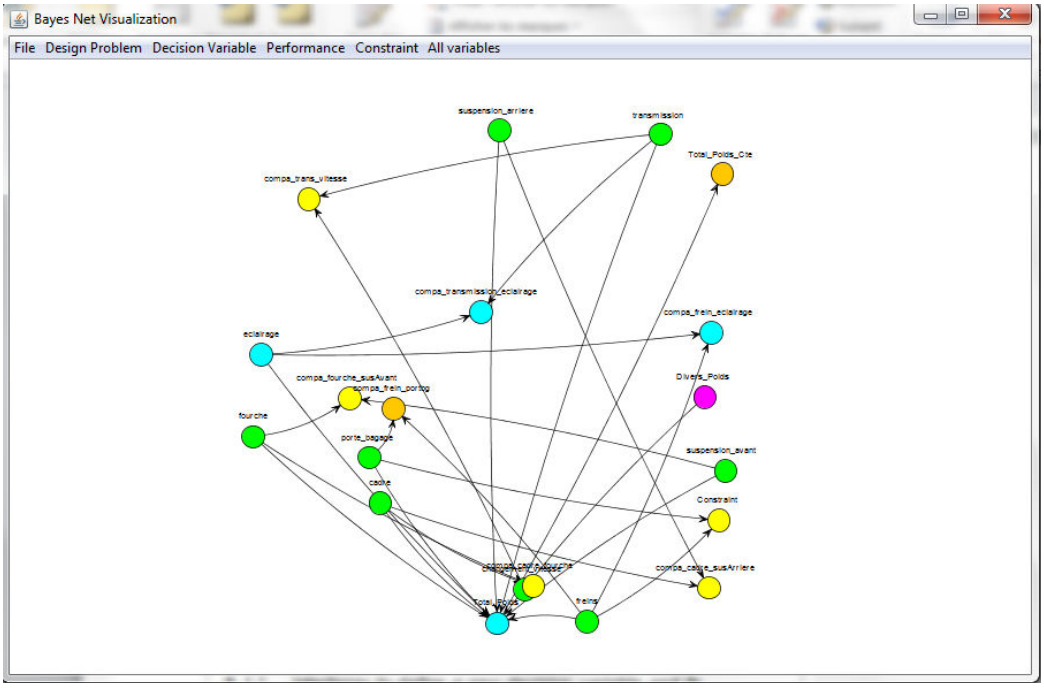


Figure B- 1. Global interface for problem modelling

B. 1.1 Interfaces to define a new decision variable and its characteristics

B.1.1.1. New decision variable

The definition and modification of decision variable is accessible through the menu "Decision variable > new". The user is asked to define the name, the type and the instances of the decision variable (Figure B- 2). There is also the possibility to add some comments. The probabilities are automatically defined as uniform.

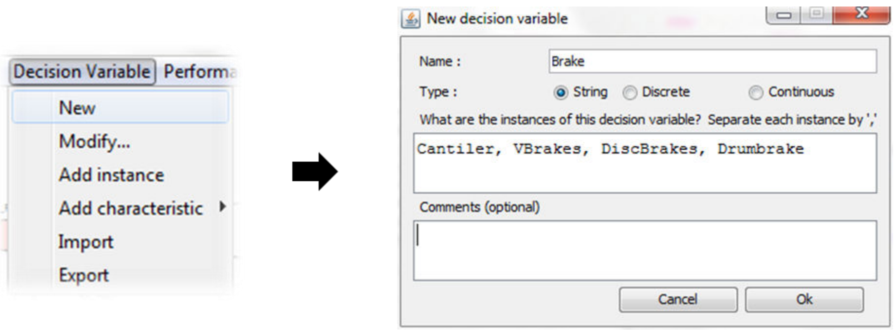


Figure B- 2. Creation of a new decision variable

B.1.1.2. Definition of decision variable characteristic

Following the creation of the decision variable, it is possible to define its characteristics. This function is also accessible through the menu "Decision variable > Add characteristic". The user has the possibility to define the characteristic using an equation or by directly entering the value of the characteristic for each instance of the decision variable, as shown in Figure B- 3.

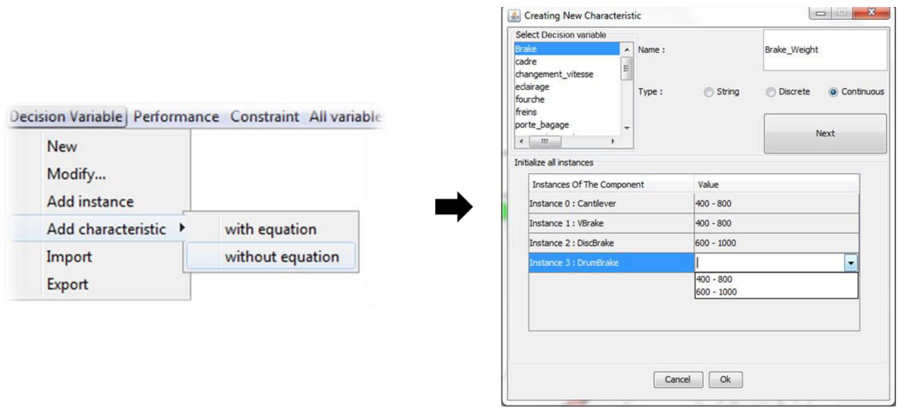


Figure B- 3. Modelling of a new characteristic

B. 1.2 Interface to define a new constraint

Adding a new constraint is accessible using the menu "Constraint > Add…". The user can choose to insert a constraint with or without uncertainty. In cases when the constraint is uncertain, the user defines it by providing only the percentage of chance that the constraint has to be true (Figure B- 4). It can also be defined using an equation.

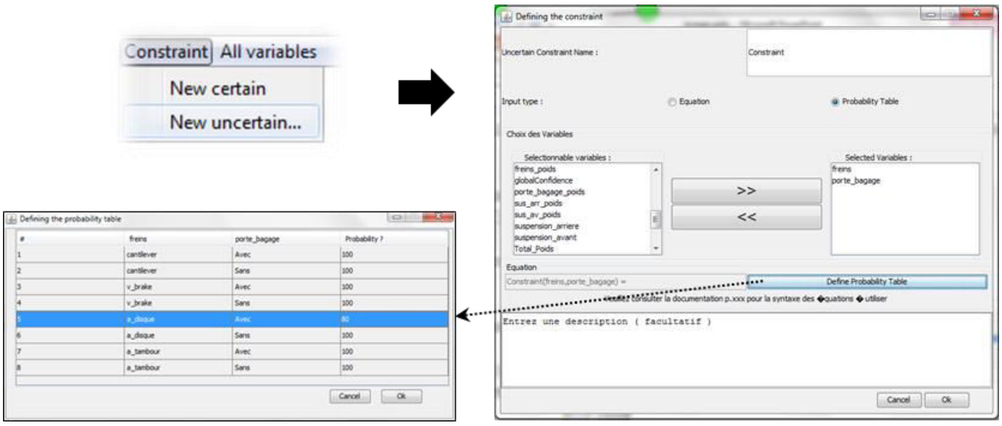


Figure B- 4. Creation of an uncertain constraint

B. 1.3 Interface to define a new performance

A new performance can be defined using the menu "Performance > New". In order to prevent errors, the user chooses the arguments of the performance in the list of nodes. The names of chosen variables are then written automatically inside the text area allowing him to define the equation (Figure B- 5). He can also define probabilistic performances by using equations or by directly defining the probability table.

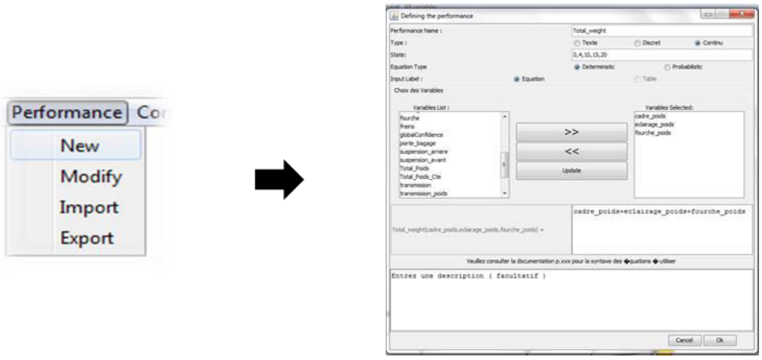


Figure B- 5. Creation of a new performance

B. 1.4 Design problem visualisation

While the user is building the design problem, a visualisation of the design problem is updated. This visualisation uses the API Jung to display a graph representing the design problem. The placement of nodes is automatically defined. By double clicking on one node, the user can see the probabilities of one variable. To simplify the graph, decision variables and theirs characteristics are displayed in one node (Figure B- 6). Different colors allows to differentiate the variables by type (green for decision variables, yellow for constraints, pink for hypothesis, etc…)

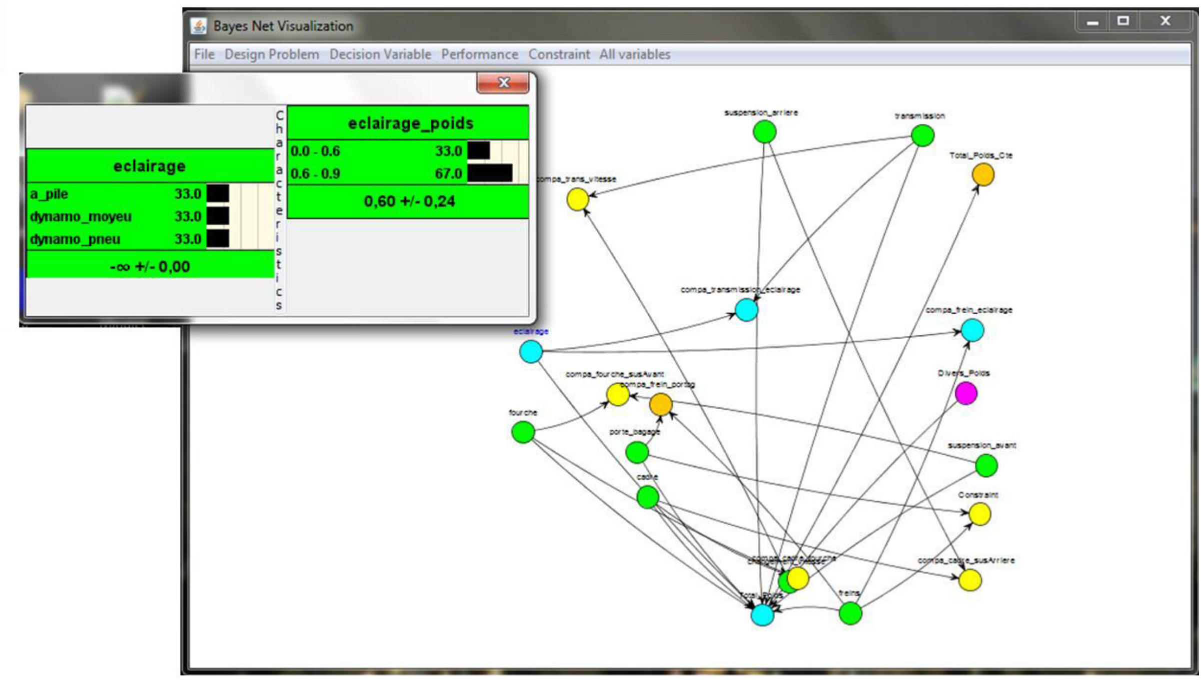


Figure B- 6. Design problem visualisation

B. 1.5 Other functions

Others functions allow importing and exporting decision variables and performances in dedicated libraries. It is also possible to modify, to copy or to delete some variables already created. But the main function is to generate the solutions and visualise the performances. Given that the possibility to define a CSP is foreseen but not yet implemented, a "provisional" interface has been developed for visualisation of solutions. It contains the list of solutions as well as display of the performances of the selected solution (Figure B- 7).

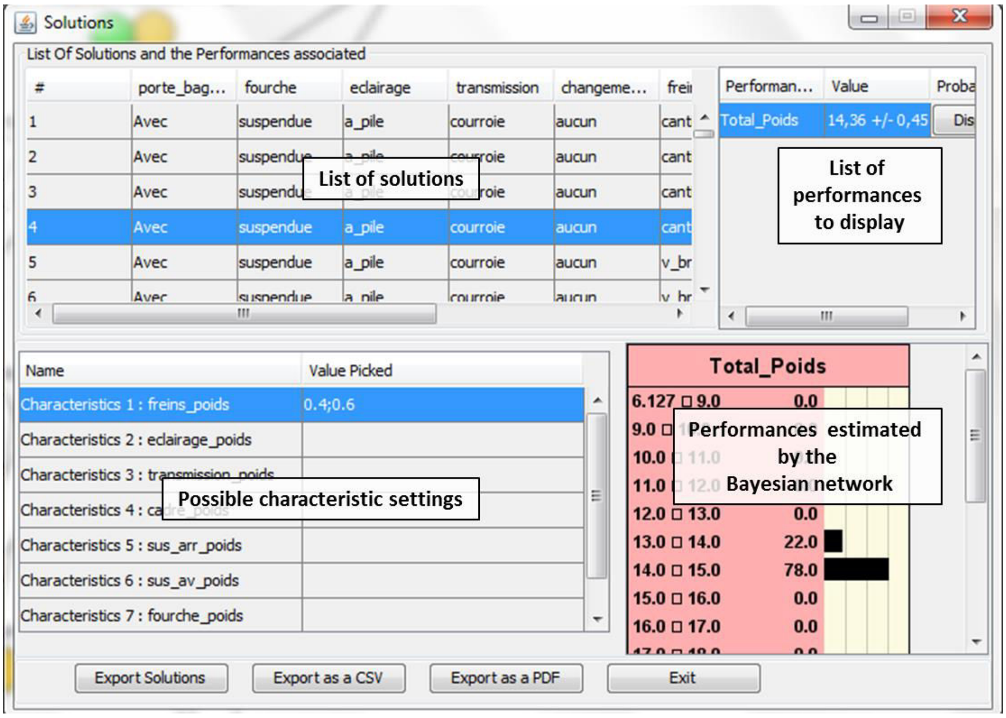


Figure B- 7. Visualisation of generated solutions

This interface is provided pending implementation of CSP part in the software. However, the final interface should closely look like the interface provided for the visualisation of architectures of the building block (our second industrial case). It is presented in the next section.

B. 2 Software for visualisation of generated solutions

This software has been specifically developed to visualise the 800 solutions of building blocks and is not integrated to software for problem modelling. But it provides a first insight into the type of interface that is required for visualisation of solutions (Figure B- 8).

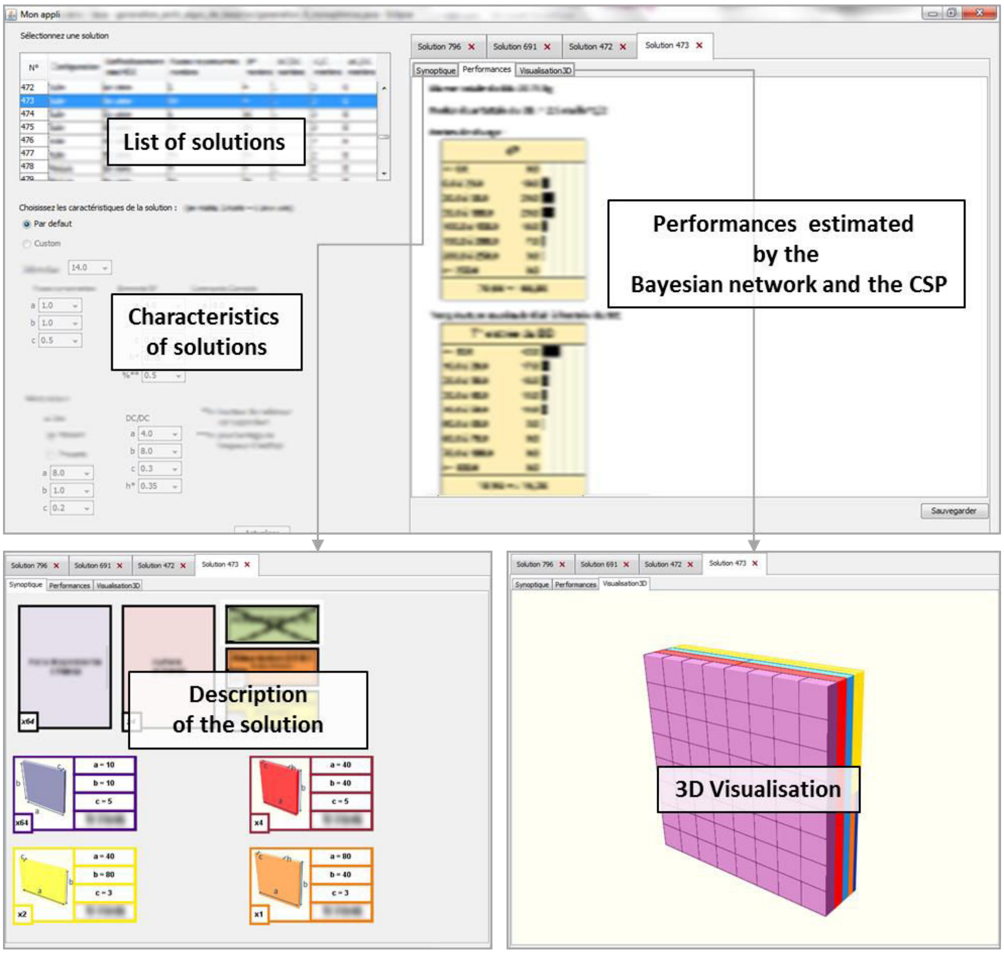


Figure B- 8. Interface for visualisation of solutions

Once an architecture is selected in the list of solutions, the panel displaying architecture characteristics is updated and 3D visualisation and performances of the solution are displayed in a new tab. In this way, the user can "play" with different values of characteristics (which are every characteristic of every decision variable) and see the impacts of solution performances. When estimated with the Bayesian network, the performances of the solution are displayed with probability distributions. When estimated with CSP, the performance consists of a single value. The 3D visualisation is displayed in a 3D-viewer in order to allow the user to see the architecture from different perspectives. Finally, the tab describing the selected solution uses scheme specific to building blocks and cannot be made generic.

This software gathers all information available about system architectures that have been generated with the aim of easing evaluation and selection stages of system architectures.