HAL
open science

TEMPAS - contribution à la qualité dans un système d'alertes contextualisées adaptable

Juan Pablo Suarez Coloma

To cite this version:

Juan Pablo Suarez Coloma. TEMPAS - contribution à la qualité dans un système d'alertes contextualisées adaptable. Systèmes et contrôle [cs.SY]. Université de Grenoble, 2014. Français. NNT : 2014GRENM076 . tel-01552629

HAL Id: tel-01552629

https://theses.hal.science/tel-01552629v1

Submitted on 3 Jul 2017

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

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

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Juan Pablo SUAREZ COLOMA

Thèse dirigée par **Christine VERDIER** et
codirigée par **Claudia RONCANCIO**

préparée au sein **de Calystene,**
du Laboratoire d'Informatique de Grenoble
dans **l'École Doctorale Mathématiques, sciences et technologies de l'information, informatique (MSTII)**

TEMPAS – contribution à la qualité dans un système d'alertes contextualisées adaptable

Thèse soutenue publiquement le **17 décembre 2014**,
devant le jury composé de :

M. Franck MORVAN
Professeur, Université Paul Sabatier, Rapporteur
M. Philippe ROOSE
MCF-HDR, Université de Pau et des pays de l'Adour, Rapporteur
Mme Corine CAUVET
Professeur, Aix Marseille Université, Présidente
M. Jean-Marc BABOUCHKINE
Président Directeur Général, Calystene, Examinateur
M. Régis BEUSCART
Professeur, Université de Lille 2, Examinateur
M. Francois PORTET
MCF, Grenoble INP, Examinateur
Mme Christine VERDIER
Professeur, Université Joseph Fourier, Directrice de thèse
Mme Claudia RONCANCIO
Professeur, Grenoble INP, Co-directrice de thèse

A ma famille. Ma mère, mon père, mes sœurs et mon frère
qui m'ont toujours soutenu

Remerciements

Je tiens à remercier tous ceux qui ont participé à la réalisation de ce projet de thèse. Je remercie Franck Morvan et Philippe Roose pour avoir rapporté mon travail et Corine Cauvet, Régis Beuscart et François Portet pour l'avoir examiné.

Christine et Claudia, merci beaucoup pour m'avoir accueilli dans l'équipe Sigma, et surtout pour m'avoir encadré pendant la réalisation de ce travail afin de franchir cette étape. Merci beaucoup pour votre accompagnement. Je souhaite remercier Jean Pierre Giraudin pour s'être investi dans la lecture de ce document et pour les conseils qu'il a su m'offrir. Je remercie aussi tous les autres membres de l'équipe Sigma, en particulier Mario C. et Aurélien F. avec qui j'ai partagé non seulement un bureau mais aussi une amitié.

Je remercie la société Calystene pour son accueil pendant le déroulement de cette thèse en plus de la financer. Je remercie tout particulièrement son PDG, Jean-Marc Babouchkine pour initialisé ce projet ainsi que pour m'avoir guidé et conseillé avec Jacques Dufour, grâce à leur expertise dans le domaine de la santé, pendant ces années de thèse. Je souhaite aussi remercier mes collègues de travail de Calystene et de Delta Consultants pour les moments vécus au sein de la société, ainsi que Wilmondes et Aurélien J. qui sont nés le même jour que moi et avec qui j'ai gardé une amitié.

Je tiens à remercier Nadine Mandran du Laboratoire d'Informatique de Grenoble et Jean Pierre Grangier de la société Calydial dont les discussions m'ont fait gagner de la confiance dans ce travail. Je remercie aussi les membres de Calydial qui ont testé le produit.

Je remercie Julian Felipe Rivera et Felipe Melo, mes amis colombiens avec qui je suis arrivé et resté en France et qui forment le noyau de ma famille colombienne en France. Ils m'ont vu commencer ce travail et me voient maintenant franchir cette étape si importante. Je n'oublie pas J.C., Stiven, Andrea, Julian V., Camilo D., Juan Manuel, Camilo M., Carito, Ivan, Daniel et tous les autres colombiens avec qui j'ai partagé des innombrables moments de joie à Grenoble facilitant le déroulement de ma thèse. Je remercie aussi à ce propos Sophie, Gosia, Marine et Stéphan les « étrangers » de ce groupe.

Je remercie de tout cœur ma famille qui m'a toujours soutenu inconditionnellement à la distance. Cette réussite ne serait pas possible sans eux et l'amour qu'ils me donnent, c'est pourquoi elle leur appartient autant qu'à moi.

Enfin, je remercie Eléonore qui a subi les dernières années d'une thèse et qui a su rester à mon côté faisant preuve de ses qualités comme compagne. Le temps partagé ensemble, sa compagnie, sa patiente, ses encouragements et son amour étaient essentiels pour que ce projet aboutisse. Je remercie aussi sa mère pour l'environnement familial récréé.

Résumé

Les systèmes d'alertes sont utilisés dans une grande variété de domaines d'applications. La croissance continue des besoins de monitoring accentue la nécessité de disposer de systèmes d'alertes adaptables permettant de contrôler un large éventail de variables.

Dans cette thèse, nous proposons un système d'alertes paramétrables nommé TEMPAS permettant de capturer des éléments sémantiques propres aux contextes d'utilisation des alertes ainsi qu'une personnalisation facile de la part des utilisateurs. Ceci est réalisé, entre autres, par l'utilisation de valeurs linguistiques et quantitatives dans la définition des alertes. En plus d'une maîtrise fine des aspects temporels, le modèle d'alertes de TEMPAS introduit des indices de qualité qui sont associés aux alertes. Ces indices de qualité permettent de mieux informer les utilisateurs sur la qualité des données utilisées pour reconnaître les situations d'alerte et sur l'applicabilité de l'alerte par rapport au profil de l'entité monitorée. Les indices de qualité permettent aussi de définir des stratégies pour adapter la notification au contexte des utilisateurs afin de réduire les faux-positifs et les faux-négatifs.

TEMPAS est adaptable à des besoins très variés en matière d'alertes. Le domaine privilégié dans cette thèse a été celui de la santé. Nous avons développé et validé nos propositions en coopération avec Calystene, société spécialisée dans des solutions d'informatique médicale.

Abstract

Alert systems are used in a wide variety of application areas. The continued growth of monitoring needs accentuates the need for adaptable alert systems to control a large range of variables.

In this thesis, we propose a configurable alert system called TEMPAS to capture specific semantic elements particular to alerts contexts of use as well as easy customization by users. This is achieved, inter alia, by the use of linguistic and numerical values in the definition of the alert situations. In addition to fine control of temporal aspects, the model of TEMPAS alerts introduce quality indices associated to alerts. These quality indicators are used to better inform users about the quality of data used to recognize the alert situations and the applicability of the alert concerning the monitored entity profile. Quality indices also help define strategies to adapt the notification to the context of users to reduce false-positive and false-negative.

TEMPAS is adaptable to a wide variety of needs for alerts. The privileged domain in this thesis was that of health. We have developed and validated our proposals in cooperation with Calystene, a company specializing in medical IT solutions.

Table de matières

Table de Figures *15*

Table de Tableaux *17*

1 Introduction *19*

1.1 Contexte général **19**

1.2 Motivations **20**

1.2.1 Contexte d'application des alertes 20
1.2.2 Dimensions de qualité 22

1.3 Proposition **24**

1.4 Domaine d'application privilégié **25**

1.5 Structure du document **26**

2 Les systèmes d'alertes *27*

2.1 Introduction **27**

2.2 Les systèmes d'alertes existants **27**

2.2.1 Personnes âgées 27
2.2.2 Notification de masse 28
2.2.3 Sécurité publique 28
2.2.4 Monitoring continu 28
2.2.5 Prévention 29
2.2.6 Partage de connaissances 30
2.2.7 Utilisateurs concepteurs et consommateurs 33
2.2.8 Réduction de faux-positifs 35

2.3 Catégorisation des systèmes d'alertes **39**

2.3.1 Définition de situations d'alerte 39
2.3.2 Notification d'alertes 41
2.3.3 Architecture des systèmes d'alertes 42
2.3.4 Dimension de qualité 42
2.3.5 Constat 44

2.4 Synthèse **45**

3 Modélisation des alertes *47*

3.1 Introduction **47**

3.2 Modèles d'évaluation **47**

3.2.1 Approche logique 48
3.2.2 Approche probabiliste 49
3.2.3 Logique floue et valeurs linguistiques 49
3.2.4 Fouille de données et apprentissage automatique 53

3.3 Interprétation du temps **54**

3.3.1 Logiques temporelles 55
3.3.2 Relations temporelles 58

3.4 Les séries chronologiques **62**

3.4.1 Approximation des séries chronologiques 62
3.4.2 Langage d'interrogation 63
3.4.3 Abstraction des tendances 63

3.4.4 Détection de tendances indépendamment du domaine 64
3.4.5 Segmentation des séries chronologiques 65

3.5 Contexte 68
3.5.1 Raisonnement temporel 68
3.5.2 Finalité et objectif 69
3.5.3 Sensibilité au contexte 69

3.6 Approche opérationnelle du traitement des données 71
3.6.1 Dimensions temporelles dans les bases de données 71
3.6.2 Fonctions temporelles dans les bases de données 73
3.6.3 Traitement d'événements 74

3.7 Synthèse 75

3.8 Positionnement 76

4 Présentation Générale de TEMPAS 81

4.1 Introduction 81

4.2 Définition et gestion d'alertes 81
4.2.1 Processus 81
4.2.2 Concepts dans TEMPAS 83

4.3 Portée du système d'alertes 84

4.4 Données et connaissances 85
4.4.1 Données 85
4.4.2 Introduction des valeurs linguistiques 86
4.4.3 Définition des situations d'alerte 90
4.4.4 Expression du contexte 93
4.4.5 Synthèse de la section 98

4.5 Indices de qualité 98
4.5.1 Niveau 1 : utilisation des données 99
4.5.2 Niveau 2 : vérification des conditions d'activation 108
4.5.3 Niveau 3 : détection de l'alerte 111
4.5.4 Instant d'évaluation 114
4.5.5 Synthèse de la section 115

4.6 Politique de notification 116
4.6.1 Filtrage d'alertes détectées par une évaluation 116
4.6.2 Notification par comparaison avec des alertes déjà notifiées 122
4.6.3 Synthèse de la section 127

4.7 Synthèse du chapitre 127

5 Implémentation et intégration du moteur d'alertes TEMPAS 129

5.1 Introduction 129
5.1.1 Futura Smart Design 129
5.1.2 TEMPAS 130

5.2 Couche des données 131
5.2.1 Schéma de données de TEMPAS 131
5.2.2 Séparation des modèles 131
5.2.3 Modélisation des observations 133
5.2.4 Accès aux observations 133
5.2.5 Groupe d'entités 134

5.3 Evaluation des alertes 135
5.3.1 Détection des tendances 135
5.3.2 Sélection des Observations 139
5.3.3 Evaluation d'une situation d'alerte 139
5.4 Intégration de TEMPAS avec FSD 142
5.4.1 Configuration du système d'alertes 143
5.4.2 Politique d'évaluation 144
5.4.3 Exploitation d'informations qui ne sont pas générées par le système maître 150
5.4.4 Alertes détectées 151
5.4.5 Application Cliente 154
5.5 Synthèse 164
6 Exemple récapitulatif 165
6.1 Introduction 165
6.2 Situations d'alerte 165
6.2.1 Sous-dialyse 165
6.2.2 Surveillance du cathéter 166
6.3 Autres systèmes d'alertes 166
6.4 Préparation des connaissances 166
6.4.1 Ressources observables 166
6.4.2 Equation d'alerte 170
6.5 Définition des situations d'alerte 170
6.5.1 Sous-dialyse 170
6.5.2 Emergence répétitive 171
6.6 Evaluation des situations d'alerte 172
6.6.1 Politique d'évaluation 173
6.6.2 Données des patients dialysés 173
6.7 Notification des alertes 175
6.8 Adaptation des alertes 178
6.8.1 Adaptations du calcul des valeurs linguistiques 179
6.8.2 Nouvelles informations pour les alertes 179
6.8.3 Notifications 179
6.9 Synthèse 181
7 Validation 183
7.1 Introduction 183
7.2 Indépendance du domaine d'application 183
7.2.1 Présentation de BT-45 183
7.2.2 Identification des situations d'alerte TEMPAS 185
7.2.3 Récupération des données 186
7.2.4 Définition des situations d'alerte TEMPAS 186
7.2.5 Résultats 189
7.2.6 Synthèse 193
7.3 Flexibilité et adaptabilité 194
7.3.1 Participants de l'expérimentation 194
7.3.2 Définition des situations d'alerte 194

7.3.3 Création des scénarii de tests 195
7.3.4 Expérimentation avec les utilisateurs experts 199
7.3.5 Résultats 200
7.3.6 Synthèse 208

8 Conclusion et perspectives 211

8.1 Conclusion 211

8.2 Perspectives 212

8.2.1 Exploitation des observations 212
8.2.2 Qualité des données 212
8.2.3 Retour des utilisateurs 213
8.2.4 Exploration d'autres domaines d'application 214
8.2.5 Impact 214

Bibliographie ..... Erreur ! Signet non défini.

Annexes 225

Table de Figures

Figure 1. Architecture générale d'un système d'alertes 20
Figure 2. Exemple d'un module MLM 30
Figure 3. Connaissances pour la gestion de diabètes gestationnels (GDM) non insulino-dépendant de type II (Asbru). 32
Figure 4. Architecture d'un système de détection de méta-alertes sur un réseau 37
Figure 5. Exemples d'ensembles flous pour la variable taille 51
Figure 6. Evénements et propriétés concernant la réalité modelée de Thérèse dans une période spécifique 56
Figure 7. Relations temporelles d'Allen entre intervalles 59
Figure 8. La relation exprimée pour chaque paire d'intervalles est différente. Néanmoins il est possible de les considérer comme étant la même. 59
Figure 9. La relation exprimée pour chaque paire d'intervalles est la même. Néanmoins il est possible de la considérer comme étant trois relations différentes. 59
Figure 10. Relations entre semi-intervalles de Freksa. 60
Figure 11. Représentation des informations temporelles. 61
Figure 12. Evénements ordonnés par date de fin. 61
Figure 13. Processus impliquant les alertes 82
Figure 14. Diagramme de cas d'utilisation de la définition des situations d'alerte 82
Figure 15. Modélisation d'une observation 86
Figure 16. Valeurs linguistiques d'une ressource observable 87
Figure 17. Fonction d'appartenance correspondante aux ensembles flous 88
Figure 18. Valeurs linguistiques pour l'âge. 88
Figure 19. Modélisation d'une tendance 89
Figure 20. Définition d'une situation d'alerte 90
Figure 21. Relation « commence avant ou en même temps que » 92
Figure 22. Représentation du temps de validité et du temps d'expiration. 94
Figure 23. Temps de validité et d'expiration dans deux contextes différents 95
Figure 24. Exemple de graphe modificateur 96
Figure 25. Fonctions d'appartenance, ensembles flous et valeurs linguistiques lorsque la température est modifiée par le sexe. 98
Figure 26. Evaluation d'une situation d'alerte 99
Figure 27. Indice de confiance de l'observation 100
Figure 28. Indice de confiance de la tendance 102
Figure 29. Détail du calcul de l'indice de confiance de la tendance 102
Figure 30. Graphe modificateur 107
Figure 31. Mise en œuvre des conditions d'activation 113
Figure 32. Indice de confiance de l'alerte à trois instants différents 115
Figure 33. Notification des alertes 116
Figure 34. Deux tendances sur la même série chronologique de la glycémie 118
Figure 35. Trois segments sur la série chronologique de la glycémie 121
Figure 36. Scenarii du cycle de vie de l'alerte 123
Figure 37. Deux alertes consécutives 124
Figure 38. Quatre alertes consécutives 125
Figure 39. Enrichissement en se basant sur les données anciennes 125
Figure 40. Enrichissement en se basant sur les données récentes 126
Figure 41. Schéma de la base de données 131
Figure 42. Séparation des modèles 132
Figure 43. Références faibles 132
Figure 44. Vues contenant les observations 133
Figure 45. Relation ressources observables et vues 134
Figure 46. Groupes de patients 134
Figure 47. Sélection d'observations pour la détection de tendances 136

Table de Figures

Figure 48. Correspondance entre variables et ressources **146**
Figure 49. Fenêtre d'évaluation **150**
Figure 50. Récupération des alertes détectées **152**
Figure 51. Composant : gestion des variables **155**
Figure 52. Composant : groupe de patients **156**
Figure 53. Composant : groupe de destinataires **156**
Figure 54. Composant : équation de l'alerte **157**
Figure 55. Composant : tendances **158**
Figure 56. Conditions temporelles **158**
Figure 57. Conditions intrinsèques **159**
Figure 58. Conditions avec des valeurs numériques **160**
Figure 59. Seuils de filtrage **160**
Figure 60. Configuration et paramétrage de la situation d'alerte **161**
Figure 61. Cycle de vie de l'alerte **162**
Figure 62. Composant : liste d'alertes **163**
Figure 63. Equation d'alerte – dialyse **170**
Figure 64. Equation d'alerte – émergence **170**
Figure 65. Equation d'alerte modifiée – dialyse **180**
Figure 66. Validation des situations d'alerte **186**

Table de Tableaux

Tableau 1. Catégorisation par expressivité 40
Tableau 2. Catégorisation par notification d'alertes 41
Tableau 3. Classification par architecture 42
Tableau 4. Classification des dimensions de qualité 43
Tableau 5. Transformation en valeurs linguistiques 51
Tableau 6. Exemple des règles pour programmer des rendez-vous 52
Tableau 7. Valeurs linguistiques et centres de gravité pour la variable RDV 52
Tableau 8. Transformation des valeurs linguistiques en une valeur numérique 52
Tableau 9. Formules du calcul d'événements de Shanahan. 56
Tableau 10. Requêtes similaires entre SQL Server et Oracle 73
Tableau 11. Positionnement de TEMPAS 78
Tableau 12. Exemple d'observations exploitables par TEMPAS 86
Tableau 13. Transformation de valeurs quantitatives en valeurs linguistiques 88
Tableau 14. Transformation des segments en valeurs linguistiques 90
Tableau 15. Indices de qualité 99
Tableau 16. Exemple du calcul de l'indice linguistique révisé 106
Tableau 17. Interprétation des alertes détectées à partir des indices de qualité 115
Tableau 18. Deux résultats pour la même vérification de conditions d'activation 117
Tableau 19. Deux résultats pour la segmentation d'une série chronologique 118
Tableau 20. Satisfaction des conditions d'activation 118
Tableau 21. Satisfaction des conditions d'activation avec des IAA égales 120
Tableau 22. Satisfaction des conditions d'activation avec des segments dont l'intersection est vide 121
Tableau 23. Normalisation des observations composant une série chronologique 137
Tableau 24. Fichier de configuration 143
Tableau 25. Exemples de calcul du temps d'attente 148
Tableau 26. Attente courte 150
Tableau 27. Changement d'état d'une alerte 154
Tableau 28. Exemples Pourcentage de changement de poids 167
Tableau 29. Ressource observable – Pourcentage de changement de poids (PCP) 167
Tableau 30. Ressource observable – Tension 168
Tableau 31. Score émergence 169
Tableau 32. Ressource observable – Emergence 169
Tableau 33. Politique de notification – sous-dialyse 171
Tableau 34. Politique de notification – sous-dialyse 172
Tableau 35. Données – sous dialyse 173
Tableau 36. Données – émergence répétitive 174
Tableau 37. Résultat des évaluations de la sous-dialyse 174
Tableau 38. Résultat des évaluations de l'« émergence répétitive » 175
Tableau 39. Notification – sous dialyse 176
Tableau 40. Patients cardiaques 179
Tableau 41. Notifications adaptées 180
Tableau 42. Différences entre les systèmes 181
Tableau 43. Résultats attendus des tests 184
Tableau 44. Changements dans les situations d'alerte 188
Tableau 45. Nombre de raffinements par scénario 188
Tableau 46. Résultats attendus O, OV, OVN 190
Tableau 47. Calcul de la précision et l'exactitude avec les catégories OVNI 191
Tableau 48. Calcul de la précision et l'exactitude avec O-I 192
Tableau 49. Conditions d'activation par situation d'alerte 192
Tableau 50. Temps d'exécution 193
Tableau 51. Tests des utilisateurs 197
Tableau 52. Résultats des expérimentations 201

Table de Tableaux

Tableau 53. Difficultés de TEMPAS **204**
Tableau 54. Formulaire SUS **208**

1 Introduction

1.1 Contexte général 19
1.2 Motivations 20
1.3 Proposition 24
1.4 Domaine d'application privilégié 25
1.5 Structure du document 26

1.1 Contexte général

Les systèmes d'information actuels gèrent de plus en plus de données issues des bases de données persistantes mais aussi des données externes provenant de dispositifs de capture ou de services de la toile. Les profils des utilisateurs sont très divers, en raison de leurs compétences, mais aussi de leurs préférences. L'intérêt des systèmes d'alertes dans ces « nouveaux » systèmes d'information est accrut car il pourrait utiliser toutes ces sources de données afin de définir des situations d'alerte qui intéressent les utilisateurs selon leur profil. Ceci accentue le besoin de systèmes d'alertes capables de prendre en compte différents types de données, adaptables aux préférences des utilisateurs pour obtenir des alertes de qualité.

D'un point de vue général, une alerte et une alarme sont des synonymes qui définissent un « *signal automatique d'avertissement de danger* » (WordNet, 2014). Le même dictionnaire définit un système d'alarme comme « *un dispositif qui signale l'occurrence d'un événement indésirable* ». Toutefois, il n'y a pas une définition universelle d'un système d'alertes mais des définitions dans des cadres spécifiques : un système d'alertes médical est défini comme « *des dispositifs de communication visant à demander de l'aide […]* » (Brouhard, 2012) ; ou « *tout dispositif ou ensemble de dispositifs capable de déclencher une alarme lors de la détection automatique de la chaleur, de la fumée* […] » (City of Decatur, 2008).

Dans cette thèse, nous choisissons une acception large où un système d'alertes est un ensemble de processus permettant de définir, détecter et/ou notifier des alertes à partir d'événements intéressants.

La Figure 1 montre l'architecture générale d'un système d'alertes tel que nous le considérons. Les données susceptibles de provoquer des alertes sont issues de différentes *sources de données*. Par exemple, dans le domaine de la santé, les données peuvent correspondre à la température, la tension, le pouls ou d'autres constantes médicales surveillées sur les patients et relevées par des capteurs ou saisies par le personnel médical. La température extérieure, le vent, ou l'humidité relative des villes sont plutôt utilisées dans le domaine de la météorologie et récupérées par des stations météo. La *base d'alertes* contient les conditions à surveiller. Les alertes inclusives sont déclenchées lorsque des conditions sont satisfaites, contrairement aux alertes

exclusives où la non-satisfaction des conditions déclenche des alertes. Le *moteur d'alertes* vérifie les conditions sur les données et détermine s'il s'agit d'une alerte. Le moteur peut utiliser des informations autres que les données utilisées pour vérifier les conditions afin de réduire le nombre d'alertes à notifier, par exemple, si des alertes similaires ont été déjà détectées, ou si la source de la donnée n'est pas fiable. Finalement, le *système de notification* a la responsabilité de notifier les utilisateurs des alertes détectées.

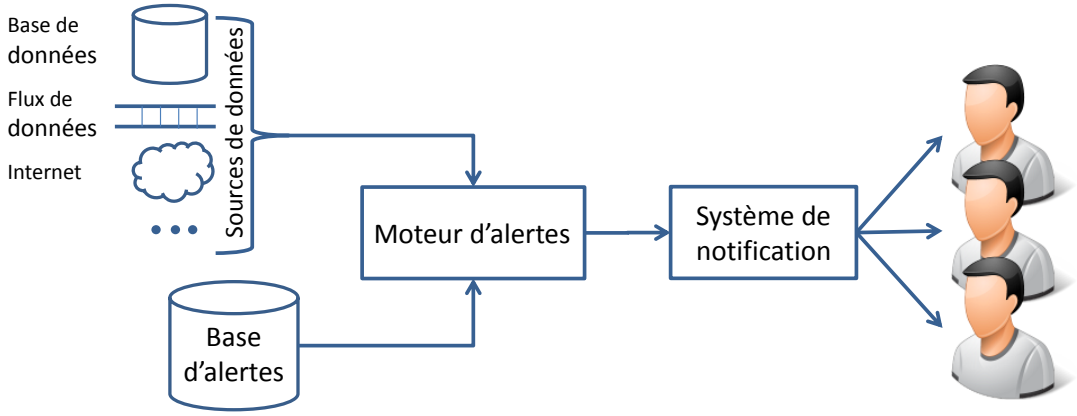


Figure 1. Architecture générale d'un système d'alertes

1.2 Motivations

La motivation de ce travail résulte de deux constatations : le fait que les systèmes d'alertes existants sont définis dans un contexte unique d'application ce qui diminue leur qualité intrinsèque et le fait que les utilisateurs ne sont pas en mesure de définir le niveau de qualité du système en fonction du contexte. Nous nous proposons donc dans cette thèse de répondre à ces deux constats.

1.2.1 Contexte d'application des alertes

Nombreux sont les systèmes d'information dont des situations d'alerte sont pré-paramétrées (Nihon Kohden Corporation, 2014), (Diatelic HD, 2014). Ces situations d'alerte pré-paramétrées présentent de nombreux inconvénients car elles génèrent beaucoup de faux-positifs, soit des alertes détectées par le système et qui ne sont pas considérées ainsi par les utilisateurs, et des alertes inutiles, soit des vrais alertes qui n'ajoutent pas une valeur lorsqu'elles sont notifiées. Ces systèmes sont faiblement adaptables ou inadaptables et ne permettent pas une gestion fine des alertes détectées.

1.2.1.1 Gestion d'alertes

Une gestion riche des situations d'alerte et des alertes détectées implique des interfaces graphiques ergonomiques qui facilitent la tâche des utilisateurs *concepteurs* et *consommateurs* d'alertes. Dans la plupart des cas, le rôle de *concepteur d'alertes* autorise seulement à fixer des seuils ou activer/désactiver des alertes puisque le modèle sous-jacent a une expressivité pauvre. C'est le cas des alertes du type : « déclencher une alerte si la température dépasse 37.7 °C ». Des modèles très

expressifs comme ceux utilisés dans HELP (Musen, et al., 2006), l'Arden Syntax (Pryor & Hripcsak, 1993), ou Asbru (Shahar, et al., 1998) obligent que le rôle du *concepteur* soit attribué uniquement à des utilisateurs experts ayant une double compétence en santé et en informatique. En outre, les interfaces graphiques qui permettent aux *concepteurs* de définir des situations d'alerte expressives sont inexistantes ou difficiles à utiliser.

1.2.1.2 Adaptation des situations d'alerte

L'adaptation des situations d'alerte peut mener à la réduction des faux-positifs ou des alertes inutiles. Toutefois, l'adaptabilité est dépendante du modèle. Si le modèle est complexe, l'utilisateur doit développer des compétences spécifiques. Si le modèle est trop simple, les adaptations possibles ne sont pas en mesure de réduire les faux-positifs ou les alertes inutiles.

Comme la plupart des utilisateurs ont un rôle de *consommateurs* d'alertes, les systèmes d'information optent pour des situations d'alerte pré-paramétrées. Les paramètres, dans ce cas, sont choisis dans un contexte spécifique qui n'est pas toujours le contexte d'utilisation. Nous utilisons la définition de contexte proposée par (Dey, 2001) qui est très expressive et très répandue dans la littérature. *« Le contexte est toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une personne, un endroit, ou un objet qui est considéré pertinent à l'interaction entre un utilisateur et une application, y compris l'utilisateur et les applications elles-mêmes »*.

Par exemple, il est connu que suite à une opération, la tension d'un patient descend environ de 20% à cause des produits utilisés. Si la tension d'une personne adulte doit se trouver entre 90 et 140 mm Hg, le système d'alertes dans la salle de réveil déclenche une alerte si la tension ne se trouve pas entre 72 et 112 mm Hg. Toutefois, cette alerte ne prend pas en compte le contexte du patient, qui peut être hypertendu et prendre des médicaments hypotenseurs dans sa vie quotidienne.. Dans cette situation, le système d'alertes détectera beaucoup des faux-positifs parce que la tension post-opératoire sera sans doute plus élevée que 112 mm Hg.

Une solution serait de fixer manuellement un seuil plus élevé pour le patient hypertendu à partir des valeurs de tension lors de la *consultation* et en *pré-opération*. Cette solution est cependant peu pertinente car elle obligerait les utilisateurs à identifier en « temps réel » quand la situation d'alerte n'est pas adaptée au patient.

Notre première motivation est donc l'adaptation des systèmes d'alertes à des contextes changeants ou différents en minimisant les modifications apportées au système.

1.2.1.3 Prise en compte du contexte

L'adaptation au contexte est nécessaire puisque tout environnement réel est unique (Gee & Moorman, 2011). De nombreux travaux se sont focalisés sur les applications et la sensibilité au contexte (Dey, et al., 2001) (Henricksen, 2003) (Mayrhofer, 2004), le but étant de fournir aux concepteurs d'applications un outil pour

que les applications elles-mêmes puissent être adaptées. L'approche la plus courante est de proposer une architecture en couches dont l'une est responsable des informations contextuelles.

Des efforts ont été réalisés sur la récupération/découverte du contexte dans des environnements très dynamiques ; en premier lieu par la fusion de données issues de capteurs hétérogènes qui rendent des informations plus riches en exprimant mieux un contexte (Gürgen, 2007) (Petit, 2012). Par exemple « capteur 1 : température élevée » et « capteur 2 : salon », « résultat : température élevée au salon » ; ensuite à l'aide de techniques d'apprentissage automatique (Laerhoven, 2001) ou de fouille de données (Heierman & Cook, 2003) pour abstraire le contexte à partir d'un grand nombre de données issues des capteurs.

1.2.1.4 Problématique

Comment obtenir des situations d'alerte adaptées au contexte, dans n'importe quel domaine d'application, exploitables par les utilisateurs ?

Utiliser un grand nombre de données issues de capteurs ne résout pas notre problème. Puisque les situations d'alerte doivent être adaptées au contexte, il serait nécessaire d'avoir de nombreuses données relatives à tout type de contexte, à la maison, en salle de réveil, au bureau, etc. De plus, ce sont des contextes avec une péremption où des nouveaux capteurs peuvent changer le contexte qui oblige les algorithmes à réévaluer le nouvel ensemble des données issues de tous les capteurs afin d'abstraire le nouveau contexte. Une telle mise en place nécessite des architectures et des déploiements spécifiques. En outre, ces contextes ne prennent pas en compte les préférences des utilisateurs qui peuvent ne pas être mesurables ou changer constamment.

Nous proposons une approche où les utilisateurs prennent en main le système d'alertes afin de maîtriser ce qu'ils font. Ceci réduit la frustration possible (Krall & Sittig, 2002) de l'utilisateur et augmente sa capacité d'adapter le système d'alertes au contexte par et pour lui-même. Nous envisageons une solution générique, intégrable avec les systèmes d'information existants et dont les alertes sont expressives.

1.2.2 Dimensions de qualité

La notion de qualité est subjective et les définitions varient selon le contexte d'utilisation. Les dimensions de qualité sont, pour nous, des valeurs quantitatives ou qualitatives utilisées pour représenter des propriétés.

Certaines dimensions de qualité sont exprimées par des besoins non-fonctionnels des systèmes. Nous ne sommes pas intéressés par ce type de dimensions de qualité, mais par des dimensions de qualité qui concernent directement la notion alerte.

1.2.2.1 Précision et exactitude

Deux dimensions de qualité sont largement utilisées pour caractériser les systèmes de classification tels que les systèmes d'alertes, la *précision* et l'*exactitude* (Njogu, et al., 2013). La précision exprime le ratio d'alertes correctement identifiées. L'exactitude prend en compte les faux-négatifs, soit des alertes qui n'ont pas été identifiées mais qui auraient dû l'être. Un système dit « précis » exprime que toutes les alertes qu'il identifie sont des vraies alertes tandis qu'un système dit « exact » exprime que toutes les alertes sont identifiées et qu'aucune n'est manquée (Phansalkar, et al., 2010). Notons qu'une fausse alerte peut réduire la qualité de service.

La précision et l'exactitude se calculent à partir d'un ensemble de données pour lesquelles le résultat attendu est connu à l'avance. Si pour l'entrée e_1 une alerte doit être détectée et que le système d'alertes la détecte, il s'agit d'un *vrai-positif*. Inversement, si le système d'alertes ne la détecte pas il s'agit d'un *faux-négatif*. Si pour l'entrée e_2 aucune alerte ne doit être détectée et que le système en détecte une, il s'agit d'un *faux-positif*. Par contre, si le système n'en détecte pas, c'est un *vrai-négatif*. L'inconvénient avec ces dimensions de qualité est que le résultat attendu pour une entrée doit être connu afin que le système d'alertes puisse évaluer les entrées et qu'il soit possible de comparer les résultats obtenus avec les résultats attendus. Cela implique que la précision et l'exactitude du système indiquent son comportement par rapport à des situations connues représentées par un ensemble d'entrées. Supposons un système d'alertes d'incendie dont la précision est de 95% c'est-à-dire que sur 100 alertes détectées, 95 sont vraies. La situation d'alerte incendie est représentée par des données issues des capteurs de CO2, O2 et température. La précision exprime le comportement du système par rapport à la situation « incendie ». Par contre, la précision du système est inconnue pour la situation « incendie et personnes présentes dans le bâtiment ». Pour la calculer, il faudrait introduire de nouvelles données mais surtout connaître la réponse attendue. Ainsi, pour un système d'alertes qui permet de *concevoir* des situations d'alerte et les adapter en temps réel, calculer la précision ou l'exactitude devient une tâche difficile car la prise en compte des *consommateurs* d'alertes est indispensable pour valider les alertes détectées et les catégoriser en vrais ou faux-positifs. Puisque les faux-négatifs et les vrais-négatifs ne sont pas affichés car ils ne sont pas considérés comme des alertes, il est encore plus difficile de calculer l'exactitude.

1.2.2.2 Autres dimensions associées aux alertes

Des dimensions de qualité comme l'*interprétabilité*, la *valeur ajoutée* ou la *pertinence* sont importantes pour satisfaire les besoins des utilisateurs (Wang & Strong, 1996) (Strong, et al., 1997). Cependant, leur mesure reste subjective. Leur importance s'accroît dans le domaine de la santé où les systèmes d'alertes sont délaissés à cause du manque de finesse de ces paramètres (Cvach, 2012).

Certains travaux utilisent des dimensions de qualité dans le but de réduire les faux-positifs et les alertes inutiles. Les dimensions telles que la *précision, la fiabilité, l'exactitude*, et la *sensibilité* sont utilisées par (Bakar & Belaton, 2005) afin de raisonner

à un niveau plus élevé et rendre des alertes plus riches. Le travail de (Chandrasekaran, et al., 2006) utilise l'*écart* entre le nombre d'alertes, la *pertinence* et la *variété* des alertes générées afin de préparer des méta-alertes. Le travail de (Alsubhi, et al., 2012) se focalise sur la détection d'intrus et utilise sept dimensions pour calculer un score et prioriser ainsi les alertes détectées : *l'applicabilité*, *l'importance de la victime*, *le statut du capteur*, *le placement du capteur*, *la sévérité de l'attaque*, *la vulnérabilité du service*, *et la relation entre les alertes*. Les dimensions de qualité sont spécifiques au domaine d'application et se calculent souvent avec des facteurs prédéfinis comme par exemple l'adresse IP d'où provient l'attaque et si la machine attaquée est à jour.

1.2.2.3 Problématique

Quelles dimensions de qualité expriment la qualité associée aux alertes détectées, indépendamment du domaine d'application et qui permettent de réduire les faux-positifs et les faux-négatifs ?

Selon le contexte de l'application, un faux-négatif est plus dangereux qu'un faux-positif. Ainsi, il vaut mieux sacrifier la précision et améliorer l'exactitude. Lorsque le système d'alertes est indépendant du domaine, il est souhaitable que les faux-positifs et/ou les faux-négatifs puissent être réduits. Pour cela, il est nécessaire de définir des indices de qualité également indépendants du domaine.

1.3 Proposition

Nous proposons une représentation des situations d'alerte permettant de générer des alertes et de calculer deux indices de qualité qui leur sont associés. Nous utilisons cette représentation dans un système d'alertes paramétrable par les utilisateurs, personnalisable, adaptable au contexte, indépendamment du domaine d'application. Notre système est nommé TEMPAS (TEMporal Pluggable Alert System).

Comment définir les situations d'alerte ?

L'utilisateur prépare les connaissances nécessaires au système d'alertes afin d'exploiter les observations qui vont déterminer l'état ou la tendance d'une ressource. Par exemple, pour la ressource *température,* le système peut déterminer à partir d'une observation l'état *« élevée »*, ou la tendance *« à la hausse »* à partir d'un ensemble d'observations. Une fois les connaissances préparées, l'utilisateur définit les *situations d'alerte* en utilisant des *conditions d'activation* basées sur des *valeurs linguistiques* qui expriment un *état* ou une *tendance*. Le système d'alertes utilise les situations d'alerte définies par les utilisateurs ainsi que les connaissances pré-paramétrées pour déterminer si la situation d'alerte se produit à un *instant* donné suite aux observations connues. Si les observations connues satisfont au moins *partiellement* une *partie* des conditions d'activation, le système *détecte une alerte*. Le *degré de satisfaction* des conditions, le nombre de conditions satisfaites, ainsi que la *fraîcheur* des observations utilisées permettent au système de calculer *deux indices de qualité* associés aux alertes détectées. L'*indice d'applicabilité* exprime dans quelle mesure une *entité* est concernée par l'alerte, et l'*indice de confiance* exprime la fiabilité de l'alerte relativement à l'obsolescence des données utilisées lors de sa détection. L'*entité*

représente qui est observé et la *ressource* ce qui est observé (par exemple, température corporelle de Charles)

Puisque les alertes détectées peuvent être nombreuses, nous définissons le cycle de vie de l'alerte qui est un mécanisme permettant de maîtriser la sur-notification d'alertes similaires en notifiant uniquement les alertes pertinentes. Les alertes sont tout d'abord filtrées à partir d'indices de qualité. Parmi les alertes restantes, le système filtre par expressivité : choisir les tendances les plus durables et les valeurs linguistiques les plus expressives. Ensuite, dans le cas d'alertes consécutives, le système conserve uniquement les alertes souhaitées par les utilisateurs, comme celles dont l'indice d'applicabilité augmente.

Dans quel objectif ?

L'objectif est d'élargir les contextes d'utilisation des systèmes d'alertes. TEMPAS, est indépendant du contexte d'application et fidélise les utilisateurs qui s'approprient le système au fur et à mesure de son utilisation. L'utilisateur peut adapter lui-même les situations d'alerte au contexte afin d'obtenir des alertes qu'il considère comme étant de qualité. La personnalisation a lieu dans la préparation de connaissances, la définition des situations d'alerte, ainsi qu'au moment de la notification. L'utilisateur sait comment interpréter les indices de qualité et s'en sert pour réduire les faux-positifs et faux-négatifs et contrôler la sur-notification. Leur utilisation permet dans le temps à l'utilisateur de modifier les connaissances préparées ou ajuster les situations d'alerte. Le système d'alertes est flexible, paramétrable, et facile à utiliser. La construction des situations d'alerte est un processus aisément compréhensible et rapide. L'utilisateur adapte le système d'alertes en temps réel afin de mieux prendre en compte le contexte lors d'une situation d'alerte changeante ou inattendue.

Sans chercher à définir un système d'alertes intelligent, nous proposons de tirer parti des systèmes existants en apportant des fonctionnalités de dynamisme, d'évolutivité, de paramétrage facile et d'adaptation en temps réel au contexte d'utilisation.

1.4 Domaine d'application privilégié

Cette thèse CIFRE s'est déroulée grâce à une coopération entre le laboratoire d'informatique de Grenoble et la société Calystene. Cette dernière est spécialisée dans les solutions informatiques médicales et compte plus de vingt ans d'expérience ; expérience dont nous avons pu profiter grâce au contact fréquent avec des experts du domaine et des informaticiens spécialistes des SI santé. C'est ainsi, que nous sommes arrivés à proposer un système d'alertes capable d'être adapté aux différents environnements de santé et de l'intégrer dans un système d'information existant plus large.

1.5 Structure du document

Le reste de cette thèse est divisé en sept chapitres :

- L'état de l'art sur les systèmes d'alertes existantes est présenté dans le **chapitre 2**.
- Le **chapitre 3** est consacré aux modèles qui permettent aux systèmes de déterminer si une alerte s'est produite.
- Le **chapitre 4** introduit TEMPAS.
- Le **chapitre 5** est dédié à l'implémentation de TEMPAS et son intégration avec Futura Smart Design, un système de gestion d'informations de santé développé par Calystene.
- Le **chapitre 6** présente un exemple récapitulatif de TEMPAS concernant la définition des situations d'alerte dans le domaine de dialyse à domicile.
- Le **chapitre 7** regroupe deux validations de TEMPAS.
- Le **chapitre 8** synthétise dans la conclusion ce qui a été réalisé et expose nos perspectives.

L'état de l'art est développé dans les chapitres 2 et 3. Le positionnement de notre proposition TEMPAS est particulièrement appuyé par les tableaux de synthèse de la section 2.3 ainsi que par la notion de valeurs linguistiques et les aspects temporels et contextuels introduits dans le chapitre 3.

2 Les systèmes d'alertes

2.1 Introduction 27
2.2 Les systèmes d'alertes existants 27
2.3 Catégorisation des systèmes d'alertes 39
2.4 Synthèse 45

2.1 Introduction

Dans cette première partie de l'état de l'art, nous nous intéressons aux systèmes d'alertes dans différents domaines d'application que nous présentons dans la section 2.2. Ensuite, nous catégorisons dans la section 2.3 les systèmes étudiées en fonction de différents critères que nous synthétisons dans quatre tableaux. Les deux premiers tableaux (expressivité et contrôle de notification) relèvent d'une demande générale de la diversité des acteurs (experts, utilisateurs, etc.) dans les différents domaines d'application. Le troisième tableau sur les architectures de ces systèmes concerne les préoccupations des concepteurs et développeurs informaticiens. Le dernier tableau précise les exigences de qualités particulièrement importantes dans des domaines sensibles. La notion de temporalité est inhérente à ces environnements avec alertes (cf. chapitre 3). Elle est analysée au sein de nos différents critères. On retrouve par exemple, l'étude des systèmes en terme de contraintes temporelles, concurrence, événements, etc. Nous concluons cette première partie avec une synthèse du chapitre.

2.2 Les systèmes d'alertes existants

Nous considérons les systèmes d'alertes comme l'ensemble des processus permettant de définir, détecter et/ou notifier des alertes. Une alerte est un événement intéressant qui attire d'une façon ou d'une autre l'attention des utilisateurs. Nous avons étudié des nombreux systèmes d'alertes que nous regroupons selon leur finalité : le pourquoi de ces systèmes.

2.2.1 Personnes âgées

Parmi les systèmes d'alertes industrialisés les plus simples, nous trouvons ceux qui sont activés manuellement par les personnes lors des situations de danger comme (Bay Alarm Medical, 2014) ou (LifeFone, 2014). Ces systèmes ne définissent pas des situations d'alerte ni des modèles pour les évaluer. Les utilisateurs détectent eux-mêmes les alertes et choisissent de les notifier au moment d'appuyer sur le bouton pour établir une connexion avec un centre d'appels. Un *opérateur formé* discute avec l'utilisateur et selon l'urgence de la situation appelle les pompiers, le 911, la famille, ou les voisins. Le point fort de ce type de systèmes reste la notification des alertes qui s'appuie sur des infrastructures déjà déployées (réseau mobile, téléphone, etc.). Ces systèmes ont deux types d'utilisateurs, l'utilisateur qui est informé et l'utilisateur qui appuie sur le bouton. Ce dernier représente la source des données. L'opérateur est en

charge d'évaluer l'urgence de la situation et de notifier les utilisateurs concernés. On peut supposer que la qualité des alertes est élevée aux yeux des utilisateurs puisqu'ils sont à l'origine de leur activation et que la notification est réalisée par un opérateur formé pour ce type de situations.

2.2.2 Notification de masse

La solution proposée par (Reach Alert, 2014) fait partie des systèmes d'alertes simples à mettre en place. Deux rôles sont définis : les *administrateurs* et les *destinataires*. Les administrateurs décident quand une alerte doit être notifiée. Les destinataires reçoivent les alertes générées par l'administrateur par le moyen de leur préférence comme un sms, un message vocal, ou un mail. Parmi les qualités de ce système, la scalabilité est décisive. Si les nombre de messages à envoyer est très grand, le système alloue les ressources nécessaires pour les envoyer en seulement quelques secondes. Ce système peut être utilisé dans plusieurs scénarii comme l'annulation d'un cours, la fermeture d'une l'école ou pour informer les habitants d'un quartier de la présence de malfaiteurs. La solution (ReachPlus , 2014) se démarque grâce aux interfaces graphiques de l'application cliente accessible par un navigateur web et par des téléphones intelligents. Elle permet de créer facilement *des groupes de destinataires* en temps réel à partir de critères tels que les utilisateurs en ligne ou dans une zone géographique et de faire un suivi de la livraison des alertes. De plus, des images ou des liens url peuvent être partagés en même temps que l'alerte.

2.2.3 Sécurité publique

Le gouvernement belge a déployé un système (Be Alert, 2014) dans un cadre de sécurité publique. *« BE-Alert est le nouveau système qui permet de diffuser des messages rapidement et de manière très ciblée, selon la nature de la catastrophe et le niveau de crise. En cas de catastrophe, la population était, jusqu'à présent, alarmée principalement par des sirènes, […] »*. Le gouvernement français a aussi déployé (ASTER, 2010), Alerte et Surveillance en TEmps Réel. Il existe également un système d'alertes développé pour la gestion d'alertes épidémiologiques (Gaudin, et al., 2012). *« Le système est déployé dans les antennes médicales […] Ce serveur informatique est connecté à un réseau installé en métropole qui permet d'analyser en permanence la situation sanitaire. Les données transmises sont anonymes ».* Dans le cadre de l'épidémiologie, les alertes ont un caractère précoce de prévention.

2.2.4 Monitoring continu

Des dispositifs de monitoring ont des mécanismes alertes. Le moniteur de chevet BSM-9101 (Nihon Kohden Corporation, 2014) dans le domaine de la santé est utilisé pour surveiller le rythme cardiaque, la pression artérielle, les niveaux d'oxygène, de CO2, les niveaux de glucose, etc. Les situations d'alerte sont définies à partir des *paramètres* médicaux surveillés et *des plages de valeurs normales*.

Exemple

Alerte si température observée inférieure à 36,4 °C ou si température observée supérieure à 37,2 °C.

Le modèle d'évaluation est basé sur les opérateurs relationnels de base ≤, <, =, >, ≥. Les utilisateurs (le corps médical) sont en mesure d'activer, désactiver, ou modifier les plages normales pour chacun des paramètres surveillés. Dans le cas du BSM-9101, le dispositif fournit quatre entrées pour quatre paramètres fixes, et trois entrées de plus à partager entre cinq paramètres. Comme nous en discuterons dans la section 2.3.5, ce type de dispositifs avec des situations d'alerte aussi simples produit un nombre très élevé d'alertes manifestées avec des signaux sonores et visuels. On constate que nombreuses sont les fausses alertes (faux-positifs) produites par le mauvais état des capteurs ce qui provoque le désintérêt rapide des utilisateurs. Les fabricants permettent aux dispositifs de se connecter en réseau avec un moniteur central comme le CNS-9601 (Nihon Kohden Corporation, 2014) et de réduire ces fausses alertes. *«Une nouvelle fenêtre d'alertes multi patient améliore la précision des alertes et aide à réduire les fausses alertes »*. Il s'agit d'une fenêtre dédiée à la gestion d'alertes qui permet de mieux les configurer pour réduire les fausses alertes. Les utilisateurs peuvent : paramétrer les seuils d'alertes concernant les signes vitaux, afficher/cacher la fréquence d'arythmies pour voir les effets des médicaments anti-arythmie, et utiliser un nouveau type d'alertes nommé *alerte-technique*. Les alertes-techniques sont interprétées selon le nombre d'*alertes détectées* dans la dernière heure. Ainsi, si moins de dix alertes ont été détectées dans la dernière heure, l'alerte-technique est « bonne » en augmentant la *confiance* de l'alerte détectée. Si le nombre se trouve entre dix et vingt-neuf, l'alerte-technique est « pas mauvaise », dans les autres cas, l'alerte-technique est « mauvaise », ce qui signifie qu'il est nécessaire de changer les capteurs. Les alertes détectées s'affichent visuellement sur l'écran du moniteur central avec des couleurs paramétrables. Le système permet de notifier les alertes par l'activation de la lumière, d'une sirène et d'un son.

2.2.5 Prévention

Toujours dans le domaine médical, la stratégie informatique en soins de santé de Philips est basée sur l'*acquisition*, l'*analyse*, l'*interprétation*, et la *présentation* des données des patients au corps médical (Philips, 2009). Dans le cas spécifique des unités de soins intensifs, Philips propose le système eICU (Philips, 2011), (Philips, 2014) qui intègre le composant Smart Alert Prompts. C'est un module de *prévention* qui vise à alerter avant qu'un évènement ait lieu. Les alertes sont détectées à l'aide d'algorithmes d'apprentissage automatique entraînés avec des données issues de plus d'un million de patients. Parmi les types de données utilisés, on trouve les *signes vitaux*, les *médicaments administrés*, les *résultats de laboratoire*, et le *plan de soins*. Les alertes sont personnalisées pour chaque patient à partir de son état de référence et de son état sous-jacent. Ainsi, le corps médical est en mesure de fournir de soins individuels et personnels sur un grand nombre de patients en réduisant les fausses alertes. Les algorithmes sont paramétrés pour évaluer des situations d'alerte *préconfigurées* et déployées avec le composant Smart Alert Prompts. Les alertes détectées se manifestent visuellement en couleur par d'autres composants.

2.2.6 Partage de connaissances

Des efforts ont été réalisés pour développer des langages de programmation capables de formaliser et partager des connaissances médicales entre systèmes hétérogènes. Nous avons étudié principalement deux entre eux, L'Arden Syntax et Asbru.

```
01 maintenance:
02      title: Alert on low hematocrit;;
03      filename: low_hematocrit;;
04      veraion: 1.00;;
05      institution: CPMC;;
06      author: George Hriposak, M.D. (hripcsalcucis.columbia.edu);;
07      specialist: ;;
08      date: 1993-10-31;;
09      validation: testing ;;
10 library:
11      purpose: Warn provider of new or worsening anemia. ;;
12      explanation: Whenever a blood count result is obtained, the hematocrit is
checked to see whether it is below 30 or at least 5 points below the previous
value.;;
13      keywords: anemia; hematocrit ;;
14 knowledge:
15      type: data-driven;;
16      data:
17      blood count storage := event ('complete blood count');;
18      hematocrit := read last ('hematocrit') ;;
19      previous_hct := read last ('hematocrit') where it occurred before the time of
        hematocrit);;
20      evoke: blood_count_storage;;
21      logic:
22      if hematocrit is not number then conclude false ;;
23      elseif hematocrit <= previous_hct - 5 or hematocrit < 30 then conclude
        true ;;
24      endif ;;
25      action:
26      write "The patient's hematocrit ('hematocrit') is
27      low or falling rapidly.";;
28 end:
```

Figure 2. Exemple d'un module MLM

2.2.6.1 Principes généraux de l'Arden Syntax

L'Arden Syntax est un langage qui fait partie du standard d'interopérabilité de santé HL7 qui cible l'*échange*, l'*intégration*, le *partage*, et la *récupération* des informations médicales électroniques. L'Arden Syntax est un langage procédural utilisé pour écrire des modules logiques cliniques (MLM de l'anglais Medical Logic Module) construits à base de règles pour définir des connaissances médicales partageables (Pryor & Hripcsak, 1993). Un MLM est un fichier compilable composé de trois sections.

La première section, la *maintenance* contient le titre, la version, l'auteur, et d'autres informations pour faciliter la maintenance du module. La deuxième section, la *librairie*, explique à quoi sert le module, comment il marche, les mots clés destinés à faciliter son partage, et des liens et citations utiles pour expliquer le comportement du module. La troisième section contient les *connaissances* médicales, le cœur du module. Parmi les connaissances, l'utilisateur exprime : quand le module doit être appelé, quelles observations, seuils, et autres données doivent être utilisées, l'appel d'autres modules, les règles if-then qui composent la logique du module et les actions à prendre lorsque les prémisses sont satisfaites comme envoyer une alerte ou appeler d'autres modules MLM. Il possible de définir l'urgence de l'action avec une numération (1-99). Un exemple d'un MLM (Hripcsak, 1994) se trouve dans la Figure 2.

Les requêtes de la section *data* peuvent exprimer quelques contraintes temporelles entre des points temporels et des intervalles-points. La section *data* est séparée de la section *logique* afin de faciliter la réutilisation des modules. De cette façon, la partie prenante est en charge de modifier seulement la section *data* afin de l'adapter au modèle de sa base de données. L'Arden Syntax a été utilisée pour définir des directives cliniques, et des guides de bonne pratique (Jenders & Hripcsak, 1995).

2.2.6.2 Principes généraux d'Asbru

Asbru est un langage qui propose une représentation des directives cliniques (partageables) avec les connaissances nécessaires, lisible par les humains et par les machines (Shahar, et al., 1998). Le langage spécifie comment déclencher l'exécution des tâches attachées aux directives cliniques à partir des informations contenues dans des dossiers électroniques. Le langage est similaire à un langage de programmation et permet d'exprimer des contraintes temporelles telles que des processus séquentiels, parallèles, ou cycliques. Lorsque l'utilisateur *conçoit* une directive clinique, il décrit les *actions*, le *plan prévu*, et les *états* du patient.

Exemple

Administrer de l'amitriptyline le soir et un analgésique deux fois par jour et la température du patient doit rester entre 37,4 °C et 37,7 °C.

Les actions, le plan prévu, et les états du patient sont déduits à partir des informations stockées et observées. Asbru se sert des systèmes comme RESUME pour l'abstraction temporelle et déduire ainsi qu'un médicament a été administré le *matin* et le *soir* (Shahar & Musen, 1993).

La Figure 3 montre une partie d'une directive clinique pour la « gestion du diabète gestationnel (GDM) non insulino-dépendant de type II » écrite en Asbru. Une situation d'alerte représente une directive clinique dont les alertes sont détectées si les conditions sont complètes et que le système transmet *true* pour « GDM type II » (lignes 41 et 42 de la Figure 3). Les situations d'alerte sont définies à partir des conditions d'activation très expressives, comme utiliser des seuils différents selon la période de la journée afin de comparer le niveau de glucose. Cette variété de conditions permet d'exprimer des alertes complexes. Cependant, la complexité de la

syntaxe rend très difficile la définition des situations d'alerte par les utilisateurs. Des améliorations ont été réalisées au niveau des interfaces graphiques spécialisées pour l'acquisition de connaissances basées sur des outils comme PROTEGE (Tu, et al., 1995), et traduites ensuite en Asbru.

```
01:  (PLAN observing-GDM
02:
03:  ;; the following time-annotations are local to the GDM example
04:  (DOMAIN-DEPENDENT TIME-ASSIGNMENT
05:     (SHIFTS DELIVERY <- 38 WEEKS) ;; domain-specific time shift from the CONCEPTION point
06:     (POINT CONCEPTION <- (ask (ARG "what is the conception-date?")))
07:  (ABSTRACTION-ASSIGNMENT
08:     (CYCLIC
09:        MIDNIGHTS <- [0, 0 HOURS, 24 HOURS]
10:        BREAKFAST-START-TIME <- [0, 7 HOURS, 24 HOURS]))
11:  (PREFERENCES
12:     (SELECT-METHOD EXACT-FIT)   ;; The match in the filter conditions needs to be exact
13:     (START-CONDITION AUTOMATIC)) ;; the plan starts as soon as it is ready, no user input
14:
15:  (INTENTION:INTERMEDIATE-STATE
16:     (MAINTAIN blood-glucose-post-meal (<= 130) GDM-Type-II
17:        [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION])
18:     (MAINTAIN blood-glucose-fasting (<= 100) GDM-Type-II
19:        [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION])
20:  (INTENTION:OVERALL-STATE
21:     (AVOID STATE(blood-glucose) HIGH GDM-Type-II
22:        [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [7 DAYS,_], CONCEPTION]))
23:     ;; avoid, throughout the guideline, a period of high blood-glucose level lasting more than 7 days
24:
25:  (FILTER-PRECONDITIONS
26:     (one-hour-GTT (140, 200) pregnancy
27:           [24 WEEKS, 24 WEEKS], [26 WEEKS, 26 WEEKS], [_,_], CONCEPTION])
28:  (SETUP-PRECONDITIONS
29:     (PLAN-STATE one-hour-GTT COMPLETED
30:        [[24 WEEKS, 24 WEEKS], [26 WEEKS, 26 WEEKS], [_,_], CONCEPTION])
31:        ;; The patient must have completed a glucose-tolerance test (another plan in the library)
32:  (SUSPEND-CONDITIONS
33:     (STATE(blood-glucose) HIGH GDM-Type-II
34:        [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [4 DAYS,_], CONCEPTION]
35:        (SAMPLING-FREQUENCY 24 HOURS)))
36:        ;; suspend if high blood-glucose level exists for at least 4 DAYS
37:  (ABORT-CONDITIONS
38:     (insulin-indicator-conditions TRUE GDM-Type-II *
39:        (SAMPLING-FREQUENCY 24 HOURS)))
40:  (COMPLETE-CONDITIONS
41:     (delivery TRUE GDM-Type-II * (SAMPLING-FREQUENCY 24 HOURS)))
42:  (REACTIVATE-CONDITIONS
43:     (STATE(blood-glucose) (OR NORMAL SLIGHTLY-HIGH) GDM-Type-II
44:        [[24 WEEKS,  24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION]
45:     (SAMPLING-FREQUENCY 24 HOURS)))
46:
47:
48:  (DO-ALL-TOGETHER
49:     (glucose-monitoring)
50:     (nutrition-management)
51:     (observe-insulin-indicators)))
52:        ;; the plan body is a concurrent one; the three plans start together and all need to complete
```

Figure 3. Connaissances pour la gestion de diabètes gestationnels (GDM) non insulino-dépendant de type II (Asbru).

2.2.6.3 Application de l'Arden Syntax

HELP est un système d'information qui détecte des alertes lorsqu'il constate des anomalies dans le dossier médical d'un patient (Musen, et al., 2006). Les alertes se trouvent dans des modules MLM écrits avec l'Arden Syntax. Les données utilisées pour évaluer les situations d'alerte sont toutes celles accessibles par le système comme les

médicaments administrés, les *résultats de laboratoire*, même *la probabilité des maladies*. Le système utilise une architecture orientée événements afin d'évaluer les situations d'alerte. Chaque fois que les informations d'un patient deviennent disponibles, sans importer la source, Help vérifie si les données transmises correspondent aux critères pour appeler des modules MLM. Le cas échéant, le système HELP évalue les MLM sur le patient et vérifie leur pertinence. Les alertes détectées sont notifiées aux utilisateurs concernés via le système d'information HELP et des rapports écrits. Le système de (Jenders & Hripcsak, 1995) se base aussi sur les modules MLM. Ici, les utilisateurs peuvent visualiser les alertes concernant le dossier du patient en cours triées par ordre chronologique décroissant. Le système dispose de dix-huit modules définissant dix-huit situations d'alerte. Quatorze sont des modules classés comme des alertes (sévères) et quatre comme des interprétations (non sévères). En moyenne, ces modules ont généré 1080 alertes et 50357 interprétations différentes par mois. Les alertes ont été souvent non prises en compte. 73.5% et 1.3% sont les pourcentages des alertes les plus et les moins aperçues faisant référence à deux situations d'alerte différentes. Toutefois, il reste difficile d'affirmer qu'une situation d'alerte est *pertinente* après avoir été *vue*, il n'y a pas de corrélation entre les deux aspects.

2.2.6.4 Analyse de l'Arden Syntax

L'objectif de cette syntaxe est de faire participer le corps médical au développement et à la maintenance des modules logiques médicaux (MLM). La syntaxe est divisée en sections afin de faciliter la compréhension par les utilisateurs, qui malgré les efforts, doivent développer des compétences en programmation. Le langage permet de développer des modules pour la définition des situations d'alerte. Un module est développé pour chaque situation d'alerte, laquelle est composée par des nombreuses conditions exprimées comme des règles logiques « si-alors » et « sinon-alors » définies à partir d'une grande variété de types de données. Parmi les avantages de cette syntaxe se trouve la possibilité de spécifier quand le module doit être utilisé pour évaluer une situation d'alerte.

2.2.7 Utilisateurs concepteurs et consommateurs

Les systèmes d'alertes sont très courants dans les réseaux dont les utilisateurs *concepteurs* et *consommateurs* sont les administrateurs du réseau et la surveillance des dispositifs est en temps réel. Dans ces systèmes, les alertes sont utiles pour la détection d'intrus.

2.2.7.1 Notification contrôlée

Cisco Prime Infrastructure (Cisco, 2013) utilise des concepts tels qu'*événements* et *situations d'alerte*. Un événement est l'occurrence à un instant précis d'une condition par un dispositif dans le réseau.

Exemple

Le changement du port, la réinitialisation du dispositif, ou quand un dispositif devient inaccessible par la station centrale.

La condition définissant l'événement varie selon le type d'événements. Les types d'événements sont *prédéfinis* et couvrent de nombreuses catégories comme les points d'accès, les contrôleurs, le client, etc. Une *situation d'alerte* est définie à partir d'un événement. Néanmoins, la même situation d'alerte peut être définie plusieurs fois à partir d'événements différents. Les *alertes détectées* sont notifiées aux utilisateurs par mail mais elles sont aussi visibles dans la page de gestion d'alertes. Dans cette page, les utilisateurs peuvent *modifier* les événements et les alertes préconfigurées faisant partie du catalogue initial du système, ainsi qu'enrichir les catalogues avec des nouveaux événements et alertes. Une *sévérité* est attribuée à chaque événement et à chaque alerte. Un deuxième type d'alertes est appliqué à des dispositifs spécifiques et est défini à l'aide de métriques calculées à partir soit de la « santé du dispositif » (utilisation de CPU, de mémoire, ou la température), soit de la « santé de l'interface » (nombre de paquets sortants qui sont rejetés). Si la valeur de la métrique est égale ou supérieure à un *seuil* défini, une alerte est détectée.

Ce système offre une gestion d'alertes plus avancée que les systèmes vus précédemment. Une même situation d'alerte vérifiée par des événements différents n'est détectée et notifiée qu'une seule fois. Le cycle de vie d'une alerte détectée est fini lorsque l'alerte est *nettoyée*. Néanmoins, une alerte nettoyée peut être *relancée* si le même *événement* qui l'a causé se reproduit pendant une période de temps prédéfinie et paramétrable. Le nettoyage d'une alerte est fait automatiquement après le nettoyage de l'événement qui l'a causée. Le nettoyage manuel par un utilisateur d'une alerte mène aussi au nettoyage de l'événement. L'auto-nettoyage des alertes s'applique lorsque le dispositif qui a produit l'événement causant l'alerte produit lui-même l'événement qui le nettoie.

Exemple

L'événement déclenché par un dispositif défini comme le « passage d'accessible à inaccessible » est nettoyé lorsque le dispositif produit l'événement inverse défini comme le « passage d'inaccessible à accessible ».

2.2.7.2 Autogestion d'alertes détectées

(Hyperic, 2013) définit un outil de gestion du rendement qui offre une visibilité sur les applications et les infrastructures dans les environnements physiques, virtuels et le cloud. Il consolide la découverte, la surveillance, l'analyse et le contrôle des applications, du système et du réseau. Le système propose un module d'alertes qui déclenche une alerte pour une ressource lorsqu'une condition est vérifiée, notifie le personnel et les parties prenantes, exécute des opérations de contrôle de la ressource, permet le suivi de l'état de la résolution de problèmes révélés par les alertes et fait l'analyse des alertes et les actions sous-jacentes. Les alertes sont *paramétrées par les utilisateurs* et évaluées sur les *ressources*. Il y a deux façons de définir une alerte : soit à partir des métriques comme « disponibilité de l'application ≤ 99% », soit à partir des événements comme le « changement du fichier de configuration du serveur ». Parmi les *ressources* surveillées et utilisées pour la définition de situations d'alerte se trouvent les « plateformes », les « serveurs », les « services », ou les « applications » faisant partie de l'*inventaire* ainsi que des informations comme « Win32 » pour le type

de plateforme, ou le « serveur » web « Apache ». Les situations d'alerte peuvent être définies jusqu'avec *trois* conditions et détectées si toutes (opérateur logique et) ou au moins l'une d'entre elles (opérateur logique ou) est vérifiée. Lorsque l'utilisateur définit la situation d'alerte, il spécifie aussi sa *priorité* de l'alerte comme « faible », « moyenne », ou « élevée », les *actions-à-exécuter* ainsi que les *actions-récupératrices* en cas de vérification positive. Les actions-à-exécuter sont dépendantes du type de ressources et de la version de Hyperic.

Les actions-réparatrices permettent d'exécuter des actions qui déclenchent un type d'alertes présenté sous le nom d'« alertes réparatrices ». Si une alerte réparatrice est détectée, l'alerte qui a déclenché l'exécution de l'action réparatrice est marquée comme « réparée ». Les situations d'alerte définissant des actions réparatrices sont désactivées lorsqu'une instance est détectée et réactivées lorsque les instances sont réparées. Les utilisateurs peuvent activer ou désactiver les situations d'alerte manuellement ainsi que définir l'état d'une alerte « réparée » ou « non-réparée ». Les alertes sont notifiées par mail aux utilisateurs de Hyperic. Cependant, il est possible d'ajouter des adresses mails externes. Les alertes sont disponibles dans la page de gestion d'alertes « Alert Center ». Il est possible de filtrer par *état*, par type de ressources, par priorité, ou par date. Hyperic offre en plus un mécanisme de contrôle de sur-notification appelé *répétition de l'escalade*. Il s'agit de définir une période de temps (escalade) débutée lors d'une première notification de l'alerte. Pendant cette période, aucune notification d'alertes détectées instanciant la même situation d'alerte n'a lieu. La période peut être répétée jusqu'à ce que l'alerte détectée et notifiée initialement soit identifiée comme « réparée ». La notification d'alertes détectées pour une même situation d'alerte s'active une fois la période d'escalade dépassée.

2.2.8 Réduction de faux-positifs

La particularité des systèmes d'alertes dans le domaine des réseaux sont les *méta-alertes* élaborées à partir d'*alertes brutes* détectées majoritairement par des systèmes de détection d'intrusion (IDS). L'objectif est de réduire le grand nombre d'alertes générées en utilisant des dimensions de qualité. Les systèmes d'alertes déployés dans des réseaux utilisent des données de bas niveau afin d'évaluer les situations d'alerte que les administrateurs définissent à partir d'événements. Cependant, les caractéristiques de chaque intrusion (conditions) sont très variées et rendent très difficile la tâche de définition des situations d'alerte ce qui provoque de nombreuses fausses positifs, soit des alertes détectées par le système mais que les utilisateurs ne les considèrent pas comme telles.

2.2.8.1 Classification des séquences d'événements

(Jan, et al., 2009) propose un système d'aide à la décision qui permet de construire un modèle de classification d'alertes pour un réseau. Le système est divisé en trois processus. D'abord, le *prétraitement de l'alerte*, ensuite la *construction du modèle*, et enfin, le *raffinage des règles*. Le premier processus transforme les nombreuses alertes détectées dans le réseau afin de les rendre exploitables par les modèles construits par le deuxième processus. Ainsi le prétraitement détecte des

mêmes *séquences d'attaque* sur des cibles différentes. Les séquences d'attaques sont construites sur deux fenêtres temporelles configurables présentées comme la « fenêtre de lots » et la « fenêtre de courte durée ». La « fenêtre de lots » permet au système de récupérer toutes les attaques pendant une *période de temps* tandis que la fenêtre de courte durée la *segmente* en (sous)séquences qui respectent certaines politiques de partitionnement. La *construction du modèle* est un processus qui permet de définir des règles pour la classification des alertes telles que « normale », « intrusion », et « suspecte » à l'aide de techniques de *fouille de données*. Les règles sont définies à partir des *scores* calculés et des *seuils prédéfinis*. Si le score d'une séquence est supérieur au seuil alors l'alerte est classifiée. Les règles « normales » filtrent les fausses alertes. Les règles « d'intrusion » sont celles qui détectent des attaques *auparavant connues*. Si une alerte n'est pas filtrée avec les règles « normales », et n'est pas détectée avec les règles « d'intrusion », elle est donc considérée comme une alerte suspecte. Le processus de raffinage de règles permet au système de vérifier seulement un nombre défini de règles (200) afin de ne pas affecter la performance du système de détection. Ainsi, si une nouvelle règle est déduite et classée, elle remplace la règle utilisée le moins récemment.

2.2.8.2 Vulnérabilités contextuelles et indices de qualité

L'approche de (Njogu, et al., 2013) se concentre sur la qualité des alertes dans un réseau avec deux types d'alertes, les alertes brutes et les méta-alertes. Les alertes brutes sont celles qui sont détectées par des systèmes de détection d'intrus tandis que les méta-alertes sont celles qui sont éventuellement présentées aux utilisateurs. Le système utilise toutes les alertes brutes détectées par des systèmes de détection d'intrus. Un premier traitement ignore les alertes qui ne sont pas importantes et les données inutiles. Les *alertes prétraitées* sont ensuite comparées avec l'historique des méta-alertes détectées à partir des informations telles que l'adresse IP source, l'adresse IP de destination, ou les ports. Si la comparaison est satisfaisante, l'alerte est prête à être exploitée par le système de notification. Dans l'autre cas, l'alerte prétraitée est considérée comme suspecte et ensuite *comparée* avec une base de données de *vulnérabilités* présentes dans le réseau à partir des informations contextuelles. La base de données est remplie par un détecteur de vulnérabilités et seules les *vulnérabilités récentes* sont utilisées pour la comparaison. Le système utilise les vulnérabilités du réseau parce qu'elles sont la cible des attaques. La comparaison de l'alerte avec les vulnérabilités présentes dans le réseau calcule quatre dimensions de qualité à l'aide de métriques, la *pertinence* (importance) de l'alerte (0-9), la *sévérité* de l'alerte (1-3), la *fréquence* de l'alerte (0-1), et la *confiance* dans la source de l'alerte (0-6).

2.2.8.3 Valeurs linguistiques qui expriment la qualité

Les *alertes prétraitées* sont classifiées en utilisant les quatre métriques et des ensembles flous définis par des fonctions gaussiennes. Les métriques des alertes sont transformées en *valeurs linguistiques* qui expriment des dimensions de *qualité*. La *pertinence* exprime l'importance de l'alerte. Elle se calcule par rapport au nombre d'attributs appareillés de l'alerte avec les attributs de vulnérabilité présents dans le

réseau. Le score pour la *sévérité* de l'alerte se calcule par rapport à la priorité de l'alerte et la sévérité de la vulnérabilité qu'elle appareille. La *fréquence* exprime l'occurrence des alertes brutes détectées par une source pendant une période de temps et un seuil prédéfini. Si le nombre d'alertes brutes dépasse le seuil alors le système attribue un score de 1, sinon, 0. La *confiance de la source* exprime le degré de confiance que l'utilisateur a dans le dispositif qui détecte l'attaque/alerte brute. Les valeurs linguistiques pour la pertinence sont « basse » et « haute », pour la sévérité « basse », « moyenne », et « haute », pour la fréquence « basse » et « élevée », et pour la confiance de la source « basse » et « haute ». Les valeurs linguistiques sont utilisées pour classifier l'*alerte-prétraitée* dans l'une des onze classes.

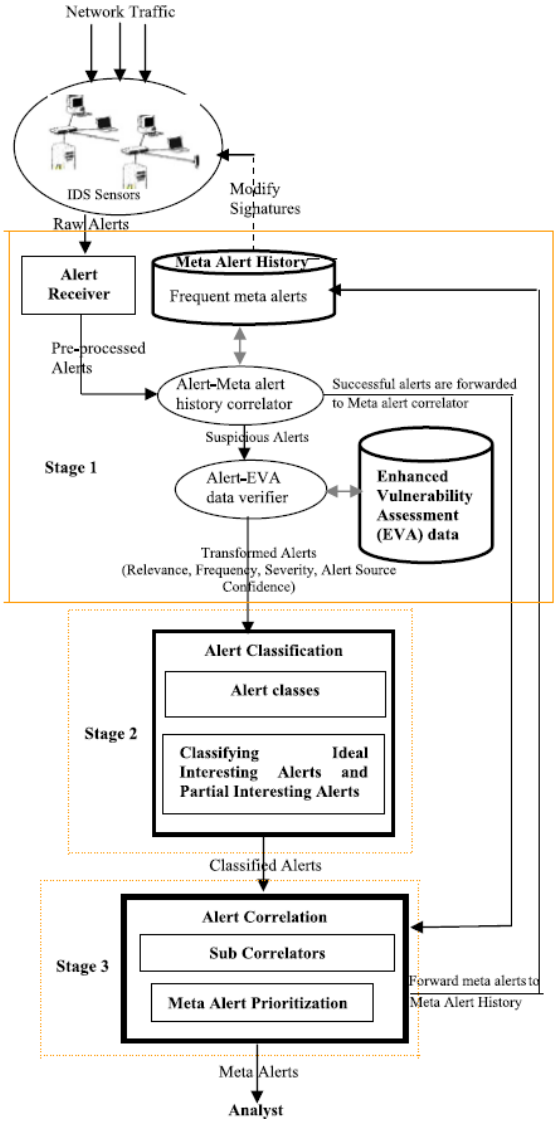


Figure 4. Architecture d'un système de détection de méta-alertes sur un réseau

Exemple

Une attaque « déni de service » avec des métriques de 9, 3, 1, et 6 pour la *pertinence*, la *sévérité*, la *fréquence*, et la *confiance* (de la source) se traduit dans les dimensions de qualité « haute », « élevée », « haute », et « haute » respectivement. Le système la classifie dans la première classe parmi les 11 classes *prédéfinies*.

Les règles de classification sont définies par des *experts* et des techniques de datamining dont l'objectif est d'obtenir la meilleure *exactitude*. Une alerte qui appartient à une classe hérite du score de la classe (celui de la classe 1 est égal à10) qui est utilisé pour classifier l'alerte prétraitée par niveau d'intérêt comme étant « intéressante » (≥7) ou « partiellement intéressante » (<7). Une dernière étape, consiste à faire une corrélation des alertes afin de réduire les alertes redondantes ou isolées. Le système crée donc des méta-alertes à partir des alertes-prétraitées. Une méta-alerte est la fusion d'alertes-prétraitées représentant le même type d'attaque, appartenant à la même classe, avec la même source (ip:port), la même destination (ip:port), et appartenant à une même période de temps prédéfinie. Les méta-alertes détectées sont *envoyées* à l'administrateur du réseau. La Figure 4 montre l'architecture du système d'alertes.

2.2.8.4 Score des alertes

Une approche avec une architecture similaire est présentée dans le travail de (Alsubhi, et al., 2012). Dans ce cas, les auteurs proposent sept métriques : l'*applicabilité*, l'*importance de la victime*, la *sévérité de l'attaque*, la *relation entre les alertes,* le *statut du capteur*, le *placement du capteur*, et la *vulnérabilité du service*. Seules les quatre premières sont utilisées pour calculer le score de l'alerte détectée. Dans ce travail, un capteur est un dispositif capable de détecter des attaques. L'applicabilité d'une alerte exprime si l'attaque/alerte détectée par un dispositif s'applique à l'environnement courant. Si l'attaque cible un service avec un *patch* pour la vulnérabilité ciblée, alors l'*applicabilité* est égale à 0 sinon 1. L'*importance de la victime* exprime si l'attaque cible une machine significative dans l'environnement courant. Elle se calcule par rapport à l'importance des services et des comptes dans la machine. Le *statut du capteur* exprime l'efficacité du capteur lors de la détection d'attaques/alertes. Il se calcule comme la *probabilité conditionnelle* de détecter une alerte étant donnée une attaque. Ce calcul nécessite que les alertes détectées par un capteur soient étiquetées comme des vrais-positifs, vrais-négatifs, faux-positifs, ou faux-négatifs. Le *placement du capteur* exprime l'importance des machines qui se trouvent dans le sous-réseau qu'il protège. Mais le travail ne spécifie pas comment il est calculé. La *sévérité de l'attaque* exprime le niveau de risque de la vulnérabilité ciblée. Il est calculé par la sévérité de l'attaque dans des bases d'attaques existantes (MITRE, CVE, etc.) Différentes bases donnent une sévérité différente pour une même attaque. Les auteurs calculent donc la sévérité de l'attaque à partir de la sévérité attribuée dans les bases d'attaques et la confiance faite à la base. La *vulnérabilité du service* est calculée par rapport aux informations récupérées des bases d'attaques comme la stabilité du service dans le passé, les patchs installés, etc. Finalement, la relation entre les alertes exprime si l'alerte détectée a une relation avec des alertes détectées dans le passé qui ciblent la même machine.

2.2.8.5 Participation en continu des utilisateurs

Le travail de (Bakar & Belaton, 2005) propose un système pour la détection d'alertes de qualité. Les auteurs utilisent quatre dimensions de qualité et s'en servent pour calculer la qualité totale de l'alerte. La *justesse* indique si la cible d'une attaque

est active (0-1). L'*exactitude* d'une alerte exprime si l'attaque concerne le système d'exploitation, les services, et les applications. La *fiabilité* d'une alerte indique la fiabilité du dispositif de détection (0-1). C'est une valeur qui doit être mise à jour fréquemment par rapport aux alertes détectées par le système. Néanmoins, il n'est pas spécifié comment calculer cette valeur. Finalement, la *sensitivité* d'alerte exprime la sensibilité du dispositif qui la détecte (0-1). Cette valeur doit être mise à jour fréquemment par rapport aux alertes qui n'ont pas été détectées par le système. La qualité totale de l'alerte se calcule comme la somme des produits des scores calculés précédemment par l'importance de la dimension de qualité vis-à-vis de la situation d'alerte.

2.2.8.6 Alertes agrégées

Nous terminons cette section avec le travail de (Chandrasekaran, et al., 2006) qui se concentre sur des hyper-alertes, une agrégation d'alertes brutes afin de faciliter l'analyse des attaques. Les alertes sont agrégées grâce à un réseau de causalité (bayésienne) qui exprime la probabilité qu'un nœud « A » soit infecté si le nœud « B » l'est déjà. Les nœuds représentent donc des machines et les arcs représentent la relation entre les machines. Chaque lien a un poids qui exprime la probabilité conditionnelle. Elle est calculée à partir de trois dimensions de qualité concernant les deux machines : l'*écart entre le nombre* d'alertes, la *pertinence* des alertes, et la *variété* des alertes. L'*écart entre le nombre* d'alertes se calcule par rapport au nombre habituel (la moyenne) d'alertes détectées par les machines. La *pertinence* des alertes exprime si l'attaque peut atteindre la machine destinée. Finalement, la *variété* des alertes indique la diversité des alertes qui sont détectées dans les deux machines. Elle est calculée à partir du nombre d'alertes qui ne sont pas détectées par les deux machines et la fonction sigmoïde pour normaliser la valeur.

2.3 Catégorisation des systèmes d'alertes

Nous catégorisons les systèmes d'alertes étudiés selon quatre critères : la définition des situations d'alerte, la notification d'alertes, les architectures, et les dimensions de qualité introduites. Le choix des critères est indépendant du domaine d'application ; il nous a servi à fonder notre proposition TEMPAS (cf. chapitre 4).

2.3.1 Définition de situations d'alerte

Les systèmes d'alertes étudiés permettent de définir des situations d'alerte qu'ils sont capables d'évaluer afin de déterminer si elles se produisent ou non. Pour l'évaluation des situations d'alerte, le système utilise des connaissances et un moteur d'inférence.

Exemple

Pour déterminer si la tendance de la température est à la hausse, le système utilise des informations qui lui permettent de construire une série chronologique sur laquelle il va détecter des tendances.

Parmi les modèles utilisés par les moteurs d'inférence, on trouve les réseaux de causalité (cause-effet), les réseaux bayésiens (probabilité conditionnelle), et les systèmes de règles (if-then). Le choix dépend de l'expressivité des situations d'alerte. Des situations d'alerte complexes peuvent nécessiter des modèles d'inférence complexes et demander des compétences spécifiques afin de les définir. Ainsi, des modèles simples, comme les systèmes de règles ou les réseaux de causalité, sont utilisés de préférence lorsque l'utilisateur final est néophyte.

Tableau 1. Catégorisation par expressivité

Systèmes d'alertes	Expressivité lors de la définition de situations d'alerte						
				Multi-conditions			
	Utilisateur dédié	Seuils	Supervisé	Disjonction	Conjonction	Addition	Contraintes temporelles
(Bay Alarm Medical, 2014), (LifeFone, 2014), (Reach Alert, 2014), (ReachPlus , 2014), (Be Alert, 2014), (ASTER, 2010)	X						
(Nihon Kohden Corporation, 2014)		X					
(Philips, 2014), (Jan, et al., 2009), (Chandrasekaran, et al., 2006)			X				
(Cisco, 2013)		X		X			
(Hyperic, 2013)		X		X	X		
HELP-MLM, Asbru				X	X		X
(Njogu, et al., 2013), (Alsubhi, et al., 2012), (Bakar & Belaton, 2005)						X	

La catégorisation par expressivité lors de la définition des situations d'alerte est synthétisée dans le Tableau 1. Nous utilisons le terme de « utilisateur dédié » si c'est un utilisateur et non le système qui décide s'il s'agit d'une situation d'alerte à un instant donné. La catégorie « seuils » regroupe les systèmes où l'utilisateur peut fixer les seuils à dépasser, ou non, afin de déclencher des alertes. La catégorie « supervisé » contient les systèmes où la situation d'alerte n'est pas définie exactement, mais la participation des utilisateurs permet d'identifier les instants où des alertes doivent être détectées. La catégorie « multi-conditions » regroupe les systèmes où les utilisateurs définissent des situations d'alerte à partir des conditions construites à partir d'événements. Nous différencions trois sous-catégories concernant le rapport entre la satisfaction des conditions et la conclusion d'une alerte : « disjonction » si la satisfaction d'une seule condition est une raison suffisante pour conclure à une alerte ; « conjonction » si toutes les conditions doivent être satisfaites pour conclure à une alerte ; et « addition » si une alerte doit être lancée même si toutes les conditions ne

sont pas satisfaites. La dernière catégorie regroupe les systèmes où il est possible de définir les situations d'alerte avec des « contraintes temporelles ».

2.3.2 Notification d'alertes

Par notification d'alertes, nous entendons les interactions entre le système et les utilisateurs lorsqu'une alerte est détectée. La notification se manifeste de façons diverses. Dans les cas les plus courants, la détection d'une alerte mène au déclenchement d'une alarme sonore et à la manifestation visuelle sur des écrans (tv, ordinateurs, portable, etc.). En parallèle, des actions peuvent avoir lieu suite à la détection d'une alerte.

Exemple

Activer l'arrosage d'eau lorsque le dioxyde de carbone, le monoxyde de carbone et la température dépassent des seuils prédéfinis dans une chambre (incendie) et notifier les pompiers et les habitants de la maison.

Tableau 2. Catégorisation par notification d'alertes

Systèmes d'alertes	Notification d'alertes				
			Contrôlée		
	Infrastructure publique	Signaux	Successive	Concurrente	App. cliente riche
(Bay Alarm Medical, 2014), (LifeFone, 2014), (Be Alert, 2014), (ASTER, 2010)	X				
(Reach Alert, 2014), (ReachPlus , 2014),	X				X
(Nihon Kohden Corporation, 2014)	X	X	X		
(Philips, 2014)		X			X
(Cisco, 2013), (Hyperic, 2013)	X		X		X
HELP-MLM					X
(Jan, et al., 2009), (Chandrasekaran, et al., 2006)				X	
(Njogu, et al., 2013), (Alsubhi, et al., 2012)			X		

La catégorisation par notification d'alertes concerne directement la communication des alertes détectées aux utilisateurs. Elle est synthétisée dans le Tableau 2. Par « infrastructure publique» nous entendons l'utilisation de l'internet, du réseau mobile, de la téléphonie afin d'adresser les alertes aux utilisateurs. La catégorie « signaux» concerne les systèmes où la notification d'alertes est faite par des signaux sonores ou visuels. Si le système fournit un mécanisme pour réduire le nombre d'alertes notifiées aux utilisateurs, il se trouve dans la catégorie « contrôlée ». Nous différencions deux types de contrôle de notification : celui qui cherche à réduire les notifications à partir des alertes détectées de façon « successive » et un deuxième qui cherche à réduire les notifications des alertes « concurrentes ». La catégorie

« application client riche » regroupe les systèmes avec une application cliente qui facilite la gestion d'alertes détectées puis notifiées.

2.3.3 Architecture des systèmes d'alertes

Le Tableau 3 synthétise la catégorisation par architecture au sens large. Si les données sont récupérées par des « dispositifs » spécifiques fournis par le système d'alertes. Les systèmes dans la catégorie « temps réel » sont ceux où le temps écoulé entre l'occurrence de la situation d'alerte et la notification aux utilisateurs est extrêmement court. La catégorie « scalabilité » regroupe les systèmes dont la notification peut se faire vers un nombre élevé d'utilisateurs. La catégorie « type d'évaluation » exprime quand le système doit évaluer une situation d'alerte : la catégorie « événementiel » concerne des événements spécifiques qui déclenchent l'évaluation des situations d'alerte, la catégorie « périodique », quant à elle, concerne l'évaluation périodique des situations d'alerte.

Tableau 3. Classification par architecture

Systèmes d'alertes	Architecture du système				
				Type d'évaluation	
	Dispositifs	Temps réel	Scalabilité	Evénementielle	Périodique
(Bay Alarm Medical, 2014), (LifeFone, 2014)	X	X			
(Reach Alert, 2014), (ReachPlus , 2014), (Be Alert, 2014), (ASTER, 2010)		X	X		
(Nihon Kohden Corporation, 2014)	X	X		X	
(Philips, 2014), (Cisco, 2013)	X	X		X	
(Hyperic, 2013), HELP-MLM, (Jan, et al., 2009), (Njogu, et al., 2013), (Alsubhi, et al., 2012), (Bakar & Belaton, 2005), (Chandrasekaran, et al., 2006)		X		X	

2.3.4 Dimension de qualité

Concernant les systèmes d'alertes, nous pouvons différencier deux groupes de dimensions de qualité, ceux qui concernent la qualité du système, et ceux qui concernent la qualité des alertes. Les dimensions de qualité du système font partie des besoins non fonctionnels, cependant parvenir à ces dimensions de qualité, nécessite souvent des topologies et des architectures spécifiques.

Exemple

La scalabilité consiste à utiliser de plus en plus de sources de données pour la détection d'alertes.

Le temps de réponse consiste à détecter les alertes en temps réel ou proche du temps réel

L'accessibilité consiste à déterminer une disponibilité des alertes de 24h/7

L'adaptabilité est définie en temps réel.

Le deuxième groupe de dimensions de qualité concerne les données utilisées pour évaluer les alertes, les alertes détectées, ainsi que le modèle utilisé pour l'évaluation des situations d'alerte.

Exemple

Quelle est la confiance dans les données utilisées ou dans la source des données ?

Les alertes détectées sont-elles interprétables par les utilisateurs ?

Quelle est la précision de la situation d'alerte et le ratio des vrais-positifs ?

Tableau 4. Classification des dimensions de qualité

Systèmes d'alertes	Dimensions de qualité		
	Système	Classification	Score
(Bay Alarm Medical, 2014), (LifeFone, 2014), (Reach Alert, 2014), (ReachPlus , 2014), (Be Alert, 2014), (ASTER, 2010), (Nihon Kohden Corporation, 2014), (Cisco, 2013), HELP-MLM, Asbru	Non exprimée		
(Philips, 2014), (Jan, et al., 2009)	Précision		
(Njogu, et al., 2013)	Exactitude	Pertinence Sévérité Fréquence Confiance	Oui
(Alsubhi, et al., 2012)		Applicabilité Importance Sévérité Relation	Oui
(Bakar & Belaton, 2005)	Fiabilité Sensitivité	Justesse Exactitude	Oui
(Chandrasekaran, et al., 2006)		Ecart Pertinence Variété	Oui

Le Tableau 4 catégorise les systèmes étudiés par rapport aux dimensions de qualité qu'ils introduisent. Les dimensions de qualité concernant le « système » sont obtenues après avoir testé le système d'alertes avec des entrées dont la sortie est connue à l'avance. La « précision » mesure la capacité de prédire des alertes tandis

que l'« exactitude » exprime que les bonnes alertes sont détectées et que des alertes n'ont pas été manquées. Les dimensions concernant la « classification » sont celles qui se calculent à l'instant de l'évaluation d'une situation d'alerte. Finalement d'autres dimensions de qualité sont calculées à partir des ensembles de « classification » et permettent de calculer un « score » sur les alertes afin de les prioriser ou de les considérer comme telles. L'interprétation de chacune des dimensions de qualité est propre au système d'alertes.

2.3.5 Constat

Le domaine de la santé est particulier car la notion d'alerte y est, à juste titre, souvent associée. Il est indispensable de lancer les alertes les plus sûres et notamment d'éliminer les faux-négatifs qui peuvent avoir des conséquences graves.

Exemple

Il est préférable de sur-notifier une alerte quand celle-ci engage la vie du patient.

Cette sur-notification (overalerting) permet aux parties concernées d'éviter les risques légaux d'ignorer une alerte potentielle (Kesselheim, et al., 2011). Les utilisateurs connaissent l'impact des faux-négatifs mais se lassent dès qu'il y a sur notification de faux-positifs.

Maria Cvach fait une analyse sur la perte d'intérêt des utilisateurs dans les systèmes d'alertes du domaine de la santé (Cvach, 2012). L'analyse est motivée par fait que les systèmes d'alertes médicaux sont largement incriminés dans les accidents de santé.

Exemple

La notification des nombreuses alertes provoque l'interruption des tâches courantes des membres du corps médical et en même temps peuvent mener à des erreurs par omission ou distraction.

Plus de soixante-dix travaux sur les alertes médicales publiés entre 2000 et le 2011 mettent en avant le haut pourcentage (80%-99%) d'alertes concernant des faux-positifs ou des alertes inutiles (Atzema, et al., 2006). Une alerte inutile est une vraie alerte qui n'ajoute pas de valeur lorsqu'elle est notifiée. Parmi les effets des alertes sur le corps médical, on trouve : la perte d'intérêt et de confiance dans les alertes, ainsi que la fatigue puisque la notification des alertes est souvent accompagnée d'une alarme sonore. Ces éléments conduisent parfois à manquer les vraies alertes et mettre en danger la vie des patients.

Parmi les solutions pour améliorer l'acceptation des systèmes d'alertes médicales, se trouvent des alertes plus élaborées comme celles d' (Iskio, et al., 2006) concernant la prescription médicamenteuse. Les auteurs proposent une base de connaissances qui permet de détecter des alertes plus riches à partir de la prescription des médicaments similaires, des médicaments avec interactions, des médicaments dont l'effet est inutile sur le patient selon les derniers résultats du laboratoire, des

médicaments contradictoires avec le profil pathologique du patient, et finalement, des médicaments à éviter sur des patientes enceintes. En total, 67% des alertes détectées ont été validées puisque la prescription qui a déclenché l'alerte a été annulée ou modifiée. Dans le même domaine, (Beuscart, et al., 2009)introduisent un Framework pour la représentation et gestion des signaux concernant des événements indésirables. Des technique de fouilles de données ont permis d'établir 600 règles de décision parmi lesquelles 80 ont été validées qualitativement par des experts. (Phansalkar, et al., 2010) s'intéressent à une meilleure notification d'alertes. Les auteurs proposent une dizaine de recommandations concernant les systèmes d'alertes. Parmi elles : prioriser les alertes, partager des informations avec l'alerte de façon à suivre le modèle mental de l'utilisateur, minimiser les alertes qui nécessitent des accusés de réception, etc. Dans le même esprit, (van der Sijs, et al., 2006) liste vingt-quatre facteurs à prendre en compte pour fournir des alertes utiles comme éviter de demander plus de données en entrée, ou éviter les alertes répétées. En général, les systèmes d'alertes doivent être configurables (surtout en post-installation), flexibles, adaptables, et faciles à utiliser puisque tout environnement clinique est unique (Gee & Moorman, 2011), (Krall & Sittig, 2002), (Wyckoff, 2009). Il est souhaitable de travailler avec des modèles plus élaborés pour mieux définir des situations d'alerte et possiblement réduire des faux-positifs (Borowski, et al., 2011) (King, et al., 2012).

2.4 Synthèse

Nous avons étudié des systèmes dont la notion d'alerte est définie explicitement. Dans le cas le plus simple, les alertes sont détectées par des humains et les systèmes d'alertes sont responsables de leur routage. Dans les cas plus élaborés, les systèmes permettent de définir des situations d'alerte qu'ils sont capables d'évaluer. Lorsqu'elles sont définies à partir des plusieurs conditions non pré-paramétrées, des syntaxes et des sémantiques complexes s'imposent et demandent des compétences spécifiques, dans la plupart des cas, différentes de celles des utilisateurs finaux. Les systèmes de détection d'intrus ne permettent pas aux utilisateurs de définir explicitement des situations d'alerte puisque celles-ci sont abstraites par les systèmes eux-mêmes.

En général, il y a deux types d'alertes. Les alertes brutes et les méta-alertes. Les alertes brutes sont définies souvent à l'aide d'événements primitifs ou d'événements conditionnels. L'évaluation des situations d'alerte de ce type produit des gros volumes d'alertes détectées et parmi elles, de nombreux faux-positifs et des alertes inutiles. Les méta-alertes sont définies à partir d'un nombre plus élevé d'informations, comme des séquences d'alertes brutes similaires, ou l'occurrence d'événements dans un certain ordre. Les situations d'alerte de ce type génèrent un nombre inférieur d'alertes détectées lors de son évaluation. Toutefois, des compétences en mathématique ou informatique sont requises pour définir ce type de situations d'alerte. Bien que les méta-alertes réduisent le nombre d'alertes brutes à notifier, nous n'avons pas trouvé de mécanismes pour réduire la notification des méta-alertes générées.

Les architectures orientées événements facilitent la tâche d'évaluation des situations d'alerte. La situation d'alerte elle-même est définie à partir d'un événement (comme dans les systèmes de détection d'intrus), ou à partir des variables surveillées (comme dans l'Arden Syntax).

Concernant la qualité des alertes, nous supposons que le but final est de réduire les faux-positifs et/ou les faux-négatifs. Il s'agit d'améliorer l'exactitude ou la précision du système d'alertes lors de la classification d'une situation d'alerte comme telle. Toutefois, ce type de qualité se calcule après la validation des alertes détectées par l'utilisateur.

Les systèmes de détection d'intrus utilisent des dimensions de qualité calculées à partir des métriques. La confiance et la sévérité d'une alerte sont souvent calculées une seule fois. Par contre, d'autres métriques se calculent à partir d'un modèle fixe qui ne laisse pas la place à d'autres informations pour le calcul de la qualité. Ajouter une nouvelle mesure de similarité pour améliorer la métrique qui exprime la pertinence de l'alerte brute affecte tout le modèle de classification pour générer une méta-alerte. Toutefois, les dimensions de qualité utilisées pour créer les méta-alertes ne sont pas associées à la méta-alerte qui est transmise aux utilisateurs. Cela se traduit comme une perte d'information qui peut-être précieuse pour les consommateurs des alertes.

Exemple

La métrique qui exprime la pertinence d'une alerte se calcule avec neuf mesures de similarité prédéfinies.

Nous avons présenté la situation actuelle des systèmes d'alertes. Ils sont peu acceptés (dans le domaine médical) puisque ils ne satisfont pas les besoins des utilisateurs (Wang & Strong, 1996) (Strong, et al., 1997). Concernant la qualité du système, la détection en temps réel est bien prise en compte, mais ce n'est pas le cas pour l'adaptation en temps réel qui peut comprendre l'ajout de nouvelles conditions d'activation ou la possibilité de paramétrer des préférences.

3 Modélisation des alertes

3.1 Introduction 47
3.2 Modèles d'évaluation 47
3.3 Interprétation du temps 54
3.4 Les séries chronologiques 62
3.5 Contexte 68
3.6 Approche opérationnelle du traitement des données 71
3.7 Synthèse 75
3.8 Positionnement 76

3.1 Introduction

Cette deuxième partie de l'état de l'art est divisée en cinq parties. La section 3.2, présente des modèles utilisés par des systèmes d'alertes afin de déterminer qu'une situation d'alerte se produit. Nous dédions ensuite deux sections à l'aspect temporel : la section 3.2.4 où nous présentons l'interprétation du temps dans les systèmes d'information et dans les modèles afin d'améliorer le raisonnement des moteurs d'alerte ; nous étudions les séries chronologiques et leurs différentes applications dans la section 3.4. Nous présentons des travaux sur le contexte dans la section 3.5 et une approche opérationnelle du traitement des données dans la section 3.6. Nous terminons avec une synthèse et un positionnement de nos travaux dans les sections 3.7 et 3.8.

3.2 Modèles d'évaluation

Nous avons présenté des systèmes d'alertes existants et brièvement les modèles utilisés pour déterminer une situation d'alerte. La plupart d'entre eux utilisent des modèles basés sur des règles logiques. Dans cette section, nous étudions quatre approches. Dans les trois premiers, sections 3.2.1, 3.2.2, et 3.2.3 les relations entre les facteurs intervenant dans une situation d'alerte sont faciles à établir. D'autres approches logiques où le système détermine avec certitude si une situation d'alerte se produit sont présentées dans la section 3.2.1. Dans la section 3.2.2 nous étudions des approches probabilistes qui permettent au système de calculer la probabilité qu'une situation d'alerte se produise. En revanche, avec la logique floue présenté dans la section 3.2.3, il possible de déterminer qu'une situation d'alerte se produit partielle ou complètement. Finalement, les techniques de fouille de données et d'apprentissage automatique, que nous étudions dans la section 3.2.4, sont utilisées lorsqu'il est difficile de modeler la corrélation entre les facteurs intervenant dans les situations d'alerte. Les modèles d'évaluation des systèmes d'alertes peuvent utiliser plus d'une des techniques présentées dans cette section.

Les systèmes d'alertes peuvent être considérés comme des systèmes d'aide à la décision où le lien entre les observations et les connaissances est établi afin d'aider les utilisateurs à prendre des décisions qui améliorent les services fournis. Les systèmes d'alertes prennent différents types de décision et jouent différents rôles dans les systèmes d'information (Musen, et al., 2006). Les systèmes d'aide à la décision ont généralement accès à une base de connaissances, un moteur d'inférence pour arriver à des conclusions et un mécanisme de communication avec les utilisateurs. La base de connaissances contient les faits et les informations nécessaires pour que le moteur d'inférence puisse déduire des nouvelles connaissances qui peuvent également être stockées afin de les utiliser a posteriori.

Exemple

Aboutir à un diagnostic peut nécessiter l'étude des diagnostics antérieurs.

3.2.1 Approche logique

Le travail de (Console & Torasso, 1988) propose une approche logique pour raisonner avec des modèles de causalité incomplets ou incertains utilisés pour faire du diagnostic. Les connaissances sont modelées à l'aide d'un *réseau causal* qui permet de construire des cycles afin de modéliser correctement le monde réel. Les nœuds expriment la *cause-initiale*, des *hypothèses*, des *états*, des *actions*, et des découvertes. Le type d'arc nommé « définit-comme » fait le lien entre un ensemble d'états et une hypothèse. Le type d'arc « cause-effet » fait le lien entre les nœuds exprimant des états. Ce type d'arc est associé à un type de nœud action qui exprime comment l'effet dépend de la cause. Finalement le type d'arc nommé « a-comme-manifestation » fait le lien entre un nœud « état » et un nœud « découverte ». Un arc est considéré comme étant « obligatoire » ou bien « possible » afin de pouvoir travailler avec des informations incomplètes ou incertaines. Un arc de type « définit-comme » est toujours « obligatoire ». Il existe deux manières possibles d'arriver à une conclusion : soit toutes les manifestations sont présentes à un instant donné et permettent d'arriver à l'*hypothèse* ; soit le système arrive à une *contradiction* logique. Ce raisonnement logique permet de travailler avec des arcs étiquetés comme par exemple « possible » qui permet d'exprimer l'incertitude à partir de *suppositions*. Ainsi un fait est supposé être vrai sans certitude qu'il le soit. Le modèle détermine que le *monde* se trouve à un instant donné dans l'un des trois états : « Confirmé » si les affirmations ont été *prouvées*, « Hypothétique » s'il y a eu des *suppositions* et « Inconsistant » si le système arrive à une *contradiction*.

Exemple

Une situation d'alerte peut être associée à un diagnostic spécifique. Si le moteur détermine que le monde se trouve dans l'état « confirmé » ou « hypothétique » une alerte est détectée et notifiée.

La logique classique est nécessaire mais non suffisante, car nous devons prendre en compte d'autres facettes comme l'ambigüité et la temporalité.

3.2.2 Approche probabiliste

L'approche probabiliste est construite à partir de relations cause/effet et de la probabilité conditionnelle correspondante. La méthode la plus populaire pour travailler, représenter et raisonner avec des informations probabilistes sont les réseaux bayésiens. Les *réseaux bayésiens* sont souvent représentés comme des graphes dont les *liens* expriment la *probabilité conditionnelle* : soit la probabilité qu'un événement destinataire du lien se produise étant donné qu'un autre événement, source du lien, a déjà eu lieu. Une approche probabiliste est présentée par (Ngo, et al., 1997). Les auteurs proposent un langage pour représenter des connaissances temporelles et probabilistes sensibles au contexte. Le processus d'inférence utilise les *contraintes contextuelles* pour se concentrer uniquement sur les parties pertinentes de la connaissance probabiliste. Leur travail est une extension des réseaux bayésiens avec des contraintes contextuelles. Les contraintes contextuelles sont des conditions logiques qui déterminent l'*applicabilité* d'une relation probabiliste exprimée par un lien à partir de connaissances déterministes. Les auteurs proposent un algorithme qui, à partir d'une requête temporelle, construit un réseau bayésien en prenant en compte les informations contextuelles. Les requêtes temporelles sont exprimées à partir d'un nombre spécifique des points avant ou après un point fixe. Dans le cas de *prédiction,* la requête exprime le futur.

Exemple

La situation d'alerte est associée à l'arythmie. Quelle est la probabilité que le patient souffre d'arythmie dans les deux ans à venir ? Si la probabilité est supérieure à 90%, une alerte est détectée et notifiée.

Cette approche probabiliste doit être associée à d'autres notions comme l'incertain et la chronologie de faits.

3.2.3 Logique floue et valeurs linguistiques

Les modèles mathématiques permettent de raisonner et donner des réponses concernant l'univers du discours.

Exemple

Est-ce que le nombre « 75 » est supérieur au nombre « 60 » ?

Le modèle mathématique classique transforme la question en une condition relationnelle telle que « 75 > 60 ?». La théorie des ensembles de Cantor permet de définir des ensembles afin de raisonner à partir de la notion d'appartenance. Un ensemble contient donc tous les éléments de l'univers qui satisfont une condition.

Exemple

L'ensemble « A » comprend tous les nombres supérieurs à 60.
Le nombre 75 appartient-il à l'ensemble « A » ?

La réponse est booléenne. Si 75 appartient à l'ensemble alors « oui », sinon « non ». Cependant, cette modélisation est moins robuste lorsque la certitude n'est pas systématique et *l'ambigüité* présente.

Exemple

Une personne âgée de 18 ans est-elle « jeune » ?
75 est-il un nombre beaucoup plus grand que 60 ?

La *logique floue* est une solution pour raisonner de façon similaire à un humain lorsque *l'ambigüité* est présente (Zadeh, 1965). Un *ensemble flou* se caractérise par sa *fonction d'appartenance* qui calcule le *degré d'appartenance* d'un élément de l'univers à l'ensemble flou. Le *degré d'appartenance* d'un élément à un ensemble flou est un nombre réel qui appartient à l'intervalle [0, 1] tandis que dans la théorie des ensembles classique, l'appartenance d'un élément est 0 ou 1. Toutefois, la logique floue définit les opérateurs tels que l'union, l'intersection, l'égalité, l'inclusion, ou le complément, et leurs respectives propriétés d'associativité ou commutativité, déjà présents dans la théorie des ensembles classique.

Exemple

Si le degré d'appartenance du nombre 18 à l'ensemble « jeune » est de 0.9, une personne âgée de 18 ans est-elle « jeune » ? La réponse peut-être similaire à « oui, mais pas complètement ».

3.2.3.1 Valeurs linguistiques

La logique floue a été utilisée pour rapprocher le monde des mathématiques de celui du langage naturel afin de concevoir des applications et des résultats plus compréhensibles par les humains grâce à l'utilisation de variables linguistiques (Bai, et al., 2007). Une variable linguistique utilise, comme son nom l'indique, des *valeurs linguistiques*.

Exemple

La variable « taille » dans sa version numérique prend des valeurs telles que 120 cm, 165 cm, 178 cm, etc. tandis que la version linguistique utilise les valeurs « petit », « normal », « grand ».

Une variable linguistique peut se formaliser comme il suit. $V: < U, L, \mu >$, où V est la variable, U est l'univers du discours, L est l'ensemble de valeurs linguistiques, et μ l'ensemble des fonctions d'appartenance caractéristiques de chaque ensemble flou associé à chaque valeur linguistique. La formule (1) montre une formalisation possible de la variable linguistique taille.

$$Taille: < [2 - 220], \{petit, normal, grand\}, \{\mu_{petit}, \mu_{normal}, \mu_{grand}\} > \quad \textbf{(1)}$$

La Figure 5 montre trois exemples d'ensembles flous associés aux valeurs linguistiques « petit », « normal », et « grand » représentant la « taille », et le degré d'appartenance de la valeur « 175 » aux ensembles flous associés aux valeurs linguistiques « normal » et « grand ». Notons que les ensembles flous ainsi que les

valeurs linguistiques sont dépendants du point de vue de chacun, et de sa perception de la « taille ».

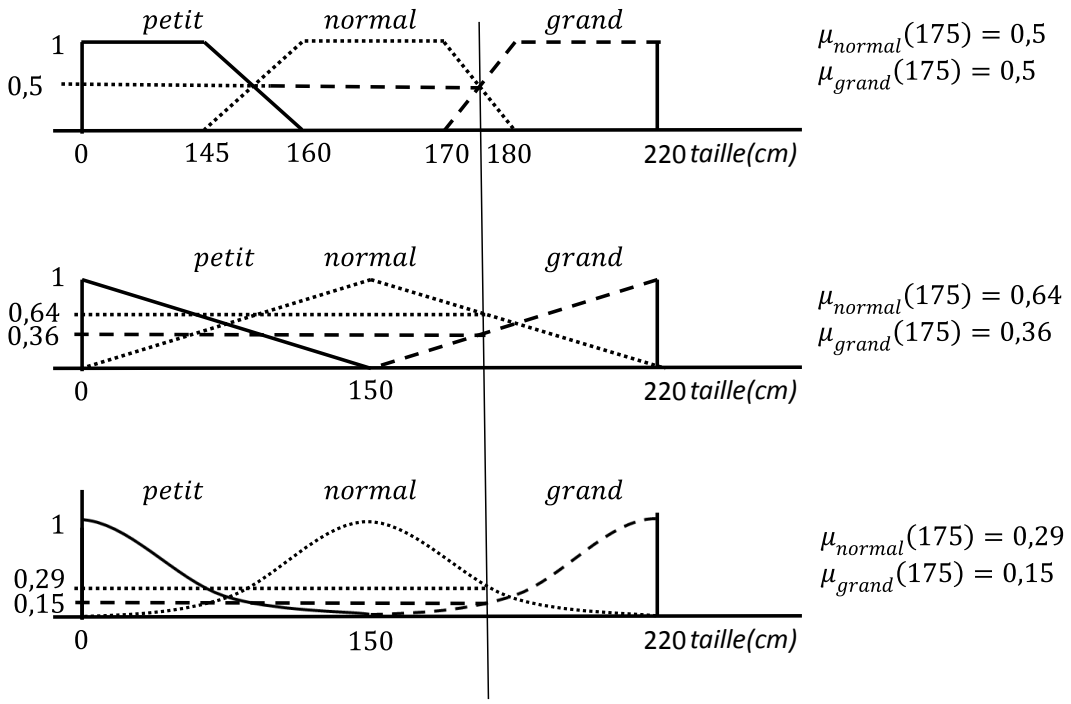


Figure 5. Exemples d'ensembles flous pour la variable taille

3.2.3.2 Raisonnement

Les systèmes de raisonnement qui utilisent la logique floue passent généralement par trois processus. Le premier processus transforme des valeurs numériques en valeur linguistiques. Le deuxième processus applique des règles qui, dans la plupart des cas, sont définies à partir d'une table de vérité construite à partir des valeurs linguistiques. Appliquer une règle retourne une valeur linguistique qui est transformée en une valeur numérique par le dernier processus. Dans la plupart des cas, la méthode utilisée est celle du *centre de gravité* (Bai & Wang, 2006). La valeur numérique obtenue est ensuite exploitable par une machine.

Tableau 5. Transformation en valeurs linguistiques

Variable	Valeur numérique	Valeur linguistique	Degré d'appartenance
Taille	175	Normal	0,5
Taille	175	Grand	0,5
Poids	80	Normal	0,3
Poids	80	Obèse	0,7

Exemple

Imaginons un système qui programme automatiquement un « nombre de rendez-vous » entre une personne et un diététicien. Le patient « Martin » mesure « 175 cm » et pèse « 80 Kg ». Le Tableau 5 résume la transformation de « 175 cm » en deux valeurs possibles linguistiques « normal » et « grand », et « 80 Kg » en « normal » et « obèse ». Le Tableau 6 liste les règles définies par des experts. Parmi elles, « aucun » « rdv » n'est programmé si la « taille » et le « poids » de la personne sont « normaux », tandis que si la « taille » est « normal » et le « poids » est « obèse », « beaucoup » de « rdv » doivent être programmés. Le Tableau 7 affiche les centres de gravité des ensembles flous associés aux valeurs linguistiques « aucun », « peu », « beaucoup » qui représentent le « nombre de RDV » à programmer. Le Tableau 8 utilise les résultats des tableaux précédents pour calculer que « 2 » ou « 3 » RDV doivent être programmés pour le patient « Martin ».

Tableau 6. Exemple des règles pour programmer des rendez-vous

Taille	Poids	Rendez-vous
…	…	…
Normal	Normal	Aucun
Normal	Obèse	Beaucoup
Grand	Normal	Aucun
Grand	Obèse	Peu

Tableau 7. Valeurs linguistiques et centres de gravité pour la variable RDV

Valeur linguistique	Plage de valeurs numériques	Centre de gravité
Aucun	[0-2]	1
Peu	[1-3]	2
Beaucoup	[2-8]	5

Tableau 8. Transformation des valeurs linguistiques en une valeur numérique

Taille		Poids		Min(Taille, Poids)
Valeur L.	Degré d'app.	Valeur L.	Degré d'app.	
Normal	0,5	Normal	0,3	0,3
Normal	0,5	Obèse	0,7	0,5
Grand	0,5	Normal	0,3	0,3
Grand	0,5	Obèse	0,7	0,5
Somme 1				1,6

Min(Taille, Poids)	Taille	Poids	Rendez-vous		Produit
			Valeur L.	Centre de gr.	
0,3	Normal	Normal	Aucun	1	0,3 * 1 = 0,3
0,5	Normal	Obèse	Beaucoup	5	0,5 * 5 = 2,5
0,3	Grand	Normal	Aucun	1	0,3 * 1 = 0,3
0,5	Grand	Obèse	Peu	2	0,5 * 2 = 1
Somme 2					4,1

Somme 1	Somme 2	Nombre de rendez-vous à programmer
1,6	4,1	4,1 / 1,6 = 2,56

La logique floue constitue pour notre proposition la base de raisonnement en incluant les valeurs linguistiques (cf. section 4.4.2).

3.2.4 Fouille de données et apprentissage automatique

La fouille de données ou datamining est une technique utilisée pour trouver des patterns sur de grands volumes de données. Elle permet d'abstraire des informations inconnues qui sont souvent difficiles à modéliser ou invisibles par les humains. L'apprentissage automatique (ou machine-learning) utilise des grands volumes de données afin d'apprendre et faire des prédictions. La principale différence entre les deux techniques réside dans leur finalité puisque les méthodes sont souvent partagées. Les techniques utilisées pour la fouille de données et l'apprentissage automatique se basent sur des *modèles statistiques*, sur des *distributions de probabilités*, et sur des *fonctions mathématiques* avec des propriétés particulières. Ces techniques deviennent très intéressantes lorsque le *nombre de variables augmente* et qu'*établir des corrélations devient compliqué* par des processus simples. La fouille de données permet d'abstraire *les causes* d'une alerte, tandis que l'apprentissage permet de *prédire* si une entrée générera, ou non, une alerte.

3.2.4.1 Définition de la situation d'alerte

Considérons deux options qui permettent de savoir quand une alerte a eu lieu *sans définir* ce qui l'a déclenché. Dans la première option, le label « alerte » est exprimé explicitement à un instant précis où un traitement simple sur des données comme « si patient fatigué », ou « intervention d'infirmière », ou « patient intubé » permet de créer le label. La deuxième option consiste à considérer toute entrée avec un comportement inhabituel comme une « alerte ».

Avant d'appliquer les techniques d'apprentissage automatique, les données peuvent être nettoyées afin de garder uniquement celles qui sont intéressantes, les valeurs manquantes remplies avec une valeur moyenne ou les doublons supprimés. Une fois les données traitées, différentes techniques peuvent être utilisées pour classifier une entrée comme étant une « alerte ». Si le label « alerte » est connu, des techniques d'apprentissage supervisé comme la *régression logistique*, *Machine à vecteurs de support (SVM* de l'anglais *Support Vector Machine)*, ou des *réseaux neuronaux* sont utilisés pour l'apprentissage. Si les alertes sont définies comme les situations inhabituelles, et que le nombre de situations est très inférieur au nombre de situations normales, les techniques comme la *détection d'anomalies* sont utilisées pour l'apprentissage.

3.2.4.2 Fonction de classification

En partant du principe que les situations d'alerte qui intéressent l'utilisateur ont été identifiées préalablement, les algorithmes utilisent souvent des techniques de classification (binaires). Ainsi, classifier une entrée (x) comme générant une alerte (y = 1) peut se représenter comme dans la formule (2).

$$\text{Prédire}\begin{cases} y = 1 \; si \; h_{\theta(X)} \geq \varepsilon \\ y = 0 \; si \; h_{\theta(X)} < \varepsilon \end{cases} \quad \textbf{(2)}$$

Où $h_{\theta(X)}$ est une fonction qui paramètre X et la valeur ε est choisie selon la fonction $h_{\theta(X)}$ utilisée. L'hypothèse peut donc s'interpréter comme la probabilité que $y = 1$ étant donné X et *paramétrée* par θ. La fonction de paramétrage $h_{\theta(X)}$ varie selon la technique utilisée et la distribution des données.

Exemple

Si $X = \{x_1, x_2\}$ est composé de deux variables pour la température (x_1) et la tension systolique (x_2), la fonction $h_{\theta(X)}$ peut être définie par des polynômes à plusieurs variables comme :

$$h_{\theta(X)} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$
$$h_{\theta(X)} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 {x_1}^2 + \theta_5 {x_2}^2$$
$$h_{\theta(X)} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \cdots + \theta_{i-1} {x_1}^4 {x_2}^2 + \theta_i {x_1}^5 x_2 + \theta_{i+1} {x_1}^6 + \theta_{i+2} {x_2}^6 + \cdots$$
$$h_{\theta(X)} = f(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 {x_1}^2 + \theta_5 {x_2}^2) \; o\grave{u} \; f(x) = \frac{1}{1 + e^x} \; (fonction\ logistique)$$

Des analyses statistiques des données guident le choix entre les fonctions $h_{\theta(X)}$ possibles à utiliser lors de la conception de l'algorithme. Les *paramètres* θ sont calculés pendant le processus d'apprentissage et sont dépendants de l'ensemble des données utilisées. Le modèle est validé par la *précision*, la *mémoire*, ou l'*exactitude* qui expriment le « bon » comportement du modèle par rapport à une situation représentée des données similaires à celles utilisées dans le processus d'apprentissage.

Des médecins, des utilisateurs experts dans le domaine d'application, peuvent aider les concepteurs, experts dans ces techniques, pendant la conception des modèles. Cependant, il est fréquent que les meilleures prédictions soient faites par des modèles qui utilisent une plus grande quantité de données pendant le processus d'apprentissage (Banko & Brill, 2001). Toutefois, dans le domaine de la santé, ces techniques peuvent ne pas être acceptées par les médecins puisqu'il n'est pas possible d'expliquer la raison exacte pour laquelle le modèle utilise des données X avec les paramètres appris θ pour conclure une alerte $y = 1$.

Dans les systèmes que nous visons, nous sommes confrontés d'une part à une masse de données dont l'utilité est souvent limitée, et d'autre part à des absences de données liées au contexte (indisponibilité à temps, inaccessibilité système, etc.).

3.3 Interprétation du temps

L'aspect temporel a été abordé bien avant l'apparition de l'informatique. Des philosophes comme Aristote ou Diodore Cronos s'étaient déjà intéressés à la logique temporelle (Hasle & Øhrstrøm, 1995). La logique temporelle est un système de règles qui permet de faire du raisonnement et des représentations liées au temps. Cette

section est formée par des modèles qui représentent le temps, ainsi que par des concepts et des opérateurs temporels.

3.3.1 Logiques temporelles

Prior a été le premier à s'intéresser à la logique temporelle (Prior, 1957). Cette logique contient en plus des opérateurs de la logique propositionnelle (∧, v, ¬, →) quatre opérateurs temporels. "P" « il y a eu une fois le cas … ». "F" « il y aura une fois le cas … ». "H" « il y a toujours eu le cas … ». "G" « il y aura toujours le cas … ». "P" et "F" sont considérés comme des opérateurs faibles tandis que "H" et "G" des opérateurs forts. La logique temporelle s'applique à l'intelligence artificielle, au calcul, au langage naturel et à d'autres disciplines qui ont besoin de prouver la faisabilité et la cohérence.

3.3.1.1 Calcul situationnel

McCarthy et Hayes s'attaquent au problème des programmes intelligents capables de choisir quoi faire selon les circonstances (Mccarthy & Hayes, 1969). Ils s'intéressent à la formalisation des concepts de causalité, capacité et connaissances. Leur travail tourne autour du calcul de la *situation*. L'univers est représenté comme un ensemble d'états où les actions sont des fonctions qui permettent de passer d'un état à un autre dans une *situation spécifique*. Une situation est l'état entier de l'univers à un instant temporel. Néanmoins, comme il n'est pas possible de décrire tout l'univers, on ne peut que donner quelques faits à propos de la situation. Finalement, une *stratégie* est une séquence d'actions avec un début et une fin. Le temps est exprimé implicitement sous forme de causalité, soit le passage entre états. McCarthy et Hayes définissent le « problème du cadre ». Il est difficile d'exprimer le dynamisme d'une situation avec un formalisme logique de premier ordre sans avoir défini tout ce qui n'est pas affecté par les actions.

3.3.1.2 Logique des événements

Kowalski et Sergot proposent un calcul basé sur la logique des événements (Kowalski & Sergot, 1986). La logique permet de représenter les connaissances à l'aide du temps afin de s'en servir pour faire du raisonnement (conclure). Ils partent du principe que le système a suffisamment d'informations pour arriver à une conclusion. Ainsi, toute nouvelle information est utilisée pour mettre le système de connaissances à jour à partir de ce qui a été conclu. La logique proposée est non monotone : apprendre une nouvelle connaissance mène à la révision des connaissances existantes. Elle permet de conclure à partir du raisonnement par défaut et l'hypothèse du monde clos. Ainsi, ce qu'il n'est pas possible de démontrer comme étant vrai est considéré comme étant faux. Le calcul à base d'événements a été développé afin d'éviter le « problème du cadre » de McCarthy. La notion temporelle de base est l'événement. Un *événement* décrit une relation (propriété) à un instant donné (point temporel). Les connaissances sont exprimées avec des événements. Les événements ont une *durée* représentée par un *intervalle*. Les propriétés sont vraies pendant un intervalle de temps commencé et terminé par des événements.

Un événement ne représente pas quelque chose qui vient d'être connu. C'est la relation exprimée dans l'information connue qui exprime un événement.

Exemple

« La personne âgée s'est couchée le 10 janvier 2012 à 21h10 » générera un événement ev_1. Si le système apprend que « La personne âgée s'est levée le 11 janvier 2012 à 7h23 » il générera un événement ev_2. Le système peut ainsi conclure que la période commencée par ev_1 est la même que celle terminée par ev_2 étant donné qu'elles sont identiques.

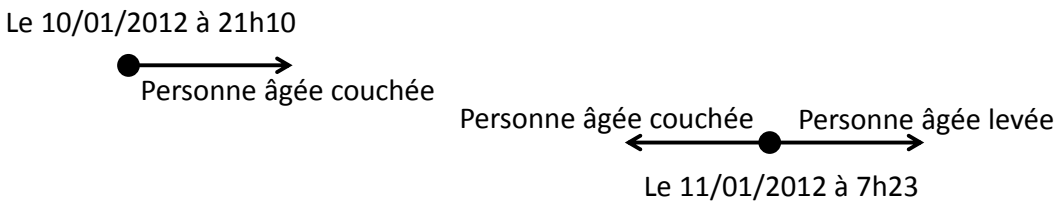


Figure 6. Evénements et propriétés concernant la réalité modelée de Thérèse dans une période spécifique

Les deux périodes sont identiques puisque la même personne a la même position, une des périodes commence avant l'autre et il n'est pas possible de démontrer qu'un autre événement a eu lieu entre ev_1et ev_2 qui affecte la même personne et sa position. La Figure 6 montre la représentation des événements et des propriétés affectant la position de la personne âgée.

3.3.1.3 Calcul d'événements

Le calcul d'événements de Shanahan est une extension du travail de Kowalski et Sergot qui permet d'inférer ce qui est vrai à partir des informations qui expriment *quoi* et *quand* quelque chose se passe (actions) ainsi que ce qui a lieu après ces actions (Shanahan, 1999). De façon similaire aux deux travaux précédents, ce formalisme logique permet de faire du raisonnement temporel. Les concepts de base sont les *axiomes* et des *prédicats* utilisés pour tirer des conclusions. Shanahan utilise le concept action au lieu d'événement, mais il ne fait aucune différence entre les deux. La notion de *fluent* est utilisée pour exprimer n'importe quoi qui peut changer dans le temps. Finalement, le travail utilise des *points temporels* pour représenter les instants. Le formalisme avec les prédicats et axiomes du Tableau 9 permet de représenter des actions avec des effets indirects, des actions avec des *effets non déterministes*, des actions *composées*, *concurrentes* et *changeant en continu*.

Tableau 9. Formules du calcul d'événements de Shanahan.

Formula	Signification
Initiates(α,β,τ)	fluent β vaut vrai après l'action α à l'instant τ
Terminates(α,β,τ)	fluent β ne vaut plus vrai après l'action α à l'instant τ
InitiallyP(β)	fluent β vaut vrai depuis le temps 0
$\tau 1 < \tau 2$	L'instant $\tau 1$ est antérieur à l'instant $\tau 2$

Formula	Signification
Happens(α,τ)	L'action α se passe à l'instant τ
HoldsAt(β,τ)	fluent β vaut vrai à l'instant τ
Clipped(τ1,β,τ2)	fluent β s'est arrêtée entre les instants τ1 et τ2

Les situations sont modélisées en deux parties. Les axiomes qui décrivent le comportement du système et la narrative des événements. A partir de ces informations, le système est capable de conclure l'état actuel (l'arme est déchargée).

Exemple

Tirer avec une arme va la décharger, est une action-effet
L'arme a été chargée avant qu'on ait tiré, est la narrative des événements

3.3.1.4 Indétermination dans le futur

La logique temporelle de McDermott permet d'évaluer l'indétermination dans le futur (Mcdermott, 1982). Des *faits* se passent à des instants temporels (*points*) tandis qu'un *événement* constitue l'intersection des *intervalles* dans lesquels l'événement a eu lieu. Les *intervalles* sont formés par des *états* qui représentent des *images de l'univers* à un instant donné. Les états sont ordonnés partiellement par date dans le passé et se ramifient dans le futur. Chaque branche de la ramification est connue comme un *chronicle*, une histoire possible de l'univers.

3.3.1.5 Système spécialiste du temps

Le système spécialiste du temps (Kahn & Gorry, 1977) est utilisé pour la maintenance des données temporelles. Le système est capable de répondre aux questions en relation avec des *événements* et leur *durée*. Le système utilise les *nouvelles* informations pour garder une cohérence temporelle des informations connues. Trois ordonnancements sont proposés pour faciliter l'inférence, par *temporalité*, par la *position dans une séquence d'événements* (A avant B), et par leur *relation avec des événements importants* comme « maintenant » (A 3 semaines avant maintenant). Des nouvelles informations floues (évènement A il y a « quelques semaines ») sont acceptées par le système mais ne sont pas prises en compte lorsqu'il faut répondre aux questions. Trois types basiques de questions sont acceptés par le système : l'évènement X a-t-il eu lieu à la date T ?, quand a eu lieu l'événement X ?, et qu'est-ce qui s'est passé à la date T ? Les réponses du système ont une sémantique proche du langage naturel et sont produites à l'aide des méthodes qui infèrent les relations temporelles entre les informations connues par le système.

La plupart des formalismes logiques étudiés permettent de faire du raisonnement et prennent en compte l'aspect temporel. La relation de causalité est utilisée implicitement par la plupart de ces formalismes. Quelques-uns d'entre eux expriment des relations temporelles comme « avant » ou « après » que nous étudions en détail ci-dessous.

3.3.2 Relations temporelles

3.3.2.1 Concepts temporels

La modélisation d'une réalité dont l'aspect temporel est important nécessite la représentation du temps. Elle prend deux formes *primitives temporelles* bien connues, des *points* et des *intervalles*. Lorsque les intervalles n'ont qu'une seule limite (début ou fin), ils sont connus sous le nom de *semi-intervalles*. Les *concepts temporels* permettent de représenter la réalité. Les relations temporelles sont utilisées par les algorithmes afin de comparer les primitives temporelles et créer des patrons, lesquels sont utilisés pour l'exploration, la maintenance, le raisonnement ou l'expression des données temporelles.

Exemple

La durée, l'ordre, la synchronie, la périodicité, sont des concepts temporels.
Avant, après, pendant, sont des relations temporelles.

La *durée* est un concept temporel utilisé pour exprimer que la réalité modélisée persiste dans le temps. Elle est souvent exprimée par un entier et associée à un intervalle. Si la durée est exprimée explicitement, elle peut être utilisée pour la construction des intervalles lors de la méconnaissance d'un des points. Elle peut être aussi calculée à partir des points qui forment l'intervalle. L'*ordre* s'applique aux intervalles et aux points. Il exprime l'occurrence dans une séquence. Deux réalités sont *synchrones* si elles sont représentées par des points et elles se déroulent exactement au même temps, ou si elles sont représentées par des intervalles et elles débutent et finissent en même temps. Une réalité est *périodique* si elle se répète avec une périodicité constante. La *concurrence* exprime qu'au moins deux réalités *se déroulent* les unes proches des autres, tandis que la *coïncidence* exprime qu'au moins deux réalités se déroulent pendant une même période de temps.

Les relations temporelles les plus courantes sont *avant*, *après*, et *égal* (en même temps). Ces opérateurs se basent sur le concept d'*ordre*. D'autres opérateurs comme *proche* sont moins courants mais satisfont des concepts comme la *concurrence*.

3.3.2.2 Relations temporelles d'Allen

Allen a proposé une logique temporelle basée sur des intervalles fondée sur sept relations de base et leurs symétriques faisant un total de treize relations (Allen, 1983) La Figure 7 montre les relations de base « avant », « rencontre », « chevauche », « égal », « pendant », « démarre », « termine » et leurs symétriques « après », « rencontré par », « chevauché par », « contient », « démarré par », « terminé par ».

Deux intervalles sont toujours affectés par une seule relation d'Allen. Ainsi, apprendre des nouvelles informations temporelles ajoute au maximum treize nouvelles relations entre l'intervalle appris et chacun des intervalles existants. Ceci est significativement important puisque c'est coûteux en termes de mémoire ($O(N^2)$). Pour garantir la consistance temporelle entre les intervalles et réduire le calcul des relations entre deux intervalles, Allen propose l'utilisation des *intervalles de référence*.

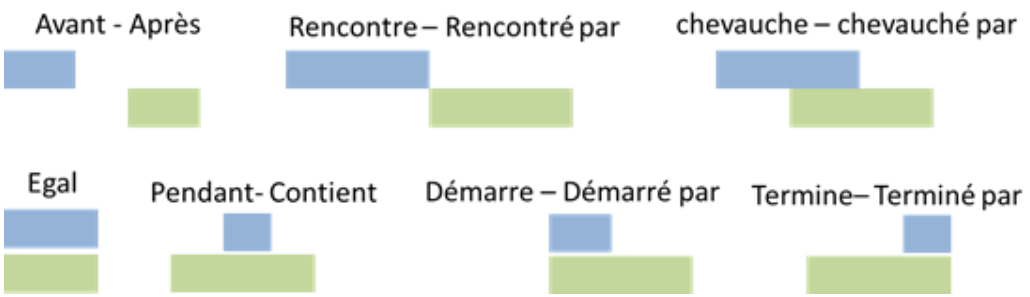


Figure 7. Relations temporelles d'Allen entre intervalles

Les relations d'Allen présentent quelques désavantages lorsqu'il s'agit d'abstraire des connaissances sur des masses de données (Mörchen, 2006). Entre autres désavantages, les relations proposées par Allen ne sont *pas robustes*. La situation réelle entre deux intervalles peut être « similaire » mais être exprimée par des relations d'Allen différentes sans capturer la similarité existante. Les relations peuvent être aussi *ambigües*. La même relation d'Allen exprime des relations réelles « très différentes » qui devraient peut-être être représentées par des relations d'Allen différentes. La Figure 8 et la Figure 9 illustrent ces deux inconvénients.

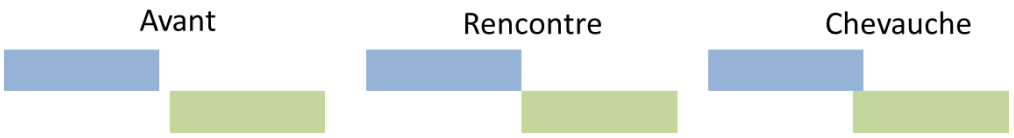


Figure 8. La relation exprimée pour chaque paire d'intervalles est différente. Néanmoins il est possible de les considérer comme étant la même.

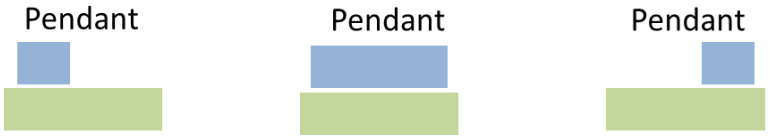


Figure 9. La relation exprimée pour chaque paire d'intervalles est la même. Néanmoins il est possible de la considérer comme étant trois relations différentes.

De nombreux auteurs se sont basés sur les travaux d'Allen. (Roddick & Mooney, 2005) proposent une extension des relations d'Allen à partir du point milieu de l'intervalle. Ainsi la relation « chevauche » d'Allen est représentée par neuf relations. Un total de quarante-neuf relations remplace les treize relations proposées par Allen. Le travail de (Schockaert, et al., 2008) remplace les relations d'Allen par des relations floues et traite l'incertitude existante sur les données réelles.

3.3.2.3 Relations entre semi-intervalles

Les *semi-intervalles* proposés dans le travail de (Freksa, 1992) permettent de traiter l'incertitude. Un semi-intervalle est défini par un *point* qui représente le point

de début ou le point de fin d'un *événement*. Les relations définies entre semi-intervalles sont « plus ancien », « tête à tête », « survie », « queue à queue », « précède », « né avant la mort », et ses symétriques « plus jeune », « survécu par », « succède », « mort après la naissance » (cf. Figure 10). En outre, Freksa définit la relation « contemporain » qui exprime que l'intersection entre deux semi-intervalles n'est pas vide. Parmi les avantages d'utiliser des semi-intervalles se trouve la réduction à deux opérations pour identifier la relation entre les semi-intervalles. Par contre, la relation entre un semi-intervalle et un intervalle peut être exprimée par plus d'une relation.

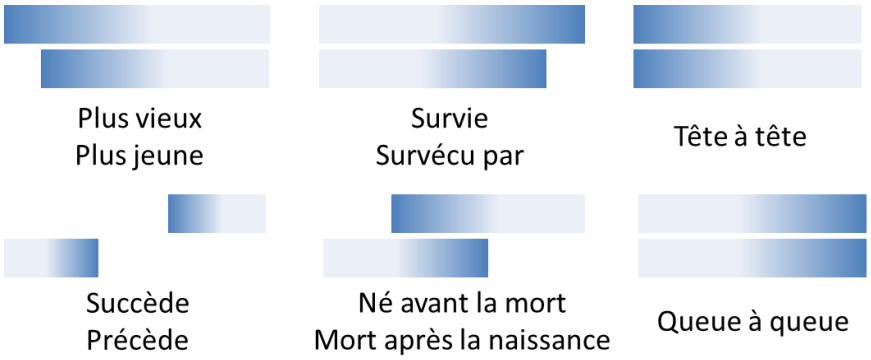


Figure 10. Relations entre semi-intervalles de Freksa.

3.3.2.4 Relations n-aires

Le travail de (Villafane, et al., 1999) se base sur la relation « contient » pour faciliter la compréhension par les utilisateurs. Ainsi A « contient » B remplace les relations d'Allen A « égal à » B, B « démarre » A, B « pendant » A, et B « termine » A. En plus, « contient » est une relation transitive ce qui peut réduire la complexité des algorithmes. La relation de « coïncidence » est proposée par (Mörchen, 2006). Plusieurs intervalles « coïncident » si l'intersection entre tous les intervalles est différente du vide. La relation est utilisée pour faciliter l'interprétation des résultats et augmenter la vitesse des algorithmes d'exploration de données puisque les mêmes événements peuvent s'interpréter sous plusieurs formes différentes (Figure 11).

Exemple

i -) A « contient » B, B « rencontre » C, C « est chevauché par » D.
ii -) A « chevauche » C, C « est rencontré par » B, B « démarre » D.
iii -) A est « terminé par » D, A « chevauche » C, A « contient » B, D « chevauche » C, D « est démarré par » B, C « est rencontré par » B.
iv -) A, D et B « coïncident » « avant » C.

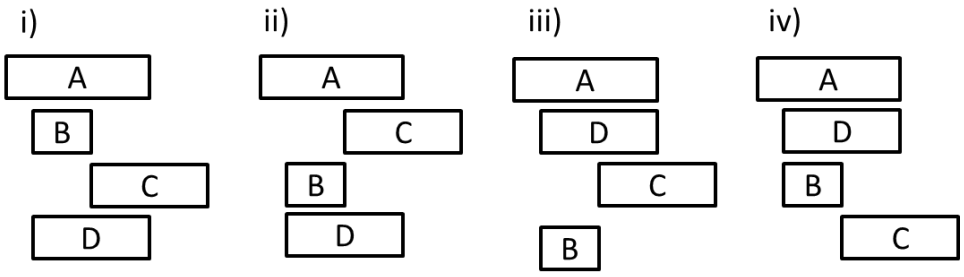


Figure 11. Représentation des informations temporelles.

Dans le travail de (Kam & Fu, 2000) il y a deux types de patrons temporels, les patrons *atomiques*, et les patrons *composés*. Les patrons *atomiques* sont des occurrences des événements. Les patrons *composés* sont exprimées à l'aide de relations temporelles d'Allen (3) afin de réduire le nombre de possibilités (exponentiel) des relations entre événements. En outre, les événements sont ordonnés par la date de fin afin de réduire le nombre de représentations des événements. La Figure 12 montre les événements i -) sans ordre, et ii -), iii -) ordonnés par la date de fin.

Exemple

Le patron (((B pendant A) terminé par D) chevauche C) est représenté dans ii -)
Le patron (((B démarre D) termine A) chevauche C) est représenté dans iii -)

$$\left(\ldots\left(\left((A\ r_b\ B)\ r_c\ C\right)\ldots\right) r_z\ Z\right)\ \text{où}\ r_b, r_c, r_z\ \textit{sont des relations d'Allen} \quad \textbf{(3)}$$

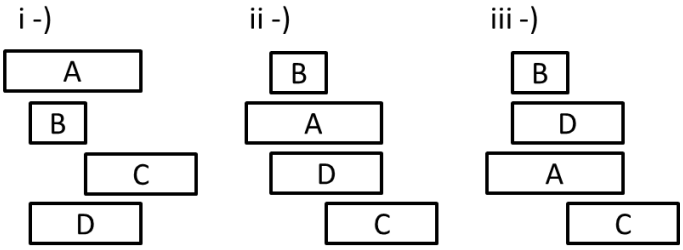


Figure 12. Evénements ordonnés par date de fin.

3.3.2.5 Relation entre objets et lignes du temps

Une ontologie temporelle est proposée par (Keravnou, 1996) pour décrire les processus afin de faire du diagnostic médical. Les primitives les plus importantes sont l'*axe temporel* et les *objets temporels*. Chaque axe temporel a des unités temporelles qui expriment sa granularité.

Exemple

Le temps de vie, la période fœtale, l'enfance, l'année 1990, la période du monitoring fœtal, sont des axes temporels.
Année, mois lunaire, mois calendrier, journée, sont des granularités.

Les objets *temporels* peuvent exister en plus d'un axe temporel. La relation d'un objet avec un axe temporel définit la « date de début » et « date de fin » de l'objet dans l'*axe*. Une deuxième relation avec l'*axe* permet de déterminer si l'*objet temporel* est « fermé » lorsque les dates sont connues, « ouvert » si la durée est connue mais pas les dates, « semi-ouverte » si au moins une date est connue, ou « en mouvement » si l'objet temporel se produit « maintenant ». Les relations binaires entre axes temporels peuvent être « concurrent », « inclus », « immédiatement antérieur », « suffisamment antérieur ». Les relations entre les objets temporels correspondent à celles d'Allen. Un troisième type de relations définit le « lien entre deux instants dans deux axes temporels distincts » pour déterminer l'instant d'intersection entre les axes, et le « lien entre deux objets temporels distincts dans deux axes temporels distincts » représentant une même correspondance dans le monde réel. Ainsi deux objets temporels peuvent avoir deux relations différentes dans deux axes distincts.

Exemple

Un objet A peut chevaucher un objet B dans un axe temporel X et être concurrent (égal) dans un axe Y (avec une plus grosse granularité).

Les systèmes visés sont souvent concernés par des enchaînements temporels caractéristiques que nous pouvons analyser sous forme des séries chronologiques.

3.4 Les séries chronologiques

Une série chronologique est un ensemble de points bidimensionnels séquentiels qui représentent dans beaucoup de cas des observations.

Les séries chronologiques sont multidisciplinaires. Leur application la plus courante est la prédiction utilisée par exemple dans les domaines boursiers ou météorologiques. Dans le domaine de la santé, les séries chronologiques sont utilisées pour la visualisation des variables médicales d'un patient afin de mieux comprendre leur évolution. Les abstractions sur les séries chronologiques, telles que les tendances peuvent augmenter l'expressivité des situations d'alerte.

Exemple

Notifier une alerte si la température est « à la hausse » et la tension reste « stable »

3.4.1 Approximation des séries chronologiques

Un hôpital qui reçoit à peu près 2000 patients par an surveillés en moyenne pendant 3 jours, dont 4 constantes médicales sont relevées toutes les minutes, cela produit environ trente-quatre millions d'observations (34 560 000).

Etant donné la quantité de données utilisées dans une série chronologique, de nombreux travaux visent à réduire des dimensions en utilisant des techniques d'approximation. Cela peut réduire la taille des bases de données, ou accélérer la recherche de résultats similaires.

Exemple

Trouver des compagnies avec une croissance similaire.
Trouver des produits avec de patrons de vente similaires.
Trouver des patients avec une évolution similaire des constantes médicales.

Dans le travail de (Agrawal, et al., 1993), les séries chronologiques sont considérées comme des *signaux*. La transformation de Fourier permet d'abstraire les principales caractéristiques utilisées pour représenter la série chronologique. La transformation par *l'ondelette de Haar* est proposée par (Chan & Fu, 1999). Les premiers coefficients de la transformée sont utilisés pour l'*indexation* dans un R-Tree utilisé pour la recherche de similitudes. (Keogh, et al., 2001) proposent *PPA* (de l'anglais Piecewise Aggregate Approximation), une technique dont la série chronologique est approximée par morceaux. Les auteurs comparent PPA avec les approximations précédentes et concluent que PPA est plus effective et efficiente dans la perspective de la fouille de données.

Le travail de (Chan & Mahoney, 2005) fait la corrélation entre plusieurs séries chronologiques afin de détecter des anomalies, des comportements inattendus. Un algorithme ascendant trouve un nombre défini de *boîtes* pour chacune des séries chronologiques. Dans un deuxième temps, l'algorithme *fusionne* les boîtes des séries chronologiques différentes, les plus proches, jusqu'à trouver les boîtes les plus représentatives. Une *anomalie* est un point dont la « distance » à la boîte la plus proche dépasse un seuil prédéfini.

3.4.2 Langage d'interrogation

The Shape Definition Language (SDL) est un langage pour retrouver des objets à partir de la *forme* de leur historique (Agrawal, et al., 1995). L'historique est représenté par une série chronologique. La *forme* représente un pattern souvent constitué par la concaténation des *symboles* appartenant à un *alphabet*. Chaque *symbole* de l'alphabet représente une *transition* possible comme « transition ascendante douce » ou « transition ascendante forte ». Une requête trouve toutes les subséquences dans une série chronologique semblables à une *forme*. Le langage fournit trois opérateurs. L'opérateur « concat » spécifie la concaténation des *symboles*. « Any » est l'opérateur joker qui matche n'importe quel symbole. Enfin, l'opérateur « at least » permet de faire un filtrage semblable à celui d'expressions régulières.

Exemple

« Ascendante » « stable » « ascendante », est une forme.

3.4.3 Abstraction des tendances

L'*état abstrait* de (Larizza, et al., 1997) est une description qui exprime le comportement journalier d'un patient. Ce comportement comprend un ensemble de variables mutuellement exclusives qui doivent être surveillées chaque jour. Les états abstraits sont construits avec des méthodes d'abstraction temporelle qui extraient l'information plus importante des données longitudinales. Deux types de méthodes d'abstraction sont utilisés. Les méthodes du premier type obtiennent des *intervalles* à

partir d'une série chronologique selon les tendances les plus importantes « augmente », « stationnaire », ou « diminue ». Chaque intervalle représente un *épisode*. Les méthodes d'abstraction du deuxième type utilisent les *épisodes* afin de faire des abstractions temporelles complexes, en utilisant les relations temporelles (d'Allen) entre deux *épisodes*. Les *tendances* sont des abstractions temporelles calculées à partir de trois propriétés. Le « taux » définit la vitesse minimale du changement. La « granularité » spécifie la distance maximale (en jours) entre deux observations afin de former un intervalle (épisode), et la « durée minimale » pour qu'un intervalle puisse être considéré comme étant signifiant (Larizza, et al., 1995).

Exemple

3 unités par jour est un « taux ». 2 jours est une « granularité ». 5 jours est une « durée minimale »

Dans une séquence d'intervalles, il y a plusieurs paires candidates à la fusion. De plus, sur une série chronologique, il est possible de faire des nombreuses abstractions d'intervalles qui partagent des périodes temporelles. (Shahar & Musen, 1993) et (Larizza, et al., 1995) utilisent un tableau d'abstraction d'inférences avec toutes les combinaisons possibles afin de savoir quelles paires peuvent être fusionnées dans un contexte spécifique. Ces sont les connaissances du contexte qui permettent de choisir quelles sont les bonnes abstractions.

Exemple

« Croissant » suivi de « croissant », « décroissant suivi de stable », sont des combinaisons.

3.4.4 Détection de tendances indépendamment du domaine

Un processus destiné à retrouver toutes les tendances dans une série chronologique indépendante du domaine d'application est proposé par (Udechukwu, et al., 2004). Un *alphabet* représente tous les changements possibles dans une série chronologique. Les *symboles* sont des lettres de l'alphabet représentant des *incréments angulaires*. Les lettres minuscules sont des changements positifs tandis que les majuscules des changements négatifs. Le travail est dit indépendant du domaine d'application grâce à l'utilisation des *angles* pour classifier les tendances (Lowe, et al., 1999).

Exemple

Les degrés entre « 0 » et « 39 » sont codés avec les lettres entre « a » et « h » avec des incréments de 5 degrés.

Les degrés entre « 40 » et « 49 » sont codés avec les lettres entre « i » et « r » avec des incréments de 1 degré.

Les degrés entre « 50 » et « 90 » sont codés avec les lettres entre « s » et « z » avec des incréments de 5 degrés.

Un système pour l'abstraction temporelle fondée sur la connaissance est proposé par (Shahar, 1997). Ce système est indépendant du domaine et inclut des modèles pour le *temps*, les *paramètres*, les *événements*, et les *contextes*. L'abstraction

temporelle est une interprétation du temps pour obtenir des résultats tels que « fièvre pendant les trois dernières jours ». La tâche *d'abstraction de connaissances* est composée de cinq sous-tâches résolues par un *mécanisme d'abstraction* indépendant du domaine d'application, la *restriction temporelle du contexte*, l'*inférence temporelle verticale*, l'*inférence temporelle horizontale*, l'*interpolation temporelle*, et l'*appariement de formes temporelles*. Chaque *mécanisme* est lié naturellement au moins à deux types de connaissances parmi les connaissances « structurelles », de « classification », de « sémantique temporelle », ou de « dynamique temporelle ». La « sémantique temporelle » exprime quand il est possible de concaténer deux intervalles. La « classification » attribue une *valeur* à un intervalle ou une *valeur* pour le résultat de la jointure de deux, voire trois, *valeurs*. L'*interpolation temporelle* détermine si deux paramètres peuvent être abstraits en un seul. L'*inférence temporelle horizontale* permet de savoir quelle est la valeur de l'abstraction résultante. La classification utilise des fonctions d'abstraction dont les paramètres sont l'« état », le « gradient », ou le « taux ».

Exemple

Deux intervalles consécutifs avec des valeurs « croissante » et « stable » peuvent être abstraits comme étant un seul intervalle dont la valeur est « croissante ».

3.4.5 Segmentation des séries chronologiques

La détection des tendances est souvent liée à la segmentation des séries chronologiques dont chaque segment résultant du processus est utilisé pour détecter une tendance. Il y a trois types d'algorithmes utilisés pour la segmentation des séries chronologiques. Les algorithmes « ascendants », « descendants », et de « fenêtres temporelles ».

3.4.5.1 Segmentation offline

Les algorithmes « ascendants » et « descendants » sont largement utilisés dans le scénario « offline » où ce que le système doit trouver sur la série chronologique est connu à l'avance, comme le nombre de tendances à détecter. En outre, le système a accès à des informations du passé et du futur par rapport à un instant t.

Les algorithmes « ascendants » divisent la série chronologique en petits segments, et les fusionnent jusqu'à trouver les k segments les plus représentatifs. Les algorithmes « descendants » fractionnent la série chronologique en un seul segment qui est segmenté en continu jusqu'à trouver les k segments les plus représentatifs.

L'algorithme « ascendant » de (Salatian & Hunter, 1999) détecte des tendances dans des scénarii « offline ». Le système essaie de *fusionner* des segments consécutifs dont la « valeur de la tendance » et le « taux de changement » est le même. Ensuite, le système fusionne les segments depuis le plus ancien jusqu'au plus récent à l'aide d'une *table d'inférence* pour deux segments qui se rencontrent. La fusion s'arrête lorsque le segment résultat de la fusion s'infère comme étant « stable ». Ensuite, une nouvelle fusion commence à partir du segment d'après. Une fois qu'il n'est plus possible de

fusionner deux segments consécutifs, le système utilise une deuxième *table d'inférence* pour fusionner trois segments consécutifs.

Exemple

Des segments « croissant », « décroissant », et « croissant » ne peuvent pas être fusionnés.

Des segments « croissant », « stable », « croissant » sont fusionnés dans un segment « croissant ».

3.4.5.2 Segmentation online

Dans les scénarii « online », le système a accès seulement aux informations du passé par rapport à l'instant présent, « maintenant ».

Exemple

Il faut réduire le temps de calcul en temps réel pour accélérer le temps de réponse tout en obtenant des résultats satisfaisants, c'est un scénario online.

Il faut déterminer le comportement en temps réel et les informations connues ne permettent pas de prédire les valeurs futures, c'est un scénario online.

Les algorithmes de « fenêtre glissante » sont utilisés dans des scénarii « *online* ». Une fenêtre contient un ensemble d'observations qui forment un segment. Si la « nouvelle observation » arrivant peut être fusionnée avec le segment de la fenêtre, la fenêtre s'agrandit et intègre ainsi la « nouvelle observation » qui fait ainsi partie du segment. Dans le cas contraire la fenêtre glisse en laissant derrière un segment défini en commençant le calcul d'un nouveau. Le fait de « glisser » permet d'achever une réduction de calcul en utilisant un nombre limité de données pour réduire le temps de lecture et maximiser la place dans la mémoire.

« Glisser » une fenêtre consiste à glisser sur une certaine distance la limite inférieure et la limite supérieure. La distance à glisser peut être constante ou être attachée à d'autres facteurs comme le nombre de données à l'intérieur de la fenêtre, ou au comportement du segment dans la fenêtre (Petit, et al., 2010).

L'algorithme pour la segmentation online de (Salatian & Hunter, 1999) essaie de fusionner la « nouvelle observation » avec le dernier segment connu à partir d'une table d'inférence. Si la fusion n'est pas possible, la limite supérieure de la fenêtre glisse sur une distance prédéfinie préalablement, qui lorsqu'elle est atteinte, déclenche une nouvelle analyse pour trouver les segments plus représentatifs dans la fenêtre.

3.4.5.3 Approche mixte

Une comparaison entre les trois types d'algorithmes est faite par (Keogh, et al., 1993). Les auteurs utilisent les trois types d'algorithmes sur différentes séries chronologiques pour obtenir les segments qui représentent le mieux les séries chronologiques. L'algorithme « ascendant » retourne les meilleures approximations tandis que l'algorithme de « fenêtre glissante » retourne les pires. Une approche mixte entre « ascendante » et « glissante » pour segmenter les séries chronologiques permet

d'obtenir de bonnes approximations tout en gardant la particularité d'être online (Keogh , et al., 2001).

3.4.5.4 Utilisation de la logique floue

(Sittig , et al., 1992) utilisent la logique floue et proposent des fonctions d'appartenance pour classifier la « fréquence cardiaque » par rapport à la « sévérité », « durée », « stabilité ». D'autres fonctions d'appartenance permettent de classifier le résultat de la classification de la « fréquence cardiaque » selon la « confiance de l'estimation ». Les fonctions d'appartenance sont sigmoïdales et représentent les différents ensembles flous pour l'« ampleur » (battements par minute), le « changement », l'« intervalle temporel », la « confiance », et la « stabilité du changement » de la « fréquence cardiaque ».

Exemple

Des ensembles flous représentent les valeurs « menace faible », « marginal faible », « hautement satisfaisante », « marginal élevée », « menace », et « artefact » de l'« ampleur » de la « fréquence cardiaque ».
Des ensembles flous représentent les valeurs « zéro », « stable », « lent », et « rapide » du « changement » de la « fréquence cardiaque ».
Des ensembles flous représentent les valeurs « instantané », « très court », « court », « moyen », et « long » de l'« intervalle temporel » de la « fréquence cardiaque ».
Des ensembles flous représentent les valeurs « artefact », « minimale », « faible », « moyenne », et « élevée » de la « confiance » de la « fréquence cardiaque »
Des ensembles flous représentent les valeurs « instable », « faible », « moyenne », et « élevée » de la « stabilité du changement » de la « fréquence cardiaque ».

L'algorithme essaie de trouver l'intervalle le « plus long » qui s'adapte le mieux à une *situation* spécifique. La recherche est faite à partir de toutes les observations dans un intervalle de soixante minutes (4 observations par minute) et calcule l'« interception », la « pente », et la « déviation absolue moyenne ». Les valeurs sont utilisées pour déterminer si l'approximation est « très imprécise » ou « très proche » de la *situation* cherchée. L'intervalle le « plus long » qui s'adapte le mieux à la *situation* recherchée est celui dont l'approximation n'est pas « très imprécise » ni « très proche ». Si l'approximation est « très imprécise », la taille de l'intervalle est augmentée de 50% et le processus recommence. Si l'approximation est « très proche », la taille de l'intervalle se réduit de moitié et le processus recommence.

Exemple

Trouver l'intervalle le plus long dont l'ampleur de la fréquence cardiaque représente une menace.

3.4.5.5 Applications dans le domaine de la santé

Un algorithme pour la détection d'*évènements* dans des séries chronologiques est proposé par (Hunter & McIntosh, 1999). Les *évènements* sont définis à partir des segments détectés sur les séries chronologiques avec un algorithme ascendant.

Exemple

L'événement « sonde débranchée » est définie comme un segment détecté sur la série chronologique formée par les observations de l'« oxygène » dont la « pente » est « supérieure » à « 0.4 kPa/sec » et qui « chevauche » un segment détecté sur la série chronologique formée par les observations du « dioxyde de carbone » avec une pente de « -0.04kPa/sec ».

Le travail de (Charbonnier & Gentil, 2007) est un système d'alertes définies à partir des *tendances* sur les données des unités de soins intensifs c'est-à-dire des observations des signes vitaux surveillés périodiquement (plusieurs par minute) et formant des séries chronologiques régulières. Trois primitives représentent des *épisodes* qui décrivent l'évolution des signaux, « décroissant », « stable » et « croissant ». La détection de tendances se passe en quatre temps. D'abord la série chronologique est *segmentée*. Ensuite, le dernier segment est classifié dans une *forme* parmi sept possibles, « stable », « croissant », « décroissant », « étape positive », « étape négative », « transition croissante ou décroissante », ou « transition décroissante ou croissante ». Par la suite, La *forme* est transformée dans l'une des trois *primitives* (« décroissant », « stable » ou « croissant ») représentant des *semi-épisodes*. Finalement, le *semi-épisode* est fusionné avec des *épisodes* déjà existants afin d'obtenir des *épisodes* plus longs. Les *épisodes* sont utilisés pour définir des situations d'alerte. Une nouvelle version de l'algorithme de segmentation est présentée dans (Charbonnier & Portet, 2012) dont les *seuils* utilisés pour la *segmentation* et la *fusion* s'auto-adaptent à partir d'une estimation de la variance du bruit du signal. Deux paramètres fixes α et β sont utilisés pour recalculer les valeurs des *seuils*. L'algorithme est moins sensible à la présence d'artefacts.

3.5 Contexte

La définition du contexte proposé par (Dey, 2001), ci-dessous, est largement répandue dans la littérature. Cette définition a été choisie dans le cadre de l'intelligence ambiante qui consiste en l'utilisation intelligente des informations fournies par des sources hétérogènes telles que les réseaux de capteurs. Néanmoins, elle s'applique largement dans le cadre des systèmes d'information.

« Le contexte est toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une personne, un endroit, ou un objet qui est considéré pertinent à l'interaction entre un utilisateur et une application, y compris l'utilisateur et les applications elles-mêmes »

3.5.1 Raisonnement temporel

Dans les travaux de calcul situationnel de (Mccarthy & Hayes, 1969) la logique d'événements de (Kowalski & Sergot, 1986) la notion de contexte est utilisée pour lever l'ambiguïté des opérateurs temporels ou modaux. Toutefois, le contexte n'y est pas formalisé. Les tables d'inférence de (Shahar & Musen, 1993) et (Larizza, et al., 1995) utilisent des tableaux d'abstraction d'inférences définies selon le contexte afin de fusionner des segments consécutifs approximant des séries chronologiques.

Le système pour l'abstraction temporelle fondée sur la connaissance de (Shahar, 1997) définit une *proposition* et une *structure* concernant le contexte. L'interprétation du contexte est une *proposition* qui représente un *état* de choses qui, lorsqu'il est *interprété* sur un *intervalle* de temps peut changer l'interprétation d'un ou plusieurs paramètres. L'*intervalle contextuel* est une *structure* qui représente l'interprétation du contexte sur un intervalle temporel.

3.5.2 Finalité et objectif

Postérieurement, le contexte a été associé à la notion de *finalité* (Bucur, et al., 2005) où chaque *finalité* est liée explicitement à un ensemble d'attributs « pertinents ». Les finalités permettent au système de sélectionner et récupérer les instances des attributs pertinents dans la finalité.

Deux dimensions de contexte sont différenciées dans le travail de (Prekop & Burnett, 2003) : la dimension *extérieure* du contexte utilise les éléments de l'environnement physique et la dimension *intérieure* du contexte définie à partir des objectifs des utilisateurs.

Exemple

L'« emplacement », la « proximité d'autres objets », la « température », le « temps », les « niveaux d'éclairage », sont des éléments de l'environnement.
Les « tâches », le « contexte de travail », l'« état émotionnel », l'« état physique », sont des objectifs de l'utilisateur.

La plupart de systèmes dits sensibles au contexte utilisent seulement la dimension extérieure du contexte.

3.5.3 Sensibilité au contexte

La notion de contexte n'est pas nouvelle. Toutefois, elle a pris de l'importance avec les systèmes dits sensibles au contexte. « *Un système est sensible au contexte s'il utilise le contexte afin de fournir des informations et/ou des services pertinents à l'utilisateur, où la pertinence dépend de la tâche de l'utilisateur* » (Dey, 2001). Pour (Baldauf, et al., 2007) l'adaptation automatique doit être prise en compte. Ainsi, *« un système est sensible au contexte s'il est capable de s'adapter au contexte actuel sans l'intervention d'un utilisateur »*.

Deux approches sont les plus utilisées pour capturer le contexte. Le premier approche par l'intermédiation de middlewares, des systèmes de gestion à grande échelle de flux de données hétérogènes qui permettent de faire la fusion des données issues des capteurs afin de produire des informations qui expriment le mieux le contexte (Gürgen, 2007).

Exemple

Capteur 1 : température élevée au salon, capteur 2 : présence d'une personne dans le salon, donc il y a une personne au salon et la température est élevée.

La seconde approche capture le contexte à l'aide d'algorithmes d'apprentissage automatique qui permettent aux applications de détecter les contextes à partir de l'interprétation des données, souvent issues des capteurs (Laerhoven, 2001), (Heierman & Cook, 2003).

3.5.3.1 Gestion du contexte

Le travail de (Henricksen, 2003) étudie le contexte dans le cadre de la conception d'applications. Il propose trois couches dédiées au contexte : l'« acquisition » des données pertinentes issues des capteurs, la « réception » qui transforme les données hétérogènes en informations exploitables par la gestion du contexte et la « gestion » qui stocke toutes les informations susceptibles d'être sensibles au contexte. A cause des informations contextuelles imparfaites, la couche de « gestion » du contexte permet d'associer une valeur de qualité aux faits. Les *dimensions de qualité* ainsi que les *métriques* pour les calculer varient selon le type d'information.

Exemple

La qualité est exprimée par la « précision » lorsque les informations sont issues des capteurs.

La qualité est exprimée par la « crédibilité » de la personne si c'est elle qui fournit les informations.

3.5.3.2 Prédiction du contexte

Dans la thèse de (Mayrhofer, 2004), un système est sensible au contexte s'il s'adapte à son environnement. A partir d'un ensemble d'observations, le système retourne le *label* qui identifie le contexte actuel et un ensemble de contextes *prédits* dans un futur proche. La prédiction des contextes est divisée en cinq processus. L'« acquisition de données » des capteurs, l'« abstraction des caractéristiques », la « classification » (par rapport à la similarité), l'« attribution des labels », et la « prédiction ». La « classification » et la « prédiction » sont faites à partir des algorithmes d'apprentissage automatique. Le « contexte prédit » est utilisé pour l'adaptation du système.

(Benazzouz, 2011) travaille sur la découverte du contexte afin d'obtenir une *adaptation* automatique des services ubiquitaires. Le contexte est limité au niveau le plus bas (les données) afin d'utiliser des techniques de fouille de données pour la découverte automatique. Chaque donnée contient toutes les informations utiles pour la caractériser : sa signification, sa confiance, l'horodatage, l'entité concernée, et la source émettrice. A partir de ces informations et d'une ontologie, chaque donnée devient une donnée de contexte : un vecteur de 7 valeurs qui dépendent de l'ontologie. Les contextes sont formés avec toutes les données du contexte similaires. Deux données sont similaires si la fonction de similitude retourne une valeur supérieure à un seuil μ fixe. La similitude entre des données se calcule à partir des distances syntaxiques (différence syntaxique entre les termes), sémantique (différence entre les concepts), et temporelle (importante de la distance entre les horodatages). La découverte du contexte est résumée comme le processus d'extraction automatique des situations à partir d'un large volume de données.

3.6 Approche opérationnelle du traitement des données

Dans cette section nous présentons une approche opérationnelle plus large concernant le traitement des données dans les systèmes d'information. Nous abordons les dimensions temporelles dans les bases de données, les fonctions temporelles ainsi que le traitement d'événements.

3.6.1 Dimensions temporelles dans les bases de données

Les bases de données représentent le cœur des systèmes d'information. Elles sont utilisées pour stocker des données et faciliter leur accès. Ils existent des bases de données temporelles qui intègrent la gestion du temps. Les systèmes de gestion de base de données fournissent en plus des fonctions classiques, des fonctions temporelles qui permettent de faire des requêtes avec des contraintes temporelles.

En général, les données ont plusieurs dimensions temporelles. Dans beaucoup de cas, deux dimensions temporelles suffisent, le *temps de validité* et le *temps de transaction* (Jensen, et al., 1997). Toutefois, d'autres dimensions temporelles qui sont souhaitables dans le système d'information ne peuvent pas être décrites à partir du temps de validité et du temps de transaction (Ozsoyoilu & Snodgrass, 1995).

3.6.1.1 Le temps de validité et le temps de transaction

Le temps de validité exprime quand un fait vaut *vrai* dans la réalité représentée. Il peut être représenté à l'aide de points ou d'intervalles temporels. Cette valeur n'a pas de contraintes temporelles, elle peut appartenir au passé, au présent ou au futur. D'habitude, c'est l'utilisateur qui fournit cette valeur. La définition du temps de validité n'est pas très stricte et permet de considérer ce temps comme l'union de plusieurs intervalles et/ou instants. Le *temps de transaction* est fourni par le système et ne peut pas être une valeur du futur. Il exprime quand un fait a été stocké dans la base de données et reste accessible. Dans le domaine médical, le temps de validité permet d'avoir un historique du patient. Le temps de transaction est utilisé pour voir la trace des interactions avec le système.

Exemple

Le patient Dupont a eu de la Fièvre du 10/02/2013 à 8h30 au 10/02/2013 à 10h58. L'infirmier a rentré une valeur de température pour le patient Dupont le 10/02/2013 à 11h10, une valeur pour le pouls et quatre valeurs pour la diurèse le 10/02/2013 à 11h12.

Les temps de validité et de transaction sont souvent définis à l'aide des *horodatages* (timestamps) que (Jensen, et al., 1997) définissent comme un temps associé à un objet. Le *format* du temps est défini dans le type de la donnée temporelle, dans une représentation conforme aux besoins des utilisateurs. En général, la représentation suit un format propre à la culture comme le pays, la langue, ou le calendrier. Le standard international ISO 8601 a été développé afin de faciliter l'échange de données temporelles.

3.6.1.2 Le temps d'événement

(Kim & Chakravarthy, 1994) expliquent qu'il n'y pas de *validité absolue* dans le monde réel, sinon, qu'elle est dépendante *de l'interprétation*. Ainsi, l'interprétation est multiple. Si nous considérons les systèmes d'information comme le flux d'information d'une organisation et ses ressources humaines et techniques, le temps de validité et le temps de transaction ne font pas la différence entre des mises à jour *rétroactives* et des mises à jour *différées*.

Exemple

Dans un premier scénario, le patient passe du bloc opératoire à la salle de réveil le 10/02/2013 à 11h. Ce changement a été stocké dans la base de données le 10/02/2013 à 11h23. C'est une mise à jour différée étant donné que le changement est connu (10/02/2013 à 11h) avant d'être inséré dans la base de données.
Dans un deuxième scénario, le changement de service est stocké le 10/02/2013 à 11h23, mais le changement a été validé depuis le 10/02/2013 à 11h. C'est une mise à jour rétroactive étant donné que le changement a été validé avant d'être connu.

Pour pouvoir différencier entre des mises à jour rétroactives et différées, (Kim & Chakravarthy, 1994) proposent le *temps de l'événement*. Dans leur travail, un événement est la cause de nouvelles propriétés et le *temps de l'événement* est le temps de survenue de l'événement ; l'idéal étant que les *propriétés* affectées par un événement soient connues aussitôt que l'événement a eu lieu. Il se trouve que dans beaucoup de cas, le *temps d'affectation* diffère du *temps de l'événement*. Trois types d'événements sont considérés, les « événements à l'heure », les « événements rétroactifs », et les « événements proactifs ». (Combi & Montanari, 2001) mettent en évidence que des événements rétroactifs peuvent mener à la confusion.

Exemple

Modifier la prescription d'un médicament en changeant le temps de validité rétroactivement est interdit dans le domaine de la santé.

Ainsi un seul temps pour représenter les événements ne suffit pas. (Combi & Montanari, 2001) expliquent que le temps de l'événement utilisé par les systèmes est celui qui commence le temps de validité. Cependant, commencer un nouveau *temps de validité* termine implicitement le *temps de validité actuel* lié à la même propriété. Cela devient plus compliqué lorsqu'un événement termine le temps de validité actuel lié à une propriété différente. Il est nécessaire de distinguer les événements qui commencent le temps de validité et ceux qui le terminent.

Exemple

Il faut arrêter l'administration d'un médicament avant la date prévue à cause de l'évolution de l'état de santé du patient.

3.6.1.3 Le temps de disponibilité

Combi et Montanari proposent deux dimensions temporelles additionnelles, le *temps d'événement qui initie* le *temps de validité* et le *temps d'événement qui termine* ce *temps de validité*. Bien que ces dimensions temporelles semblent délimiter les

événements de la *logique d'événements* de (Kowalski & Sergot, 1986), deux différences s'imposent. D'abord, les temps délimitant les événements de la logique d'événements se calculent au moment d'une requête, tandis que pour (Combi & Montanari, 2001), le temps de validité est connu à l'avance. La deuxième différence, est que (Combi & Montanari, 2001) travaillent non seulement avec les « événements instantanés », mais aussi avec des événements *rétroactifs* et *proactifs*.

Les auteurs proposent aussi le *temps de disponibilité* qui est un intervalle pendant lequel un fait est considéré comme étant correct par le système d'information. Puisque les auteurs considèrent les systèmes d'information comme le flux d'information d'une organisation et ses ressources humaines et techniques, l'intérêt est de différentier le temps quand un fait est disponible dans le système d'information et quand il est ajouté à la base de données (temps de transaction).

3.6.2 Fonctions temporelles dans les bases de données

Les systèmes de gestion des bases de données supportent des langages pour construire des requêtes qui permettent de récupérer des données. Le standard SQL (Langage de Requêtes Structurées - Structured Query Language) est utilisé dans les bases de données relationnelles. SQL est un langage puissant lorsqu'il s'agit d'obtenir des informations sur leur état actuel. En revanche, les requêtes qui expriment des contraintes temporelles sont plus difficiles à structurer, même pour des utilisateurs experts (Jensen & Snodgrass, 2008).

Le but des systèmes de gestion des bases de données temporelles est de faciliter les requêtes dont l'aspect temporel est important. Une solution proposé par (Soo, et al., 1995) permet de construire des requêtes dont la jointure est faite à partir des données temporelles dont l'intersection des horodatages est différente de l'ensemble vide. Les résultats de la requête sont associés à un horodatage qui correspond à l'intersection des horodatages des données jointes. Une autre solution est TSQL2, une extension du SQL avec une syntaxe temporelle (Snodgrass, 1995).

Exemple

La clause Valid(p) retourne le temps de validité de p.

Les systèmes de gestion de bases de données supportent des fonctionnalités pour la manipulation des dates, parmi elles, retourner la date et/ou l'heure actuelle, ajouter des unités temporelles à une date, retourner la différence temporelle entre deux dates en spécifiant l'unité temporelle, ou changer le format d'une date. Toutefois, les fonctions ne sont pas standardisées.

Tableau 10. Requêtes similaires entre SQL Server et Oracle

Différence en « mois » entre le 10 mai 2013 et le 14 juin 2014	
Requête	**Résultat**
SQL server : select DATEDIFF(MM, '2013-05-10 00:00:00','2014-06-14 00:00:00')	13
Oracle client 10g : select MONTHS_BETWEEN('14-06-2014','10-05-2013') from dual	13,13

Différence en « mois » entre le 17 mai 2013 et le 14 juin 2014	
Requête	**Résultat**
SQL server : select DATEDIFF(MM, '2013-05-17 00:00:00','2014-06-14 00:00:00')	13
Oracle client 10g : select MONTHS_BETWEEN('14-06-2014','17-05-2013') from dual	12,90
Adaptation des requêtes Oracle	
Requête	**Résultat**
select MONTHS_BETWEEN(LAST_DAY('14-06-2014'),LAST_DAY('10-05-2013')) from dual	13
select MONTHS_BETWEEN(LAST_DAY('14-06-2014'),LAST_DAY('17-05-2013')) from dual	13

Le Tableau 10 montre les requêtes et les résultats SQL Server et Oracle pour calculer la « différence en mois » entre deux dates. Des fonctions similaires sont appelées avec des résultats différents. Les requêtes Oracle sont adaptées afin de retourner le même résultat que la fonction de base de SQL Server.

3.6.3 Traitement d'événements

Dans un point de vue logique, un évènement est une notion utilisée pour faire du raisonnement temporel. C'est l'analogue du fait, mais il s'en différencie car il n'a pas de durée. Ainsi, les événements se produisent tandis que les faits se découvrent et durent. Nous avons présenté précédemment quelques modèles de raisonnement qui utilisent le concept d'événement afin d'arriver à des conclusions. En revanche, cette section est consacrée aux événements qui expriment des choses qui sont détectables ; point de vue depuis lequel il n'y a pas de différence entre des événements et des faits.

3.6.3.1 Types d'événements

La détection d'un *événement* peut donner lieu à une réaction ou un effet. Les événements sont *primitifs*, *conditionnels*, ou *complexes*. Un événement *primitif* a lieu après qu'une action spécifique s'est produite. Un événement *conditionnel* est exprimé par une condition impliquant la valeur affectée par l'action. Les événements sont *complexes* s'ils sont définis par la composition d'événements *primitifs* ou *conditionnels* et si la composition exprime des relations entre les événements.

Exemple

L'insertion d'une valeur dans une table de la base de données, la lecture d'une valeur par un capteur, l'initialisation d'une session, sont des événements primitifs.

Si une valeur supérieure à 38 est insérée dans la table température, si la valeur lue par le capteur est supérieure à 38, si c'est l'utilisateur Dupont qui initialise sa session, etc. sont des événements conditionnels.

Si la température est supérieure à 38 « pendant » que la tension est à la hausse « après » avoir administré du paracétamol au patient, est un événement complexe.

3.6.3.2 Détection d'événements

Le traitement d'événements est proche de différents paradigmes tels que l'*architecture orientée services* pour utiliser des sources de données hétérogènes,

l'*architecture orientée événements* qui facilite la détection des événements, la *gestion de processus métier* (BPM) qui exprime comment interagissent les événements, ou l'*intelligence en temps réel* qui permet de détecter des patterns d'événements sur des flux d'événements afin de les notifier le plus tôt possible.

Des nombreuses technologies sont utilisées pour le traitement d'événements complexes (Drools, 2013), (Storm, 2013), (Cayuga, 2013), (Esper, 2013), (Sybase Event Stream Processor, 2013). Elles sont basées sur des architectures orientées événements afin *d'écouter en permanence* les événements et définir des événements complexes. Ecouter en permanence est très important lorsque le temps de réponse (l'effet de la réaction à l'événement) doit être proche du temps réel. Ceci implique souvent l'utilisation de la mémoire volatile pour des calculs et des réponses plus rapides. Les événements complexes définis à l'aide de ces technologies (frameworks) sont très souvent fixés par des développeurs puisqu'il se peut que pour écouter en permanence une source de données, il faille modifier le code de l'application.

3.7 Synthèse

Différentes techniques permettent de considérer une situation d'alerte comme telle. Des approches logiques concluent à partir des faits et d'un ensemble de règles qu'une situation d'alerte se produit. Des approches probabilistes estiment la probabilité avec laquelle une situation d'alerte peut se produire. Des techniques d'apprentissage automatique permettent de définir des paramètres que subordonnent des données utilisées pour décrire une situation d'alerte spécifique et qui permettent au système de déterminer l'occurrence d'une situation d'alerte. La logique floue est utilisée lorsque la participation des utilisateurs quelconques est souhaitée pour définir les règles qui permettent au système de déterminer des situations d'alerte ainsi que les actions à prendre suite à la détection.

Nous avons vu l'importance de l'aspect temporel pour faire du raisonnement. Des situations d'alerte représentées par des primitives temporelles, telles que les points et les intervalles, sont associées à des concepts temporels comme les événements, les actions, ou les faits. L'utilisation explicite et implicite des opérateurs et relations temporels permet d'exprimer les interactions entre les primitives temporelles pour mieux décrire une situation ou un état d'alerte. Bien que l'utilisation d'opérateurs temporels peut aider à mieux décrire une situation d'alerte, cela peut rendre plus difficile la tâche des systèmes à cause des ambiguïtés possibles lors de l'exploitation des données.

La notion d'événement est largement utilisée. De nombreuses significations lui sont attribuées d'un point de vue rationnel, comme le début d'une propriété à un instant donné et la fin d'une autre. D'un point de vue opérationnel, un événement est quelque chose qui peut se détecter, il sert à définir les conditions d'activation des alertes qui décrivent la situation d'alerte.

Nous avons présenté comment les systèmes de gestion de bases de données fournissent des fonctions temporelles pour travailler avec les horodatages et faciliter

ainsi la prise en compte du temps dans la logique-métier des applications. Des architectures événementielles permettent de détecter différents types d'événements détectables en temps réel pour exploiter les informations efficacement.

Les séries chronologiques, formées par un ensemble d'observations d'un même type, ordonnées dans le temps, sont segmentées afin d'exploiter les segments dans des finalités diverses. Les algorithmes de segmentation et les segments trouvés sont soumis à de nombreuses conditions avec des finalités variées : si le système peut accéder aux données du passé et du futur (offline), ou seulement aux données du passé (online), si la série chronologique est sensible au bruit, si les segments doivent avoir la même taille, s'il faut les catégoriser par rapport à des propriétés, si c'est dépendant ou indépendant du domaine d'application, etc. Les séries chronologiques sont souvent considérées comme étant régulières lors de la détection des segments. Ceci est souvent vrai lorsque les dispositifs sont fiables (comme dans les unités de soins intensifs). Par contre, les algorithmes ne font pas de différence lorsque la série chronologique est constituée de données observées avec des fréquences différentes. En outre, la classification des segments dans une forme spécifique comme « à la hausse » ou « stable » est établie à l'avance, dans la plupart des cas par les concepteurs des algorithmes en limitant ainsi la participation des utilisateurs.

Le contexte est toute information qui permet de mieux caractériser une situation. Un système est dit sensible au contexte s'il utilise le contexte pour améliorer la pertinence des résultats qu'il produit. Une dimension intérieure prend en compte la situation de l'utilisateur et une autre, celle de l'environnement. Le contexte et la situation qu'il représente peuvent être définis sous une forme explicite, ou être abstraits par les systèmes afin d'adapter automatiquement les applications qui l'utilisent.

Les approches où le contexte qui représente une situation est abstrait à partir d'une grande quantité de données nécessitent de connaître à l'avance les types de données qui vont être utilisés et les caractéristiques qu'ils peuvent représenter afin d'entraîner des algorithmes de classification qui vont déterminer la situation. Dans ces systèmes, l'adaptation en temps réel consiste à adapter les actions par rapport à la situation identifiée, et non, à adapter la situation connue par l'utilisateur par rapport aux nouvelles informations disponibles.

Pour finir, nous concluons qu'aucun modèle utilisé pour la définition et l'évaluation des situations d'alerte n'est parfait. Cependant, quelques-uns sont plus pertinents et utiles que d'autres.

3.8 Positionnement

Nous proposons un système d'alertes dont les concepteurs des situations d'alerte et les consommateurs d'alertes détectées n'ont pas besoin d'une double compétence en informatique ou mathématiques. Le concepteur est de préférence l'utilisateur qui est expert dans le domaine d'application du système d'alertes et il est capable de décrire les situations d'alerte qui l'intéressent.

Les situations d'alerte sont définies utilisant des conditions formées par des valeurs numériques et linguistiques. Les valeurs linguistiques expriment l'état d'une variable, calculé à partir d'une seule observation, ou la tendance, calculée à partir de plusieurs observations. Implicitement, les primitives temporelles associées aux observations sont concurrentes pour exprimer qu'elles se sont produites les unes proches des autres. La relation temporelle « commence avant ou en même temps que » sert à exprimer l'ordre partiel des points, intervalles et semi-intervalles afin de mieux décrire les situations d'alerte si cela est nécessaire. Le système laisse à l'utilisateur le soin de définir des situations complexes avec des règles simples.

Une situation d'alerte est un contexte spécifique qui intéresse les utilisateurs et qui est décrit par nombreuses informations spécifiées par les utilisateurs. Chaque situation d'alerte peut être modifiée en temps réel en ajoutant des nouvelles informations pour mieux décrire la situation d'alerte, ou en modifiant le traitement des informations déjà connues. Tous les changements prennent effet dans l'immédiat.

L'évaluation d'une situation d'alerte génère éventuellement une alerte détectée à laquelle lui sont associés deux indices de qualité. L'indice de *confiance*, exprime dans quelle mesure l'utilisateur peut faire confiance aux données utilisées lors de l'évaluation de la situation d'alerte par rapport à sa fraîcheur. L'indice d'*applicabilité*, exprime dans quelle mesure l'entité (un patient, une maison, un bureau, etc.) est concernée par l'alerte. Les indices sont indépendants du domaine d'application et les métriques utilisées pour les calculer sont dépendantes du contexte exprimé par la situation d'alerte et les connaissances préparées par les utilisateurs. Les indices sont utilisés par les utilisateurs pour filtrer des alertes afin de trouver le meilleur compromis entre les faux-négatifs et les faux-positifs aux yeux des utilisateurs.

Nous proposons une politique de notification qui permet de réduire la notification des alertes et prioriser celles qui comptent pour les utilisateurs. Les nombreuses alertes détectées lors de l'évaluation d'une même situation d'alerte sur une même entité sont réduites à une seule par des mécanismes de filtrage et d'agrégation. Dans le cas de notifications d'alertes consécutives, seules celles qui comptent pour l'utilisateur sont notifiées explicitement, par exemple, lorsque l'alerte qui vient d'être détectée concerne plus qu'avant le patient.

TEMPAS est un système d'alertes indépendant du domaine d'application, dont la valeur ajoutée est sa simplicité et sa flexibilité bien que cela puisse affecter son exactitude. En effet, TEMPAS n'est pas un système expert mais un système adaptable aux besoins changeants et propres des utilisateurs à chaque contexte d'utilisation.

Dans le Tableau 11, nous comparons notre position par rapport aux travaux étudiés.

Tableau 11. Positionnement de TEMPAS

Travaux étudiés	TEMPAS
Raisonnement	
Nous avons étudié des travaux donc les systèmes déterminent qu'une situation d'alerte se produit à partir des modèles logiques, probabilistes, statistiques, et flous.	TEMPAS s'appuie sur la logique floue pour éviter des erreurs de classification et déterminer qu'une situation d'alerte se produit, au moins partiellement. En revanche, notre système a deux différences par rapport aux approches courantes fondées sur la logique floue. Dans TEMPAS, nous utilisons l'addition des conditions tandis que les approches courantes utilisent la conjonction de conditions. Dans TEMPAS, les utilisateurs définissent et adaptent en temps réel les seuils pour que TEMPAS puisse déterminer que c'est une vraie alerte et la notifie le cas échéant. C'est l'utilisateur qui donne la finalité à l'alerte qui lui a été notifiée. Dans les approches courantes c'est le système qui prend la meilleure décision qui a déjà été prédéfinie.
Logique Temporelle	
Nous avons étudié l'importance de l'aspect temporel dans le raisonnement. La représentation du monde par des ensembles de connaissances qui permettent de tirer de nouvelles connaissances concernant des actions, des faits, ou des événements	Etudier ces travaux nous a permis de définir des contraintes temporelles qui nous permettent de faire une confiance à l'état de l'univers décrit lorsqu'une situation d'alerte doit être déterminée.
Concepts et relations temporels	
Nous avons étudié des nombreux concepts temporels utilisés pour formaliser des connaissances tels que la durée, la synchronie ou la périodicité. Des primitives temporelles comme les points, les intervalles et les semi-intervalles, ainsi que des relations comme « avant » ou « chevauche » entre intervalles, et « plus vieux » ou « tête à tête » entre semi-intervalles. Nous avons aussi étudié des relations n-aires.	Les situations d'alerte dans TEMPAS sont définies à partir de plusieurs conditions d'activation vérifiées par des observations et des segments associés à des points et des intervalles temporels. En vue du caractère intrinsèque d'une 'alerte', une situation d'alerte TEMPAS exprime que les points et les intervalles sont concurrents. Toutefois, des conditions temporelles peuvent être utilisées pour exprimer un ordre partiel.

Travaux étudiés	TEMPAS
Gestion du temps	
Nous avons étudié des dimensions temporelles telles que le temps de validité, le temps de transaction, le temps d'événement, ou le temps de disponibilité ; dimensions utilisées pour mieux interpréter les données dans des bases de données.	Dans TEMPAS, en plus du temps de validité, nous utilisons le temps d'expiration qui nous permet de travailler avec une ambigüité similairement comme c'est fait dans la logique floue. Ainsi, si le temps de validité est dépassé mais pas le temps d'expiration nous considérons que l'information fournie par une observation n'est plus tout à fait vraie, mais elle n'est pas non plus fausse.
Séries chronologiques	
Les séries chronologiques sont approximées afin d'obtenir une représentation dont la finalité est souvent de faciliter leur interprétation. Nous avons étudié l'abstraction des tendances qui sont souvent prédéfinies comme « croissant », « stable », et « décroissant » et abstraites à partir de plusieurs paramètres comme le taux et le gradient. Nous avons aussi présenté les types d'algorithmes utilisés pour segmenter les séries chronologiques.	TEMPAS utilise un algorithme ascendant pour segmenter les séries chronologiques. Chaque segment trouvé est utilisé pour abstraire une ou deux tendances par rapport aux tendances définies par les utilisateurs. Ceux-ci définissent autant de valeurs linguistiques que nécessaire pour représenter les comportements des variables qui servent à définir des situations d'alerte. Pendant le processus de segmentation, TEMPAS calcule l'indice de confiance de la tendance qui exprime si la distance temporelle entre les observations consécutives qui composent le segment se trouve dans la distance temporelle valide selon le contexte d'application. Ainsi, cet indice remonte des informations concernant l'irrégularité des séries chronologiques qui n'est souvent pas prise en compte.

Travaux étudiés	TEMPAS
Contexte	
Nous avons étudié la notion de contexte défini comme toute information qui permet de mieux décrire une situation. Les travaux dits sensibles au contexte utilisent le contexte afin d'améliorer les services fournis à l'utilisateur. La notion de finalité est aussi associée au contexte, dont la dimension intérieure prend en compte les finalités de l'utilisateur et la dimension extérieure utilise les éléments de l'environnement.	TEMPAS est sensible au contexte. L'utilisateur utilise des valeurs linguistiques pour représenter les ressources contextualisées (l'état ou la tendance de la température, par exemple) et pour mieux définir les situations d'alerte qui concernent une entité (un patient, par exemple). Toutefois, c'est l'utilisateur qui définit la situation d'alerte par rapport à ses préférences. C'est lui qui donne la finalité qu'il veut donner à l'alerte. Dans le cas médical, les entités sont des patients et les utilisateurs des professionnels de santé. Ainsi, l'expressivité de TEMPAS permet de prendre en compte le contexte du patient ce qui génère des alertes de plus haute qualité qui satisfassent les finalités des médecins.

4 Présentation Générale de TEMPAS

4.1 Introduction 81
4.2 Définition et gestion d'alertes 81
4.3 Portée du système d'alertes 84
4.4 Données et connaissances 85
4.5 Indices de qualité 98
4.6 Politique de notification 116
4.7 Synthèse du chapitre 127

4.1 Introduction

TEMPAS est un système d'alertes adaptable au contexte et aux besoins des utilisateurs. Les situations d'alerte sont construites à partir de *valeurs linguistiques* issues de la *logique floue*. Par exemple, dans le domaine médical, une valeur linguistique peut se traduire par : *médicament administré* ou *tension systolique à la hausse*. L'évaluation d'une situation d'alerte mène, le cas échéant, à la concrétisation d'une alerte que nous nommons *alerte détectée*, ceci afin d'éviter des ambiguïtés. Une alerte détectée est liée à deux indices de qualité. *L'indice d'applicabilité* qui exprime la pertinence de l'alerte, et *l'indice de confiance* qui exprime la fiabilité des données utilisées lors de sa détection.

Nous présentons dans la section 4.2. TEMPAS comme un processus. Ensuite, dans la section 4.3 nous exposons la portée du système d'alertes, qui correspond à son contexte d'application. La section 4.4 présente les connaissances nécessaires qui permettent à TEMPAS de générer des alertes. Nous introduisons les indices de qualité et leur calcul dans la section 4.5. Nous présentons dans la section 4.6 la politique de notification qui s'appuie largement sur les indices de qualité afin de décider quelles alertes doivent être notifiées et attirer l'attention des utilisateurs. Finalement nous faisons une synthèse du chapitre dans la section 4.7.

4.2 Définition et gestion d'alertes

Nous présentons d'abord les processus impliqués dans le processus de TEMPAS. Ensuite, nous présentons les principaux concepts manipulés dans notre système d'alertes et qui sont utilisés tout au long du chapitre.

4.2.1 Processus

TEMPAS est un module indépendant qui ajoute des fonctionnalités d'alertes à des systèmes d'information existants dit systèmes maîtres. Nous différencions les trois processus dans la Figure 13 concernant le processus de TEMPAS dans un système d'information maître : la définition des situations d'alerte exprime comment TEMPAS

doit exploiter des données afin de décider si une situation d'alerte a, ou non, lieu ; l'évaluation des situations d'alerte définit le moment où TEMPAS doit exploiter les données ; et la notification d'alertes détectées exprime comment le système maître récupère et notifie les alertes détectées par TEMPAS.

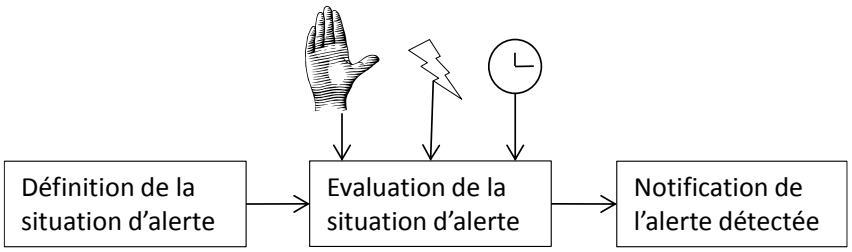


Figure 13. Processus impliquant les alertes

La *définition des situations d'alerte* permet à l'utilisateur d'exprimer ce que le système d'alertes doit évaluer et comment le faire. Le diagramme de cas d'utilisation de la Figure 14 représente l'interaction simplifiée entre un utilisateur concepteur des situations d'alerte (utilisateur expert) et le système lors de la définition/modification des situations d'alerte. L'utilisateur définit toutes les ressources impliquées dans une situation d'alerte. Il définit les valeurs linguistiques qui représentent les états et les tendances des ressources qui sont utilisées pour définir les conditions d'activation et pour calculer l'indice d'applicabilité de l'alerte. Pour chaque ressource impliquée, l'utilisateur définit le temps de validité et le temps d'expiration qui sont utilisés pour calculer l'indice de confiance de l'alerte. Finalement l'utilisateur définit le cycle de vie de l'alerte pour gérer la notification des alertes détectées. Ce processus est abordé plus en détail dans les sections 4.4 et 4.6.

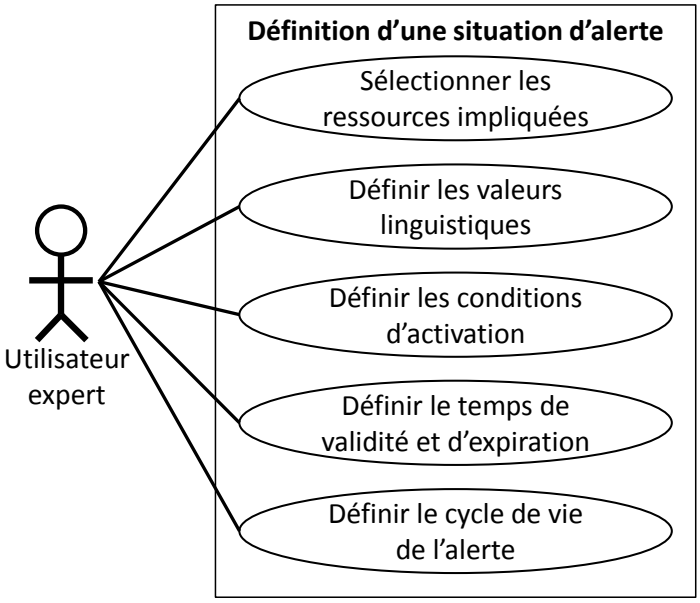


Figure 14. Diagramme de cas d'utilisation de la définition des situations d'alerte

L'*évaluation des situations d'alerte* correspond à vérifier les conditions d'activation de la situation d'alerte à un instant τ. La vérification d'une condition d'activation consiste en l'utilisation des données pour conclure si la condition est, ou non, satisfaite. Dans la section 4.5 nous présentons en détail les calculs pour arriver à ces conclusions.

La *notification d'alertes détectées* est un processus qui applique la politique de notification d'une situation d'alerte afin de conclure si l'alerte doit être notifiée et si la notification doit être implicite ou explicite. La notification explicite exprime qu'il est nécessaire d'attirer l'attention de l'utilisateur. La politique de notification est vue en détail dans la section 4.6.

4.2.2 Concepts dans TEMPAS

Une *alerte*, selon les interlocuteurs, peut représenter les facteurs qui la déclenchent ou son occurrence à un instant précis. Pour éviter toute confusion nous définissons l'ensemble des concepts suivants : *situation d'alerte, évaluation d'une situation d'alerte, alerte détectée, et alerte accessible*.

Définition 1 : Entité observable

Une entité observable correspond à l'objet ciblé par les situations d'alerte. Elle répond à la question, qui est observé ?

Exemple

les patients, dans le cas des situations d'alerte médicales, sont des entités observables

les ordinateurs, dans le cas des situations d'alerte concernant le réseau, sont des entités observables

Définition 2 : Ressource observable

Une ressource observable est tout ce qui peut être observé sur une entité observable. Elle répond à la question, qu'est-ce qui est observé ?

Exemple

données anthropométriques - taille
paramètres patients - température
résultats de biologie - glycémie

Définition 3 : Condition d'activation

Une condition d'activation sert à exprimer ce qui doit être satisfait afin de détecter une alerte. Elle est formée par une ressource observable, un opérateur, et une valeur.

Exemple

température = hyperthermie se lit « si hyperthermie »
Age ≥ « enfant » se lit « si enfant » ou « si adolescent » ou « si adulte » ou « si âgé »
tension systolique ≤ 90 mm Hg se lit « si la tension systolique est inférieure ou égale à 90 mm Hg»

Définition 4 : Situation d'alerte

Une situation d'alerte exprime une situation d'intérêt. Dans le cas spécifique de TEMPAS, une situation d'alerte est définie comme l'addition de conditions d'activation.

Exemple

« si gain de poids » et « si hypertension » alors alerte
« si médicament non administré » et « si hyperthermie » et « pouls élevé » alors alerte
« si tension systolique supérieure à 150 mm Hg » et « si nouveau-né » alors alerte

Définition 5 : Alerte détectée

Une alerte détectée est l'instanciation ou concrétisation après évaluation d'une situation d'alerte sur une entité observée à un instant τ.

4.3 Portée du système d'alertes

Nous exprimons la portée de TEMPAS à partir des deux hypothèses ci-dessous.

Hypothèse 1 : TEMPAS s'intègre à des systèmes d'information et considère que les données à utiliser pour l'évaluation des situations d'alerte sont fiables.

TEMPAS est un système qui s'intègre à d'autres systèmes d'information (dits systèmes maîtres). Des interfaces sont utilisées pour présenter les ressources observables disponibles pour la construction des situations d'alerte et pour faciliter l'exploitation des données au moment de leur évaluation. TEMPAS ne vérifie pas si les données utilisées sont correctes, s'il s'agit du bruit, etc. Il les considère comme étant fiables, leur vérification est déterminée dans le système source (dispositif de mesure, médiateur, système maître). En revanche, pour monitorer la qualité des données relevées, les utilisateurs peuvent se servir des alertes détectées avec des données non fiables (du bruit par exemple) et se rendre ainsi compte qu'il faut vérifier la source de captation des données.

Hypothèse 2 : Les conditions d'activation d'une situation d'alerte doivent être concurrentes.

L'hypothèse 2 exprime qu'une alerte est potentiellement détectée si les conditions d'activation d'une situation d'alerte sont satisfaites et sont temporellement concurrentes. Une situation d'alerte est exprimée par l'addition de conditions d'activation. Par exemple, si la température est élevée et le paracétamol administré et la tension systolique à la hausse alors l'alerte est détectée. L'expression de relations temporelles binaires (comme celles d'Allen) entre les conditions d'activation n'est pas prise en compte par le système d'alertes. On peut toutefois gérer un ordre partiel sur les conditions d'activation (commence avant ou en même temps que).

4.4 Données et connaissances

Les données sont exploitées par TEMPAS au moment de l'évaluation d'une situation d'alerte. Les connaissances procédurales permettent à TEMPAS de déterminer qu'une situation d'alerte se produit lors de l'exploitation des données à un instant précis.

4.4.1 Données

TEMPAS récupère et utilise des données lors de l'évaluation des situations d'alerte. Il s'agit par exemple des données produites par des capteurs, entrées manuellement, calculées par d'autres systèmes, etc. TEMPAS modélise les données comme étant des *observations* (Figure 15). Le Tableau 12 présente des exemples des observations interprétables par TEMPAS.

Définition 6 : Observation

Une observation est définie par une entité observée, une ressource observable (ce qui est observé), la date et l'heure à laquelle l'observation a eu lieu, et la valeur observée qui peut être quantitative ou linguistique.

Soit O *l'ensemble des observations,*
Soit E *l'ensemble des entités observables,*
Soit R *l'ensemble des ressources observables,*
Soit t *un horodatage,*
Soit v *une valeur observée,*
L'observation i *est définie comme*

$$i: < e, r, t, v >, i \in O, e \in E, r \in R$$

Exemple

la température du patient Ulrich est de 37°C le 23/01/2014 à 17h30
l'humidité relative de Grenoble est de 43% le 23/01/2014 à 17h
le paracétamol a été administré au patient Brunet le 23/01/2014 à 18h
le patient Ulrich est né le 20/04/1988
le patient Ulrich est un homme

L'horodatage est inutile si une seule observation sert à exprimer la réalité d'une ressource observable pour une entité. Par exemple, si la valeur ne change pas dans le temps (comme le sexe), ou si la valeur fournie par le système maître est dépendante de l'instant de consultation comme l'âge ou la dernière prise du paracétamol (il y en a une seule dernière), la valeur est calculée en utilisant la date de consultation. Nous y reviendrons dans la section d'évaluation des alertes.

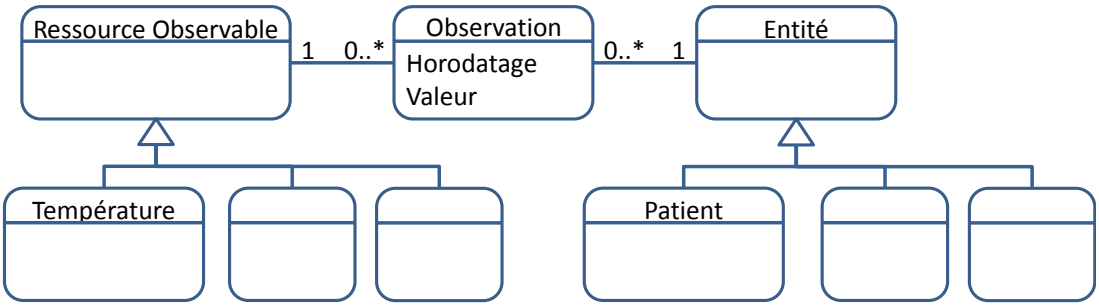


Figure 15. Modélisation d'une observation

Tableau 12. Exemple d'observations exploitables par TEMPAS

Entité	Ressource Observable	Valeur	Horodatage
Ulrich	température	37°C	23/01/2014 à 17h30
Grenoble	humidité relative	43%	le 23/01/2014 à 17h
Brunet	paracétamol	administré	23/01/2014 à 18h
Ulrich	est né	20/04/1997	–
Ulrich	sexe	homme	–
Ulrich	âge	17 ans	–
Brunet	Dernière prise du paracétamol	8 heures	–

4.4.2 Introduction des valeurs linguistiques

L'utilisation des valeurs linguistiques augmente l'expressivité des alertes. Les utilisateurs donnent ainsi du sens aux valeurs observées. Ainsi le système est capable d'exprimer que l'humidité relative de Grenoble de 60% est « élevée ». Il parvient à une telle conclusion grâce à *la logique floue*. La logique floue permet de calculer une *valeur linguistique* à partir d'une *valeur numérique* et de gérer l'ambigüité comme un humain le fait. Par exemple, déterminer qu'une humidité relative de 50% est « normale » et « élevée » en même temps, mais plus « élevée » que « normale ». La logique floue s'appuie sur la théorie des ensembles flous de (Zadeh, 1965).

Définition 7 : <u>Valeurs linguistiques d'une ressource observable</u>

L'état ou la tendance d'une ressource observable peut s'exprimer par des valeurs linguistiques. Une valeur linguistique est représentée par un ensemble flou de forme trapézoïdale auquel lui est associée une fonction d'appartenance.

Soit r une ressource observable
Soit L l'ensemble de valeurs linguistiques exprimant l'état ou la tendance de r
Soit μ l'ensemble de fonctions d'appartenance aux valeurs linguistiques dans L

$$r : < L, \mu >, L = \{l_0, \dots, l_i, \dots, l_n\}, \mu = \{\mu_{l_0}, \dots, \mu_{l_i}, \dots, \mu_{l_n}\}$$

Exemple

les valeurs linguistiques « bébé », « enfant », et « jeune » expriment l'état de la ressource observable « âge »

les valeurs linguistiques « à la baisse », « stable », et « à la hausse » expriment la tendance de la ressource observable « tension»

Les ressources observables ont des propriétés qui permettent au système d'alertes de mieux interpréter les observations, ou les segments des ressources observables. La Figure 16 illustre, sous une forme générique, n ensembles flous représentant n valeurs linguistiques pour une *ressource observable*. Notons que l'intersection entre deux *ensembles flous* n'est pas vide. Les valeurs linguistiques sont ordonnées de la même manière que les ensembles flous qui les représentent.

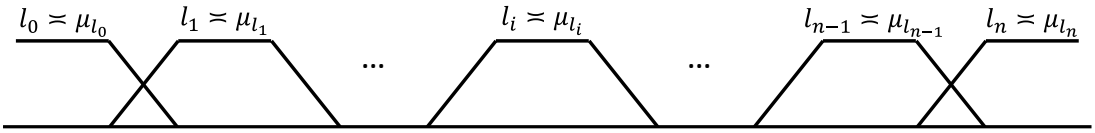


Figure 16. Valeurs linguistiques d'une ressource observable

Définition 8 : <u>Ordre des valeurs linguistiques</u>

Les valeurs linguistiques sont naturellement ordonnées.

Soit L l'ensemble de valeurs linguistiques exprimant l'état ou la tendance

$\forall\ l_0, l_1 \in L$
$l_0 < l_1 ssi\ \exists\ x_0, x_1 \in R, x_0 < x_1, \mu_{l_0}(x_0) = 1, \mu_{l_1}(x_1) = 1$

Dans TEMPAS, les *fonctions d'appartenance* associées aux ensembles flous sont des fonctions affines par morceaux. La Figure 17 montre, génériquement, comment elles sont définies à partir de la forme de l'ensemble flou. La *fonction d'appartenance* retourne le *degré d'appartenance*.

Définition 9 : <u>Degré d'appartenance</u>

Le degré d'appartenance est une valeur entre zéro et un qui exprime le degré de satisfaction de la transformation d'une valeur quantitative en une valeur linguistique donnée.

Exemple

$\mu_{normal}(37.7) = 0.4$ exprime le degré de satisfaction de la transformation de la valeur numérique 37.7 dans la valeur linguistique « normal » pour la température

$\mu_{élevée}(37.7) = 0.6$ exprime le degré de satisfaction de la transformation de la valeur numérique 37.7 dans la valeur linguistique « élevée » pour la température

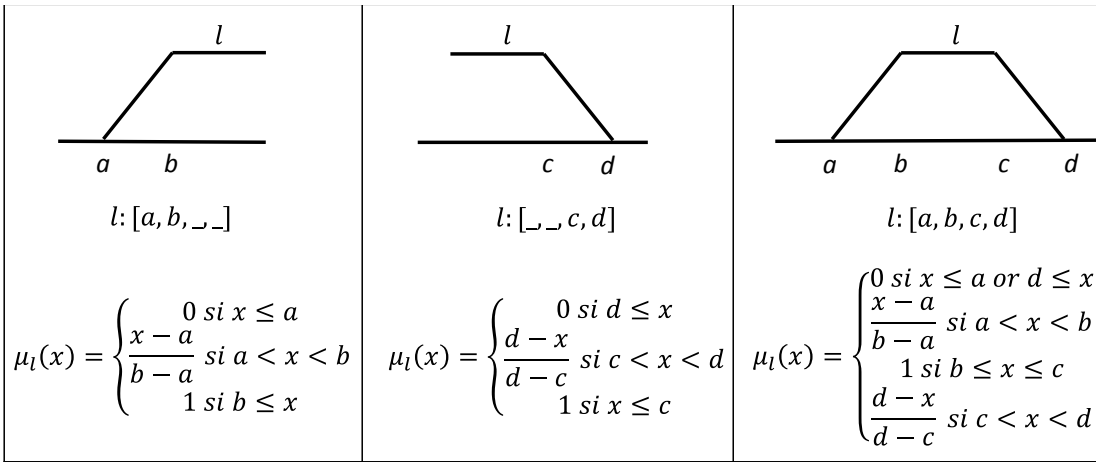


Figure 17. Fonction d'appartenance correspondante aux ensembles flous

4.4.2.1 Etat de la ressource

Les *valeurs linguistiques* qui expriment l'*état* d'une ressource sont calculées à partir de la valeur observée. Imaginons que les valeurs linguistiques « bébé », « enfant », et « jeune » expriment l'*état* de la *ressource observable* « âge » sur les patients de pédiatrie. La Figure 18 illustre les *fonctions d'appartenance* associées à chacune de ces valeurs linguistiques. Le résultat de chaque fonction pour la valeur quantitative 13 est 0, 0.25, et 0.75 respectivement. Ainsi, un patient de 13 ans est considéré comme un « enfant » et un « jeune » en même temps, mais plus « jeune » qu'« enfant ». Le Tableau 13 montre le passage des valeurs quantitatives observées en valeurs linguistiques tout en reflétant, avec le *degré d'appartenance*, qu'une même valeur quantitative peut satisfaire deux valeurs linguistiques.

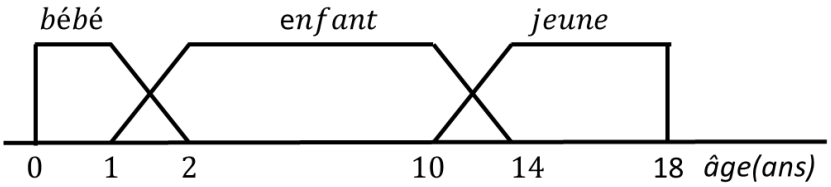


Figure 18. Valeurs linguistiques pour l'âge.

Exemple

$$\mu_{bébé}(13) = 0, \mu_{enfant}(13) = 0.25, \mu_{jeune}(13) = 0.75$$

Tableau 13. Transformation de valeurs quantitatives en valeurs linguistiques

Entité	Ressource Observable	Valeur	Horodatage
Ulrich	température	37°C	23/01/2014 à 17h30
Ulrich	âge	17 ans	–

Entité	Ressource Observable	Horodatage	Valeur Linguistique	Degré d'appartenance
Ulrich	température	23/01/2014 à 17h30	Hypothermie	0
Ulrich	température	23/01/2014 à 17h30	Normale	1
Ulrich	température	23/01/2014 à 17h30	Hyperthermie	0
Ulrich	âge	–	Enfant	0
Ulrich	âge	–	Jeune	0.25
Ulrich	âge	–	Adulte	0.75
Ulrich	âge	–	Agé	0

4.4.2.2 Tendance de la ressource

TEMPAS utilise aussi des valeurs linguistiques qui expriment les *tendances* des ressources observables. Les tendances sont des valeurs linguistiques associées à un *segment* formé par deux observations. Les segments sont modélisés dans la Figure 19.

Définition 10 : Segment

Nous interprétons un segment comme l'approximation de toutes les observations dont l'horodatage est inclus dans l'horodatage du segment.

Soient i_1 ,i_2 *deux observations*
Le segment seg *est défini comme*

$$seg: < i_1 , i_2, v >, i_1: < e, r, t_1, v_1 >, i_2: < e, r, t_2, v_2 >, t_1 < t_2$$

Où v *est une propriété du segment*

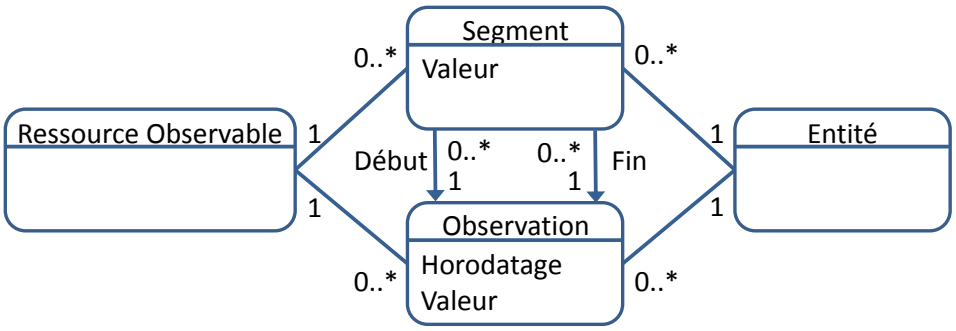


Figure 19. Modélisation d'une tendance

L'horodatage du segment est l'intervalle formé par les horodatages des observations que le délimitent. Enfin, les valeurs linguistiques qui expriment la tendance sont calculées à l'aide de fonctions d'appartenance qui utilisent une valeur propre du segment. Par exemple, dans le Tableau 14, les valeurs linguistiques du segment sont calculées avec son angle.

Tableau 14. Transformation des segments en valeurs linguistiques

Entité	Ressource Observable	Horodatage	Angle
Ulrich	température	[23/01/2014 à 12h30 - 23/01/2014 à 17h30]	30°

Entité	Ressource Observable	Horodatage	Valeur Linguistique	Degré d'appartenance
Ulrich	température	[23/01/2014 à 12h30 - 23/01/2014 à 17h30]	Tendance à la baisse	0
Ulrich	température	[23/01/2014 à 12h30 - 23/01/2014 à 17h30]	Stable	0
Ulrich	température	[23/01/2014 à 12h30 - 23/01/2014 à 17h30]	Tendance à la hausse	1

4.4.3 Définition des situations d'alerte

Nous avons présenté comment TEMPAS utilise les données afin de calculer des valeurs linguistiques. Dans cette section nous expliquons comment ces valeurs linguistiques sont utilisées pour définir les conditions d'activation d'une situation d'alerte. La Figure 20 montre le positionnement du processus de *définition de la situation d'alerte* par rapport aux autres processus impliqués dans la gestion d'alertes.

Figure 20. Définition d'une situation d'alerte

Une situation d'alerte est une situation qui intéresse un seul utilisateur ou un groupe d'utilisateurs. Beaucoup de systèmes d'alertes classiques définissent des alertes à partir d'une seule condition d'activation ou d'une condition de déclenchement de l'alerte : par exemple, si un bouton a été activé, si la tension systolique dépasse les 140 mm Hg, si la mémoire ram disponible est inférieure à 10%, etc., contrairement à TEMPAS. Nous reviendrons dans la section 4.5.3.1 sur le poids des conditions d'activation.

Raffinement de la Définition 4 — Situation d'alerte

Une situation d'alerte est définie comme l'addition des conditions d'activations auxquelles est associé un poids vis-à-vis de la situation d'alerte. La somme des poids est égale à 1.

Soit a une situation d'alerte
Soit c une condition d'activation
Soit ρ_c le poids de la condition d'activation c

$$\sum \rho_c = 1$$
$$a = +(c, \rho)$$

Les situations d'alerte sont définies en utilisant deux types de conditions d'activation : les conditions activées à partir de l'état ou la tendance d'une ressource observable appelées *conditions intrinsèques*, et des conditions qui expriment des contraintes temporelles appelées *conditions temporelles*.

Définition 11 : — Condition d'activation intrinsèque

Une condition intrinsèque exprime l'état dans lequel une ressource observable doit se trouver ou comment la tendance doit se comporter afin de pouvoir être considérée en état d'alerte.

Soit r un Ressource
Soit L l'ensemble des valeurs linguistiques exprimant l'état ou la tendance de r
La condition d'activation est c définie comme

$$c: < r, op, l_{ref} >, op \in \{\leq, =, \geq\}, l_{ref} \in L$$

Exemple

La condition « si température ≥ normale » se lit « si la température est normale ou élevée »
La condition « si tension ≤ normale » se lit « si la tension est normale ou hypotension »

Les valeurs linguistiques de L sont ordonnées (Définition 8). Il est possible de définir des conditions d'activation intrinsèques, en utilisant des valeurs numériques ou linguistiques, qui appartiennent à l'ensemble de valeurs observables d'une ressource. Si la condition d'activation est formée par une valeur linguistique observable alors le seul opérateur utilisable est l'égalité.

Exemple

« sexe = homme »
Si « Pouls ≤ 90 »

Les conditions temporelles sont facultatives puisqu'une situation d'alerte exprime implicitement des conditions d'activation concurrentes. Cependant, l'expression des contraintes temporelles entre les ressources (r) utilisées dans la définition des conditions intrinsèques améliore la définition des situations d'alerte. Une condition $r_1 \leq r_2$ exprime que la dimension temporelle (horodatage ou intervalle) de l'observation ou du segment de la ressource r_1 *commence avant ou en même temps que* celle de la ressource r_2. La Figure 21 montre l'interprétation de cette relation entre points, intervalles, et semi-intervalles temporels.

Définition 12 : <u>Condition d'activation temporelle</u>

Une condition temporelle exprime l'ordre d'occurrence des horodatages des observations ou des segments des ressources observables utilisés dans la construction de conditions d'activation intrinsèques.

Soient r_1 *et* r_2 *deux ressources observables*
La condition temporelle c *est définie comme*

$$c: < r_1 \leq r_2 > \; ssi \; \exists \; c_1, c_2, c_1: < r_1, op, v >, c_2: < r_2, op, v >$$

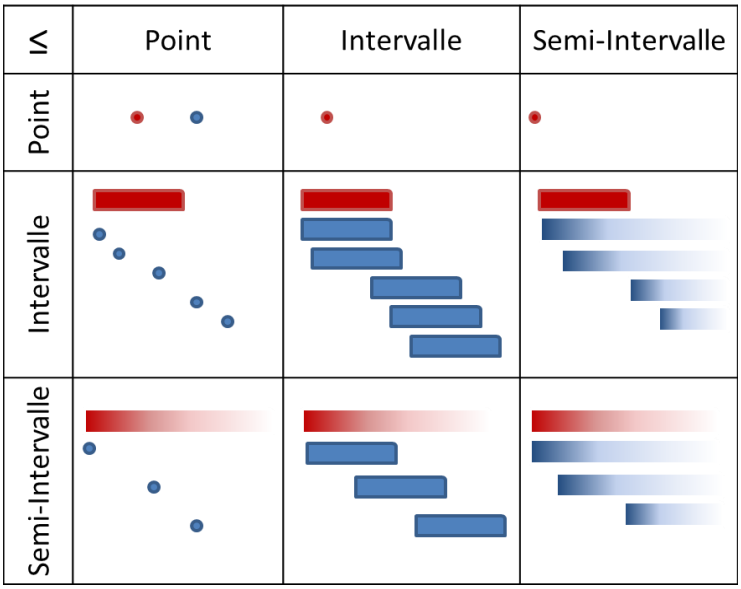


Figure 21. Relation « commence avant ou en même temps que »

Supposons une situation d'alerte destinée à surveiller la déshydratation possible des personnes âgées chaque fois qu'une vague de chaleur s'abat sur la France. Elle est définie avec des *ressources observables* telles que la « température » et l'« humidité extérieure », la « température corporelle » et l'«'âge » des citoyens. La dernière condition d'activation, *temporelle*, exprime que l'alerte est plus conséquente s'il n'y a pas eu d'hydratation une fois que la température corporelle a commencé à monter.

Exemple

$$Alerte\ de\ canicule : (< température\ extérieure\ =\ élevée >, p_1)$$
$$et\ (< humidité\ \geq\ normale >$$
$$et\ < âge\ =\ âgé >, p_2)$$
$$et\ (< tendance\ température\ corporelle =\ à\ la\ hausse >, p_3)$$
$$et\ (< hydratation =\ faux >, p_4)$$
$$et\ (< tendance\ température\ corporelle \leq hydratation >, p_5)$$

4.4.4 Expression du contexte

Nous utilisons la définition de contexte proposé par (Dey, 2001) qui est très expressive et très répandue dans la littérature. *« Le contexte est toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une personne, un endroit, ou un objet qui est considéré pertinent à l'interaction entre un utilisateur et une application, y compris l'utilisateur et les applications elles-mêmes »*. Dans cette section, nous nous concentrons sur la caractérisation des situations dans le cadre des alertes.

4.4.4.1 Informations contextuelles de la ressource observable

Une ressource n'est pas observable de la même façon sous des contextes différents. Dans le cadre d'un système d'alertes, nous nous focalisons sur deux aspects ; d'abord, la fréquence des observations et ensuite les valeurs linguistiques. Nous abordons ces aspects dans les sections ci-dessous.

4.4.4.1.1 Fréquence des observations

Une observation exprime l'état d'une ressource observable (quoi) sur une entité (qui) à un instant précis. Pour la plupart de ressources observables, la réalité exprimée avec une observation n'est pas constante et elle est susceptible de changer dans le temps. Le système d'alertes utilise les informations contextuelles de la ressource observable pour savoir si les observations utilisées expriment une réalité toujours d'actualité. En général, ces informations sont connues par le système d'information. Par exemple, le patient Gauthier, à domicile, doit mesurer son pouls toutes les 8 heures. A la maison de retraite, les constantes médicales (y compris le pouls) sont relevées toutes les heures. A la clinique, le pouls est mesuré toutes les 15 minutes. Bien qu'il s'agisse de la même ressource observable, elle n'est pas mesurée avec la même fréquence parce que le contexte n'est pas le même. Trois observations par jour expriment la situation médicale d'un patient à domicile alors qu'un suivi plus fréquent est réalisé dans un établissement de santé.

Nous définissons deux informations contextuelles exploitables par le système d'alertes : le *temps de validité* et le *temps d'expiration*. La durée du temps d'expiration est supérieure ou égale à celle du temps de validité. La Figure 22 montre la représentation de ces deux durées pour une observation (ι).

Définition 13 : Temps de validité

Le temps de validité (φ) de la ressource observable exprime pendant combien de temps une observation est considérée comme étant actuelle et toujours fiable lors de son utilisation dans l'évaluation d'une alerte

Définition 14 : Temps d'expiration

Le temps d'expiration (ψ) exprime après combien de temps une observation devient obsolète. Outre le fait qu'elle n'est plus actuelle, s'en servir pour évaluer une alerte peut introduire une erreur.

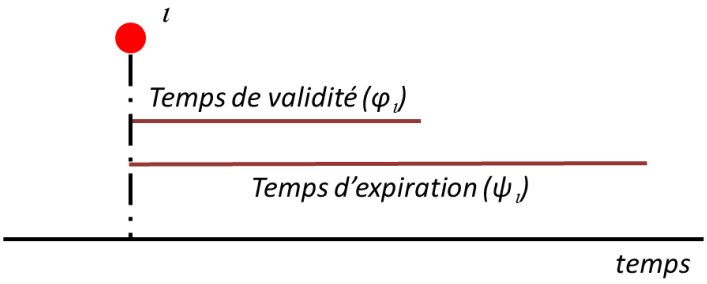


Figure 22. Représentation du temps de validité et du temps d'expiration.

Le *temps de validité* et le *temps d'expiration* sont dépendants du contexte, donc de la ressource observable et l'entité observée. Ainsi, une *ressource observable* sous un *contexte* spécifique définit un *temps de validité* et un *temps d'expiration* utilisés pour toutes les *observations* du sous-ensemble d'entités faisant partie du contexte. La Figure 23 montre le temps de validité et d'expiration de la température corporelle pour les patients dans le bloc opératoire et la salle de réveil respectivement. Les boîtes à tirets représentent une copie d'une entité afin de faciliter la lecture de la figure.

Raffinement de la Définition 7 Informations contextuelles

Les informations contextuelles permettent de mieux définir une situation (d'alerte) pour une entité.

Soit c un contexte
Soit $\mathcal{R}^+$ l'ensemble de réels supérieurs ou égaux à 0

$$r_c : < \varphi, \psi >, c \in C, \varphi \in \mathcal{R}^+, \psi \in \mathcal{R}^+, \varphi \leq \psi$$

Exemple

Un temps de validité et d'expiration de 5 et 30 secondes pour les patients au bloc opératoire

Un temps de validité et d'expiration de 15 et 25 minutes pour les patients en salle de réveil

Dans le cas où une seule observation d'une ressource observable suffit pour exprimer à l'infini l'état de la ressource observable pour une entité (par exemple le sexe ou la date de naissance d'une personne), le temps de validité et d'expiration sont infinis dans n'importe quel contexte.

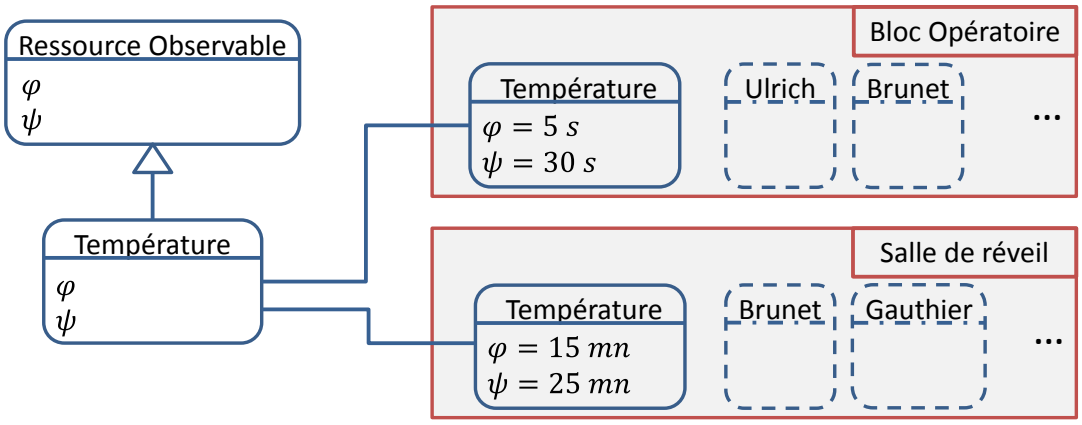


Figure 23. Temps de validité et d'expiration dans deux contextes différents

4.4.4.1.2 Valeurs linguistiques et ensembles flous

Les valeurs linguistiques pour référencer l'âge d'un patient peuvent être « enfant », « jeune », « adulte », « âgé ». Pour les patients du service de pédiatrie d'un hôpital, des valeurs telles que « nouveau-né », « nourrisson », « enfant », « adolescent » sont plus appropriées. Ainsi, les valeurs linguistiques pour une même ressource observable peuvent être distinctes pour deux contextes différents. Les valeurs linguistiques peuvent aussi être identiques dans plusieurs contextes. Cela ne signifie pas qu'elles sont calculées de la même façon.

Raffinement de la Définition 7 — Informations contextuelles

Soit c un contexte
Soit r_c une ressource observable sous contexte c
Soit L l'ensemble de valeurs linguistiques exprimant l'état ou la tendance de r
Soit μ l'ensemble de fonctions d'appartenance aux valeurs linguistiques dans L
Soient φ et ψ les temps de validation et d'expiration de la ressource r

$$r_c : < L, \mu, \varphi, \psi >, L = \{l_0, \dots, l_i, \dots, l_n\}, \mu = \{\mu_{l_0}, \dots, \mu_{l_i}, \dots, \mu_{l_n}\}$$

Regardons la température corporelle, une constante médicale très utilisée mais qui n'est pas toujours interprétée de la même façon. Par exemple, une température corporelle est considérée comme « normale » si elle se trouve entre 36.1°C et 37.8°C (Simmers, et al., 2008). Elle est aussi considérée « normale » pour un homme si elle se trouve entre 36.9±4°C ou entre 36.7±4°C pour une femme (Shoemaker , 1996). Il faut ajouter à cette incertitude la façon dont elle est mesurée : orale, rectale, tympanique, ou axillaire. Nous enrichissons la formalisation de la ressource observable dans un

contexte spécifique. Une ressource observable spécifie un ensemble de valeurs linguistiques (L), les fonctions d'appartenance pour les calculer (μ), et le temps de validité (φ) et d'expiration (ψ) pour l'interprétation de toute observation de la ressource observable sur une entité.

4.4.4.2 Ressources observables et construction du contexte

Nous proposons une approche générale où le contexte permet de mieux définir la situation d'alerte. Dans TEMPAS, le contexte est construit principalement par des ressources observables. Prenons l'exemple de la situation d'alerte de la section 4.4.3 concernant la canicule. Des informations sur le sexe enrichissent le contexte puisque la « température corporelle » est « élevée » avec des seuils différents pour les « hommes » et pour les « femmes ». Donc, des informations autres que celles faisant partie de la définition de la situation d'alerte permettent au système de mieux définir le contexte. Dans le cas de TEMPAS, le *contexte* ne prend pas seulement en compte les propriétés associées aux ressources observables mais aussi comment les ressources interagissent entre elles.

Définition 15 : Interaction entre ressources observables

L'interaction entre ressources observables exprime comment les valeurs linguistiques d'une ressource affectent les valeurs linguistiques d'une autre.

Il n'est pas tout à fait correct d'affirmer qu'une personne pesant 100 Kg est une personne « obèse » sans utiliser des informations telles que son « sexe », son « âge », ou sa « taille » ; ou que la « température » d'une personne est « élevée » sans savoir s'il s'agit d'un « homme » ou d'une « femme ». TEMPAS utilise un graphe modificateur afin de mieux représenter les valeurs linguistiques d'une ressource exprimant un état (Manzi de Arantes Junior, 2006). Le graphe illustré dans la Figure 24 contient les ressources nécessaires pour le calcul des valeurs linguistiques de l'exemple de la section 4.4.3.

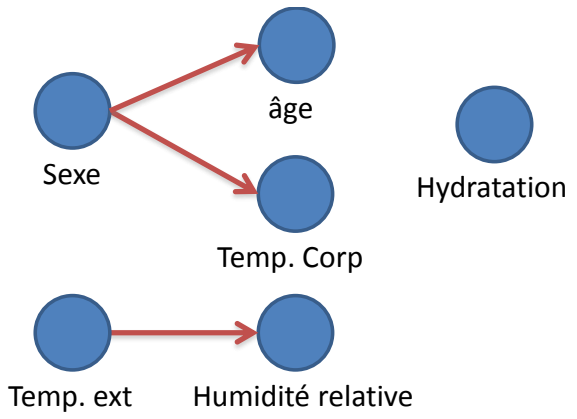


Figure 24. Exemple de graphe modificateur

Les *nœuds* représentent les *ressources observables*. Un *arc* exprime que les *valeurs linguistiques* d'une ressource affectent les *valeurs linguistiques* d'une autre. Le « sexe » affecte donc l'« âge » et la « température » d'une personne, et la « température extérieure » affecte l'« humidité ». Dans le cas du « sexe » et de l'« âge », par exemple, un homme est considéré « âgé » à partir de « 64 ans » tandis qu'une « femme » l'est à partir de « 63 ans ». Pour la « température extérieure » et « l'humidité relative », plus il fait « chaud », plus l'air est chaud, et plus il peut contenir de vapeur d'eau. Donc considérer l'« humidité relative » « élevée » dépend de s'il fait « chaud » ou « froid ».

Le graphe modificateur permet d'exprimer un type spécial de connaissances afin de mieux calculer des valeurs linguistiques. Toutefois, le calcul est dépendant du contexte.

Définition 16 : <u>Graphe modificateur</u>

Le graphe modificateur exprime, à travers des arcs, comment les valeurs linguistiques d'une ressource affectent les valeurs linguistiques d'une autre dans un contexte spécifique

Soit c *un contexte*
Soit G_c *un graphe sous le contexte* c
Soit R *un ensemble de ressources observables*
Soit E *un ensemble d'arcs exprimant l'interaction entre les ressources de* R

$$G_c : < R, E >$$
$$\forall\, e_c, \in E, e_c : < r_{1c}, r_{2c}, L_{r1}, L_{r2}, \mu, \rho_i >, r_{1c}, r_{2c} \in R,$$
$$L_{r1} = \{l_{0r1}, \dots, l_{ir1}, \dots, l_{nr1}\}, L_{r2} = \{l_{0r2}, \dots, l_{ir2}, \dots, l_{nr2}\},$$
$$\mu = \{\mu_{l_{0r1}-l_{0r2}}, \dots, \mu_{l_{0r1}-l_{ir2}}, \dots, \mu_{l_{0r1}-l_{nir2}}, \dots, \mu_{l_{ir1}-l_{0r2}}, \dots, \mu_{l_{ir1}-l_{ir2}}, \dots, \mu_{l_{ir1}-l_{nir2}},$$
$$\mu_{l_{nr1}-l_{0r2}}, \dots, \mu_{l_{nr1}-l_{ir2}}, \dots, \mu_{l_{nr1}-l_{nir2}}\}$$

Grâce à la Définition 7, nous savons que chaque ressource observable spécifie un ensemble de valeurs linguistiques (L), un ensemble de fonctions d'appartenance pour les calculer (μ), le temps de validité (φ) et le temps d'expiration (ψ) pour toute observation de la ressource observable sous un contexte spécifique c.

Chaque arc spécifie l'interaction entre deux ressources r_1 et r_2 et les valeurs linguistiques correspondantes L_{r1} et L_{r2} dans le contexte c, ainsi que l'ensemble des fonctions d'appartenance μ. La Figure 25 montre un exemple de fonctions d'appartenance associées aux ensembles flous représentant les valeurs linguistiques de la température modifiées par celles du sexe. Enfin, un poids ρ_i est attribué à chaque arc afin d'exprimer l'importance de la modification des valeurs linguistiques dans un contexte spécifique. Nous y reviendrons dans le chapitre 4.5.

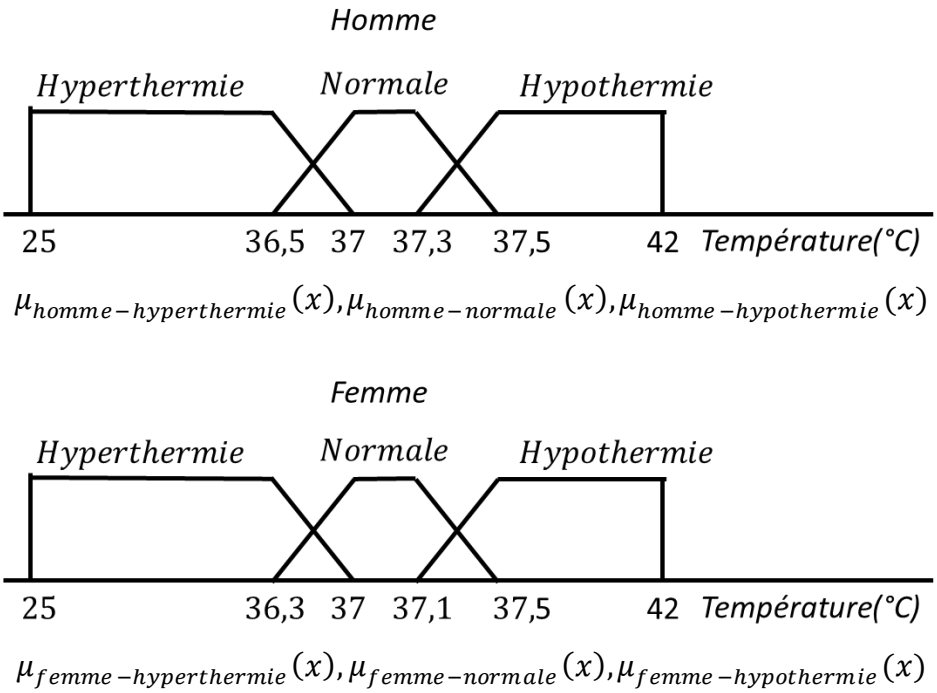


Figure 25. Fonctions d'appartenance, ensembles flous et valeurs linguistiques lorsque la température est modifiée par le sexe.

4.4.5 Synthèse de la section

Dans cette section nous avons introduit la préparation de connaissances qui concernent la transformation des valeurs quantitatives en valeurs linguistiques. La transformation prend en compte le contexte qui spécifie grâce au graphe modificateur comment les valeurs linguistiques d'une variable affectent les valeurs linguistiques d'une autre. Finalement, nous avons défini la situation d'alerte comme l'addition des conditions d'activation qui se divisent en deux types : conditions intrinsèques et conditions temporelles. Une condition du premier type est formée par une ressource, un opérateur, et une valeur linguistique de référence. Une condition temporelle est formée par deux ressources et l'opérateur temporel *commence avant ou en même temps que*.

Nous pouvons noter qu'après la définition par l'utilisateur d'une situation d'alerte à partir des valeurs linguistiques, celle-ci reste « inchangeable » puisque les conditions expriment sa préoccupation. Si l'évaluation de la situation (voir section 4.5) retourne des résultats incorrects, ce n'est pas la situation d'alerte qui doit être adaptée, mais le calcul des valeurs linguistiques sous un contexte spécifique. Les valeurs linguistiques permettent en fait de prendre en compte tous les éléments du contexte qu'ils soient qualitatifs ou quantitatifs.

4.5 Indices de qualité

Les indices de qualité sont des valeurs qui expriment la qualité des alertes détectées. Ils sont calculés lors de l'évaluation d'une situation d'alerte (Figure 26). Au

total, neuf indices sont calculés de manière ascendante, mais seulement deux entre eux sont propres aux alertes détectées. Le calcul des indices se fait en trois étapes (Tableau 15) : au niveau de l'utilisation des données, au niveau de la vérification des conditions et au niveau de la détection d'alertes.

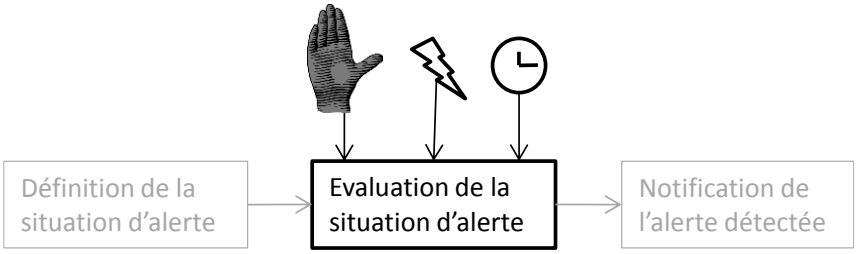


Figure 26. Evaluation d'une situation d'alerte

Tableau 15. Indices de qualité

Niveau	Etape	Indice
3	Détection d'alertes	Indice de Confiance d'une Alerte (ICA) Indice d'Applicabilité d'une Alerte (IAA)
2	Vérification des conditions	Indice de Confiance d'une Condition (ICC) Indice de Satisfaction d'une Condition Intrinsèque (ISCI) Indice de Satisfaction d'une Condition Temporelle (ISCT)
1	Utilisation des données	Indice de Confiance de l'Observation (ICO) Indice de Confiance de la Tendance (ICT) Indice Linguistique (IL) Indice Linguistique Révisé (ILR)

4.5.1 Niveau 1 : utilisation des données

Le niveau le plus bas sur lequel le système calcule des indices de qualité est l'utilisation des données. Il n'y a pas de calcul de qualité au niveau de la capture des données puisque TEMPAS considère que les données utilisées lors de l'évaluation d'une situation d'alerte sont fiables. Quatre indices sont calculés lors de l'utilisation des données. *L'indice de confiance de l'observation (ICO)*, *l'indice de confiance de la tendance (ICT), l'indice linguistique (IL), et l'indice linguistique révisé (ILR)*.

4.5.1.1 Indice de confiance de l'observation (ICO)

L'indice de confiance de l'observation concerne la « fraîcheur » d'une observation d'une ressource par rapport à un instant $\mathcal{T}$ quand la donnée va être exploitée.

Définition 17 : Indice de confiance de l'observation (ICO)

L'indice de confiance de l'observation (ICO) exprime la confiance que le système peut avoir dans une observation d'une ressource r sous un contexte c.

Exemple

Le système peut avoir confiance dans une observation de température datant de moins de 15 minutes

Le système ne peut pas avoir confiance dans une observation de température datant de 1 mois

L'indice de confiance de l'observation se calcule à l'aide du temps de validité φ et d'expiration ψ de la ressource, de l'horodatage de l'observation t et de l'instant d'évaluation $\mathcal{T}$. Dans la formule (4), la fonction ico calcule l'indice de confiance de l'observation. La fonction ico est représentée dans la Figure 27 lorsque t, φ et ψ sont connus. La fonction retourne une valeur entre zéro et un. Un ICO égal à un exprime que la donnée est fraîche, récente par rapport à l'instant $\mathcal{T}$. Inversement, un ICO égal à zéro exprime que la donnée est ancienne. Plus la donnée est récente, plus la confiance dans la donnée est élevée.

Soit i une observation d'une ressource r_c sur une entité e sous un contexte c.
Soit $\mathcal{T}$ l'instant d'évaluation.

$i: < e, r_c, t, v >$
$r_c : < L, \mu, \varphi, \psi >$

Calcul de l'indice de confiance de l'observation à l'instant $\mathcal{T}$: **(4)**

$$ico(\varphi, \psi, t, \mathcal{T}) = \begin{cases} 0 & si \quad \psi \leq (\mathcal{T} - t) \\ \dfrac{\psi - (\mathcal{T} - t)}{\psi - \varphi} & si \quad \varphi < (\mathcal{T} - t) < \psi \\ 1 & si \quad (\mathcal{T} - t) \leq \varphi \end{cases}$$

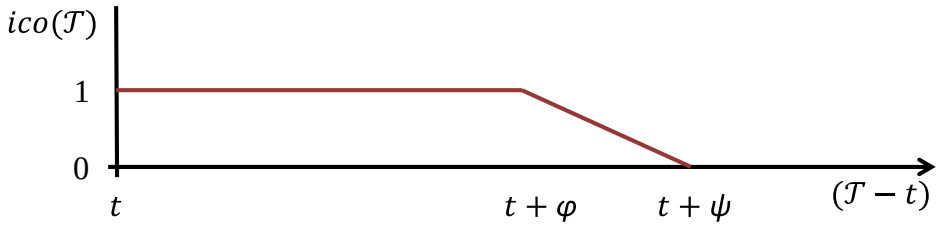


Figure 27. Indice de confiance de l'observation

4.5.1.2 Indice de confiance de la tendance (ICT)

La détection de tendances est faite souvent sur des séries chronologiques régulières formées par des données qui arrivent avec des fréquences connues. Ainsi, la distance temporelle entre deux observations consécutives est la même. Cependant, ce n'est pas toujours le cas. Un capteur, très fiable, peut envoyer des observations toutes les 10 minutes, s'arrêter et reprendre l'échantillonnage 26 minutes après. Une infirmière peu enregistrer les constantes médicales d'un patient toutes les 15 minutes, mais une indisponibilité peut changer la périodicité d'enregistrement. Ces observations sont exploitables lors de la détection de tendances, toutefois, il est

nécessaire de capturer cette irrégularité. Nous proposons de définir *un indice de confiance de la tendance* (ICT).

Définition 18 : <u>Indice de confiance de la tendance (ICT)</u>

L'indice de confiance de la tendance (ICT) reflète la confiance que le système peut avoir dans le segment associé à la tendance d'une ressource r par rapport à la distance temporelle entre les observations approximées par le segment dans un contexte c.

Exemple

Le système peut avoir confiance dans une tendance « à la hausse » formée par des températures observées tous les 10 minutes

Le système ne peut pas avoir confiance dans une tendance « à la hausse » formée par des températures observées tous les mois

Les tendances reflètent le comportement d'un ensemble de points. Une tendance est associée à un segment qui approxime et représente un ensemble d'observations. Le segment dépend de l'algorithme utilisé pour sa détection. La Figure 28 montre trois exemples de segments issus des algorithmes différents pour représenter le même ensemble de points. *L'indice de confiance de la tendance* (ICT) exprime la confiance que le système a dans le segment qui approxime un ensemble d'observations. Il se calcule avec la fonction ict présentée dans la formule (5). L'indice de confiance de la tendance (ICT) est le produit de la confiance à chaque observation par rapport à l'horodatage de l'observation suivante. Ainsi, l'ICT est le même pour chacun des segments dans la Figure 28.

Soit e une entité, soit r_c une ressource avec un temps de validité φ et un temps d'expiration ψ sous un contexte c.

$r_c : < L, \mu, \varphi, \psi >$

Soit $f(seg)$ la fonction qui retourne les horodatages des observations approximées par le segment seg

$$f(seg) = \{t_j | seg: < i_0, i_n >, i_0: < e, r, t_0, v_0 >, i_n: < e, r, t_n, v_n >, i_j: < e, r, t_j, v_j >, t_0 \leq t_j \leq t_n\} \quad (5)$$

Calcul de l'indice de confiance de la tendance :

$$ict(seg) = \prod_{j=0}^{n-1} ico(\varphi, \psi, t_j, t_{j+1}) ; t_j, t_{j+1} \in f(seg), t_j \neq t_{j+1},$$
$$\nexists t \in f(seg) | t_j < t < t_{j+1}, t_j \neq t \neq t_{j+1}$$

La Figure 29 montre plus en détail un exemple du calcul de l'indice de confiance d'une tendance représentée par un segment linéaire. D'abord, le système calcule l'indice de confiance des observations consécutives à l'aide la fonction ico et des

horodatages des observations. Ensuite l'ICT se calcule comme le produit des indices de confiance des observations consécutives.

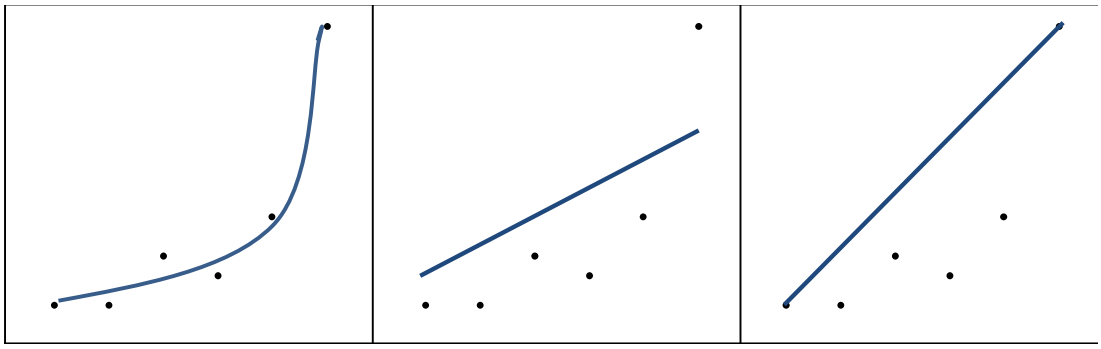
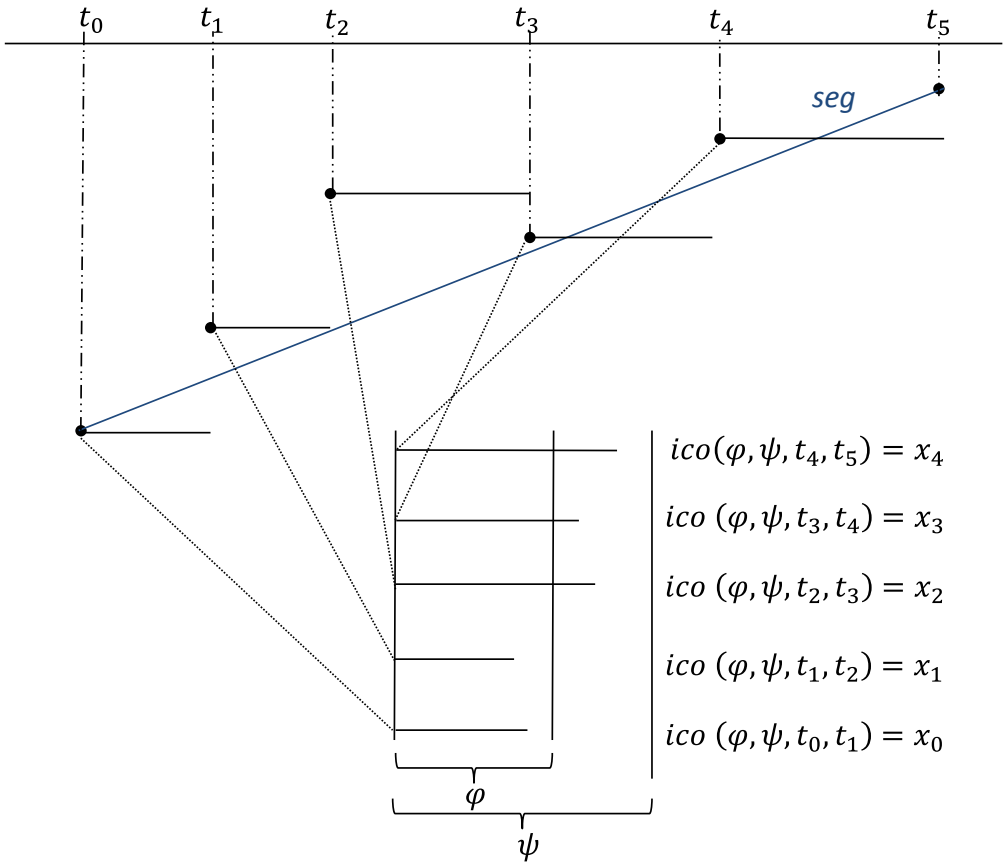


Figure 28. Indice de confiance de la tendance



$$ict(seg) = x_0 * x_1 * x_2 * x_3 * x_4$$

Figure 29. Détail du calcul de l'indice de confiance de la tendance

4.5.1.3 L'indice Linguistique (IL)

Définition 19 : <u>Indice Linguistique (IL)</u>

L'indice linguistique (IL) exprime le degré de certitude de l'utilisation de la valeur linguistique l pour représenter la valeur numérique v sous le contexte c.

Exemple

Il est certain que la valeur linguistique « adulte » représente une personne de 30 ans

Il est incertain que la valeur linguistique « adulte » représente une personne de 10 ans

L'indice Linguistique (IL) a déjà été présenté partiellement dans le Tableau 13 et le Tableau 14 de la section 4.4.2. Il se calcule avec la fonction il de la formule (6). Cet indice correspond au degré d'appartenance de la valeur v à l'ensemble flou μ_l représentant la valeur linguistique l dans le contexte c. L'IL est une valeur entre zéro et un. Plus l'IL est élevé, plus la valeur linguistique l représente la valeur numérique v.

Soit r_c une ressource dans un contexte c, soit L l'ensemble de valeurs linguistiques exprimant l'état ou la tendance de r_c, soit μ l'ensemble de fonctions d'appartenance aux valeurs linguistiques dans L

$$r_c : < L, \mu, \varphi, \psi >, L = \{l_0, \dots, l_i, \dots, l_n\}, \mu = \{\mu_{l_0}, \dots, \mu_{l_i}, \dots, \mu_{l_n}\} \quad \textbf{(6)}$$

Calcul de l'indice linguistique :

$$il(v, l, \mu) = \mu_l(v)$$

4.5.1.4 Indice linguistique révisé (ILR)

La valeur linguistique l peut représenter plus ou moins correctement une valeur numérique v relativement à la connaissance d'autres valeurs v_i.

Définition 20 : <u>Indice Linguistique Révisé (ILR)</u>

L'indice linguistique révisé (ILR) exprime le degré de certitude d'utilisation de la valeur linguistique l_0 pour représenter la valeur numérique v_0 sous le contexte c étant donné l'existence d'autres valeurs linguistiques $l_1, \dots, l_{n+1}$ représentant des valeurs numériques $v_1, \dots, v_{n+1}$ des ressources $r_1, \dots, r_{n+1}$

Exemple

Il est certain qu'une personne d' 1m50, de « petite » taille et pesant 90 Kg est « obèse »

Il est incertain qu'une personne de 2m10, de « grande » taille, pesant 90 Kg est « obèse »

L'indice linguistique révisé se calcule à l'aide du graphe modificateur de la Définition 16 qui exprime comment les valeurs linguistiques d'une ressource modifient les valeurs linguistiques d'une autre dans un contexte c. Le calcul de l'indice linguistique révisé se trouve dans la formule (7). La fonction $\ltimes$ trouve tous les chemins du graphe G_c aboutissant à la ressource r_0 (dans le pire des cas, 2^{m-2} où m est le nombre de ressources dans le graphe). Il est nécessaire de connaitre tous les chemins du graphe qui aboutissent à la ressource r_0 pour laquelle a été définie la valeur linguistique l parce qu'une ressource r_2 peut modifier la ressource r_1 qui modifie la

ressource r_0. Ceci suppose d'ajouter des contraintes au graphe, lequel doit être orienté et absent de cycles. Un seul parmi tous les chemins est choisi à l'aide de la fonction $\prec$ qui retourne celui dont la somme des poids des arcs qui le composent est la plus grande. Ce choix s'appuie sur le fait que le poids exprime l'importance du lien.

Soit G_c le graphe modificateur qui exprime à travers des arcs comment les valeurs linguistiques d'une ressource affectent les valeurs linguistiques d'une autre dans un contexte c.

$G_c : < R, E >$

Soit $\ltimes (G_c, r_0)$ la fonction qui calcule tous les chemins dans le graphe G_c qui aboutissent à la ressource r_0

$$\begin{gathered}\ltimes (G_c, r_0) = \mathcal{C}_{r_0} = \{C: e_n - e_{n-1} - \cdots - e_1 - e_0 | \\ e_n: < r_{n+1}, r_n, L_{r_{n+1}}, L_{r_n}, \mu_n, \rho_n >, \\ e_{n-1}: < r_n, r_{n-1}, L_{r_n}, L_{r_{n-1}}, \mu_{n-1}, \rho_{n-1} >, \\ \ldots, e_i: < r_{i+1}, r_i, L_{r_{i+1}}, L_{r_i}, \mu_i, \rho_i >, \ldots, \\ e_1: < r_2, r_1, L_{r_2}, L_{r_1}, \mu_1, \rho_1 >, \\ e_0: < r_1, r_0, L_{r_1}, L_{r_0}, \mu_0, \rho_0 >\}\end{gathered}$$

Soit la fonction $i(C)$ qui calcule l'importance du chemin C

$$i(C) = \sum_{i=0}^{n} \rho_i ;$$

$$C: < r_{n+1}, r_n, L_{r_{n+1}}, L_{r_n}, \mu_n, \rho_n > - \cdots - < r_1, r_0, L_{r_1}, L_{r_0}, \mu_0, \rho_0 > \quad \textbf{(7)}$$

Soit la fonction $\prec (\mathcal{C}_{r_0})$ qui retourne le chemin C dont l'importance est la plus grande parmi tous les chemins de $\mathcal{C}_{r_0}$ qui aboutissent à la ressource r_0.

$$\prec (\mathcal{C}_{r_0}) = C, C \in \mathcal{C}_{r_0}, \nexists C_i \in \mathcal{C}_{r_0}, i(C_i) > i(C)$$

Soit la fonction $v(C)$ qui retourne le vecteur V avec les valeurs observées des ressources observables dans le chemin C

$$v(C) = V = [v_{n+1}, \ldots, v_1, v_0]$$

Soit la fonction $\mathcal{Y}(V, v_0, l_0, C)$ qui retourne la matrice M dont les colonnes contiennent les valeurs linguistiques qui représentent les valeurs du vecteur V et les degrés respectifs d'appartenance. Les lignes contiennent toutes les combinaisons possibles des valeurs linguistiques (et les degrés respectifs d'appartenance) dont l'acheminement abouti à la valeur linguistique l_0 qui représente la valeur numérique v_0

$$\mathcal{Y}(V, v_0, l_0, C) = M = \begin{bmatrix} l_{1,n+1}, d_{1,n+1} & \ldots & l_0, d_{1,0} \\ \vdots & \ddots & \vdots \\ l_{k,n+1}, d_{k,n+1} & \ldots & l_0, d_{k,0} \end{bmatrix}$$

Soit la fonction $\mathfrak{y}(M)$ qui calcule le produit de tous les degrés d'appartenance dans une même ligne

$$\mathfrak{y}(M) = [d_1; \dots; d_i; \dots; d_k], d_i = d_{i,n+1} * \dots * d_{i,0}$$

Calcul de l'indice linguistique révisé

(7)

$$ilr\left(v\left(\prec\left(\ltimes(G_c, r_0)\right)\right), v_0, l_0, \prec\left(\ltimes(G_c, r_0)\right)\right) = \sum\left(\mathfrak{y}\left(\mathcal{y}\left(v\left(\prec\left(\ltimes(G_c, r_0)\right)\right), v_0, l_0, \prec\left(\ltimes(G_c, r_0)\right)\right)\right)\right)$$

Une fois le chemin le plus important trouvé, la fonction $\mathcal{y}$ calcule les valeurs linguistiques représentant les valeurs numériques des ressources impliquées dans le chemin. Le calcul commence par la ressource qui n'est pas modifiée et se propage vers l'avant du chemin jusqu'à la ressource r_0. Le résultat de la fonction $\mathcal{y}$ est une matrice, dont la colonne i contient une valeur linguistique l_i et le degré d'appartenance calculé avec la fonction $\mu_{l_{i-1}-l_i}$ (l_{i-1} correspond à la valeur linguistique de la colonne antérieure) est évalué sur la valeur numérique v_i. Si la colonne i est la première colonne, elle contient la valeur linguistique l_i et le degré d'appartenance calculé avec la fonction μ_{l_i} lors de l'évaluation sur la valeur numérique v_i. La matrice peut contenir plusieurs lignes puisque une valeur numérique peut être ambigüe quand deux valeurs linguistiques génèrent une bifurcation. Les bifurcations peuvent avoir lieu tout au long du parcours du chemin. La fonction $\mathfrak{y}$ calcule le produit des degrés d'appartenance qui se trouvent dans la même ligne. Finalement la fonction ilr calcule *l'indice linguistique révisé*.

La corpulence d'une personne est habituellement mesurée avec son indice de masse corporelle (IMC). Afin d'illustrer un exemple du calcul de l'indice linguistique révisé (ILR), nous allons considérer une personne « obèse » en utilisant son « poids », sa « taille », et son « âge ». Pour cela, nous utilisons le graphe de la Figure 30 qui exprime comment les valeurs linguistiques de l'« âge » modifient celles de la « taille » et celles du « poids », et comment les valeurs linguistiques de la « taille » modifient celles du « poids ». L'entité concernée est une personne de 16 ans, mesurant 1m72 et pesant 76 Kg. Les calculs de l'ILR affectant la valeur linguistique « obèse » pour représenter l'entité concernée se trouvent dans le Tableau 16.

L'indice linguistique révisé calculé dans le Tableau 16 est trop bas (≈0.15, l'ILR est la somme des produits des chemins qui aboutissent dans la valeur linguistique « obèse ».). Il exprime qu'il n'est pas correct de considérer une personne de 16 ans, mesurant 1m72 et pesant 76 Kg comme une personne « obèse ».

Tableau 16. Exemple du calcul de l'indice linguistique révisé

Age	Taille	Poids
16 ans	1m72	76 Kg

Age	Valeur Ling Age	Fonction d'appartenance	C_{Age}
16 ans	Jeune	$\mu_{jeune}(16)$	0.5
16 ans	Adulte	$\mu_{adulte}(16)$	0.5

Valeur Ling Age	Taille	Valeur Ling Taille	Fonction d'appartenance	C_{Age}	C_{Taille}
Jeune	1m72	Normale	$\mu_{jeune-normale}(172)$	0.5	0.2
Jeune	1m72	Grande	$\mu_{jeune-grande}(172)$	0.5	0.8
Adulte	1m72	Normale	$\mu_{adulte-normale}(172)$	0.5	0.75
Adulte	1m72	Grande	$\mu_{adulte-grande}(172)$	0.5	0.25

Valeur Ling. Age	Valeur Ling. Taille	Poids	Valeur Ling Poids	Fonction d'appartenance	C_{Age}	C_{Taille}	C_{Poids}
Jeune	Normale	76 Kg	Normale	$\mu_{normale-normale}(76)$	0.5	0.2	0.67
Jeune	Normale	76 Kg	Obèse	$\mu_{normale-obèse}(76)$	0.5	0.2	0.33
Jeune	Grande	76 Kg	Normale	$\mu_{grande-normale}(76)$	0.5	0.8	1
Adulte	Normale	76 Kg	Normale	$\mu_{normale-normale}(76)$	0.5	0.75	0.67
Adulte	Normale	76 Kg	Obèse	$\mu_{normale-obèse}(76)$	0.5	0.75	0.33
Adulte	Grande	76 Kg	Normale	$\mu_{grande-normale}(76)$	0.5	0.25	1

Valeurs Lings. du chemin		M			Produit	ILR
		C_{Age}	C_{Taille}	C_{Poids}		
Jeune-Normale-Obèse	Σ	0.5	0.2	0.33	0.033	0.15675
Adulte-Normale-Obèse		0.5	0.75	0.33	0.12375	

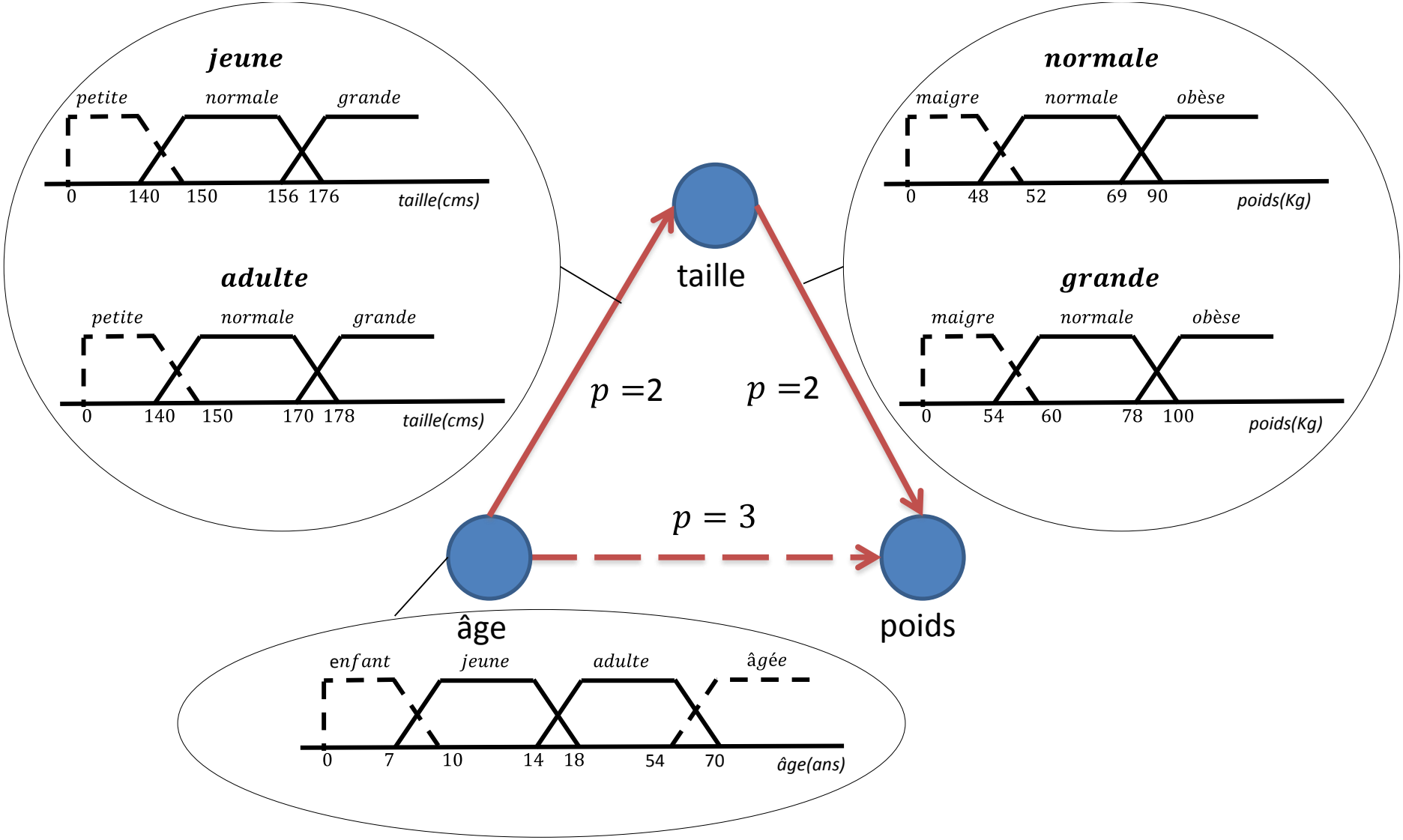


Figure 30. Graphe modificateur

4.5.2 Niveau 2 : vérification des conditions d'activation

La vérification des conditions d'activation calcule les indices du deuxième niveau. Les indices de ce niveau remontent la valeur des indices du niveau précédent (ICO, ICT, IL, ILR) lorsque la condition est satisfaite (Tableau 15). Trois indices sont calculés dans ce niveau. L'*indice de satisfaction de la condition intrinsèque (ISCI)* exprime si la, ou les observations utilisées pour vérifier la condition d'activation, la satisfont. L'*indice de satisfaction d'une condition temporelle (ISCT)* exprime si l'ordre temporel des horodatages est respecté. L'*indice de confiance de la condition (ICC)* exprime la confiance faite aux observations utilisées pour la vérifier.

4.5.2.1 Indice de satisfaction d'une condition intrinsèque (ISCI)

Le calcul de l'indice de satisfaction d'une condition intrinsèque se trouve dans la formule (8). C'est une valeur entre 0 et 1 qui exprime le degré de satisfaction de la condition. Une ISCI de 0.7 exprime que la condition est plutôt satisfaite, mais pas complètement satisfaite. Le calcul de l'ISCI varie selon le type d'opérateur (≤, =, et ≥) utilisé dans la condition.

Définition 21 : Indice de satisfaction d'une condition intrinsèque (ISCI)

L'indice de satisfaction d'une condition intrinsèque (ISCI) exprime le degré avec lequel une valeur numérique satisfait une condition intrinsèque formée par une valeur linguistique

Exemple

Une température de 40° C satisfait la condition « si la température est supérieure ou égale à normale »

Une tension de 190 mm Hg ne satisfait pas la condition « si la tension systolique est une hypotension »

Supposons que l'indice linguistique, dans sa forme simple (IL) ou révisé (ILR), est supérieur à zéro pour une valeur linguistique l représentant une valeur numérique val de l'observation ou du segment et qu'il vérifie la condition intrinsèque. Dans ce cas, la valeur linguistique l est comparée avec la valeur linguistique de référence l_{ref} utilisée pour définir la condition intrinsèque. Si les deux valeurs linguistiques sont égales, alors l'indice de satisfaction de la condition est égale à l'IL ou à l'ILR, respectivement. Puisque les valeurs linguistiques sont ordonnées (Définition 8), l'ISCI est égal à 1 si la valeur linguistique l est supérieure à la valeur l_{ref} et l'opérateur utilisé dans la condition égal à « $\geq$ » ou si la valeur linguistique l est inférieure à la valeur l_{ref} et l'opérateur utilisé dans la condition égale à « $\leq$ ». Dans tout autre cas, l'ISCI est égal à zéro.

Soit $cond$ *une condition d'activation intrinsèque formée par une ressource* r_0*, un opérateur relationnel* op*, et une valeur linguistique de référence* l_{ref}*, dont l'ensemble* L *contient les valeurs linguistiques pour représenter l'état ou la tendance de la ressource* r_0*.* **(8)**

Soit val *la valeur observée, ou la propriété du segment approximant les observations de la ressource* r_0 *sur l'entité* en, *utilisée pour vérifier la condition* $cond$.
Soit G *le graphe modificateur qui exprime à travers des arcs comment les valeurs linguistiques d'une ressource affectent les valeurs linguistiques d'une autre ressource.*

$r_0 : < L, \mu, \varphi, \psi >, L = \{l_0, \dots, l_i, \dots, l_n\}$
$cond : < r_0, op, l_{ref} >, op \in \{\leq, =, \geq\}, l_{ref} \in L$
$G : < R, E >$

Calcul de l'indice de satisfaction de la condition intrinsèque :

$$isci(cond, val, l) =$$

$$\begin{cases} \mathrm{il}(\mathrm{val}, \mathrm{l}, \mu) & \text{si } \left(\nexists\, e \in E, e : < r, r_0, L_r, L_{r_0}, \mu, \rho_i >\right) \text{ et } (\mathrm{il}(val, \mathrm{l}, \mu) > 0) \text{ et } (\mathrm{l} = \mathrm{l}_{ref}) \\ 1 & \text{si } \left(\nexists\, e \in E, e : < r, r_0, L_r, L_{r_0}, \mu, \rho_i >\right) \text{ et } (\mathrm{il}(val, \mathrm{l}, \mu) > 0) \text{ et } \left((\mathrm{l} > \mathrm{l}_{ref}) \text{ et op} = " \geq ") \text{ ou } ((\mathrm{l} < \mathrm{l}_{ref}) \text{ et op} = " \leq ")\right) \\ \mathrm{ilr}\left(\mathrm{v}\left(\measuredangle\left(\ltimes(\mathrm{G_c}, \mathrm{r_0})\right)\right), \mathrm{val}, \mathrm{l}, \measuredangle\left(\ltimes(\mathrm{G_c}, \mathrm{r_0})\right)\right) & \text{si } \left(\exists\, e \in E, e : < r, r_0, L_r, L_{r_0}, \mu, \rho_i >\right) \text{ et } \left(\mathrm{ilr}\left(\mathrm{v}\left(\measuredangle\left(\ltimes(\mathrm{G}, \mathrm{r_0})\right)\right), \mathrm{val}, \mathrm{l}, \measuredangle\left(\ltimes(\mathrm{G}, \mathrm{r_0})\right)\right) > 0\right) \text{ et } (\mathrm{l} = \mathrm{l}_{ref}) \\ 1 & \text{si } \left(\exists\, e \in E, e : < r, r_0, L_r, L_{r_0}, \mu, \rho_i >\right) \text{ et } \left(\mathrm{ilr}\left(\mathrm{v}\left(\measuredangle\left(\ltimes(\mathrm{G}, \mathrm{r_0})\right)\right), \mathrm{val}, \mathrm{l}, \measuredangle\left(\ltimes(\mathrm{G}, \mathrm{r_0})\right)\right) > 0\right) \left((\mathrm{l} > \mathrm{l}_{ref}) \text{ et op} = " \geq ") \text{ ou } ((\mathrm{l} < \mathrm{l}_{ref}) \text{ et op} = " \leq ")\right) \\ 0 & \text{dans d'autres cas} \end{cases} \quad (8)$$

4.5.2.2 Indice de satisfaction d'une condition temporelle (ISCT)

Définition 22 : <u>Indice de satisfaction d'une condition temporelle (ISCT)</u>

L'indice de satisfaction d'une condition temporelle exprime si les horodatages des observations ou des segments respectent la contrainte temporelle « commence avant ou en même temps que »

Exemple

Une observation de la température prise le 20/01/2014 à 17h20 et un segment approximant la tension entre 17h05 et 17h30 du 20/01/2014 satisfait la condition « la température commence avant ou en même temps que la tendance de la tension »

Le degré de satisfaction d'une condition temporelle est 0 ou 1. Soit la condition est satisfaite, soit elle ne l'est pas. Les conditions d'activation intrinsèques expriment par défaut une concurrence temporelle. Toutefois, les conditions temporelles servent à ajouter un ordre partiel, si nécessaire, parmi les conditions intrinsèques. Ainsi, les conditions temporelles sont associées à des conditions intrinsèques. Le calcul de l'ISCT se trouve dans la formule (9).

Une condition temporelle entre deux ressources ($r_1 \leq r_0$) peut être satisfaite seulement si les conditions intrinsèques associées (définies à partir des ressources r_1 et r_0) sont satisfaites. Dans le cas échéant, la condition temporelle est satisfaite si l'horodatage de l'observation ou du segment de la ressource r_1 *commence avant ou en même temps* que l'horodatage de l'observation ou du segment de la ressource r_0. L'horodatage d'une observation est un instant tandis que l'horodatage d'un segment est un intervalle temporel. La relation « commence avant ou en même temps que » est illustrée dans la Figure 21, section 4.4.3.

Soient $cond_0$ et $cond_1$ deux conditions d'activations intrinsèques formées à partir des ressources r_0 et r_1 respectivement
Soit val_0 la valeur observée, ou la propriété du segment approximant les observations de la ressource r_0, utilisée pour vérifier la condition $cond_0$
Soit t_0 l'horodatage de la valeur observée, ou du segment approximant les observations de la ressource r_0, utilisé pour vérifier la condition $cond_0$
Soit val_1 la valeur observée, ou la propriété du segment approximant les observations de la ressource r_1, utilisée pour vérifier la condition $cond_1$
Soit 1 l'horodatage de la valeur observée, ou du segment approximant les observations de la ressource r_2, utilisé pour vérifier la condition $cond_1$

$$\begin{aligned}
&r_0 : < L_0, \mu_0, \varphi_0, \psi_0 > \\
&cond_0 : < r_0, op_0, l_{ref0} >, op_0 \in \{\leq, =, \geq\}, l_{ref0} \in L_0 \\
&r_1 : < L_1, \mu_1, \varphi_1, \psi_1 > \\
&cond_1 : < r_1, op_1, l_{ref1} >, op_1 \in \{\leq, =, \geq\}, l_{ref1} \in L_1
\end{aligned} \tag{9}$$

Soit $cond$ une condition d'activation temporelle formée par les ressources r_0 et r_1

$$cond : < r_0 \leq r_1 >$$

Calcul de l'indice de satisfaction de la condition temporelle :

$$isct(cond) =$$

$$\begin{cases} 1 \quad si \quad \begin{array}{c} (\exists\, l_0 \in L_0, isc(cond_0, l_0,) > 0) \\ et\ (\exists\, l_1 \in L_1, isc(cond_1, l_1,) > 0) \\ et\ (t_0 \leq t_1) \end{array} \\ \\ 0 \qquad dans\ d'autres\ cas \end{cases}$$

4.5.2.3 Indice de confiance de la condition (ICC)

L'indice de confiance d'une condition se calcule seulement sur les conditions intrinsèques. Le calcul de l'ICC est présenté dans la formule (10).

Définition 23 : Indice de confiance de la condition (ICC)

L'indice de confiance de la condition (ICC) exprime le degré de confiance que le système a dans les observations ou les segments utilisés pour vérifier la condition intrinsèque.

Soient $cond$ *une condition d'activation intrinsèque formée à partir de la ressource* r
Soit i *ou* seg *l'observation ou le segment approximant les observations de la ressource* r *utilisée pour valider la condition* $cond$
Soit $\mathcal{T}$ *l'instant de vérification de la condition*

$r : < L, \mu >, L = \{l_0, \dots, l_i, \dots, l_n\}$
$i: < e, r, t, v >$
$seg: < i_1, i_2, v >, i_1: < e, r, t_1, v_1 >, i_2: < e, r, t_2, v_2 >, t_1 < t_2$
$cond: < r_0, op, l_{ref} >, op \in \{\leq, =, \geq\}, l_{ref} \in L$

Calcul de l'indice de confiance de la condition : **(10)**

$$icc(cond) = \begin{cases} ico(\varphi, \psi, t, \mathcal{T}) & si \quad \begin{array}{c}(\exists\, l \in L, isci(cond, l,) > 0) \\ et\ (l\ exprime\ un\ possible\ état\ de\ r)\end{array} \\ ict(seg) & si \quad \begin{array}{c}(\exists\, l \in L, isci(cond, l,) > 0) \\ et\ (l\ exprime\ une\ possible\ tendance\ de\ r)\end{array} \\ 0 & dans\ d'autres\ cas \end{cases}$$

L'ICC exprime la confiance faite dans l'observation ou le segment utilisé pour vérifier la condition. Si la condition intrinsèque est satisfaite (ISCI > 0), l'indice de confiance de la condition correspond à l'indice de confiance de l'observation si la valeur linguistique de la condition intrinsèque exprime l'état de la ressource, ou l'indice de confiance de la tendance si la valeur linguistique de la condition intrinsèque exprime la tendance de la ressource.

4.5.3 Niveau 3 : détection de l'alerte

Les indices de qualité lors de la détection d'une alerte se calculent à partir des indices de qualité calculés lors de la vérification des conditions d'activation (Tableau 15). Les situations d'alerte sont définies comme l'addition des conditions d'activation. Plus le nombre de conditions vérifiées augmente, plus l'entité est concernée. Une situation d'alerte peut se produire même si une condition n'est pas satisfaite. Dans

l'exemple de la canicule de la section 4.4.3, la personne peut ne pas être âgée, mais être pourtant concernée par l'alerte. Utiliser la conjonction logique mènerait à conclure que la situation d'alerte ne se produit pas si la personne n'est pas âgée. Cependant, cette condition peut ne pas être la plus importante.

Deux indices de qualité concernent les alertes détectées, l'*indice d'applicabilité d'une alerte (IAA)* et l'*indice de confiance d'une alerte (ICA*).

4.5.3.1 Indice d'applicabilité d'une alerte (IAA)

Définition 24 : Indice d'applicabilité d'une alerte (IAA)

L'indice d'applicabilité d'une alerte (IAA) exprime dans quelle mesure une entité (qui est observée) est concernée par une situation d'alerte

Exemple

Un IAA de 0.7 après l'évaluation de la situation d'alerte de canicule sur M. DUPONT (l'entité) exprime qu'il est plutôt concerné par l'alerte

Un IAA de 0.2 après l'évaluation de la situation d'alerte de canicule sur M. MARTIN (l'entité) exprime qu'il est peu concerné par l'alerte

La formule (11) montre le calcul de l'indice d'applicabilité d'une alerte. L'IAA est une valeur entre zéro et un qui se calcule à partir de l'indice de satisfaction des conditions qui composent la situation d'alerte et leur importance respective. La somme des poids des conditions est égale à 1 pour normaliser la valeur de l'IAA. Plus les conditions sont satisfaites, plus l'entité est concernée.

Soit a une situation d'alerte exprimée comme l'addition de k conditions d'activation, où un poids est associé à chaque condition exprimant son importance vis-à-vis de la situation d'alerte

$$a = +^{k}\ (c, \rho)$$

$$\sum_{i=0}^{k} \rho_i = 1$$

Soit la fonction $(isc(c))$ qui retourne l'indice de satisfaction de la condition c

$$isc(c) = \begin{cases} isci(c) & si \quad c\ est\ une\ condition\ intrinsèque \\ isct(c) & si \quad c\ est\ une\ condition\ temporelle \end{cases} \tag{11}$$

Calcul de l'indice d'applicabilité de l'alerte (IAA) :

$$iaa(a) = \sum_{i=0}^{k} (isc(c_i)) * \rho_i$$

4.5.3.1.1 Entités concernées par une situation d'alerte

En général, les situations d'alerte peuvent s'évaluer sur toutes les entités et détecter plus d'alertes que celles intéressant les utilisateurs. Les conditions d'activation se comportent en quelque sorte comme des filtres d'entités où celles qui vérifient la condition ont plus de chances d'être détectées.

Supposons une situation d'alerte définie par trois conditions d'activation c_1, c_2 et c_3 avec des poids de 0.6, 0.3, et 0.1 respectivement. La Figure 31 montre une vue d'ensembles C_1, C_2 et C_3 représentant la satisfaction des conditions d'activation correspondantes et leur poids lors du calcul de l'indice d'applicabilité de l'alerte (IAA). A chaque région est associée une valeur correspondant à la valeur maximale de l'IAA d'une alerte détectée sur une entité. Une entité appartient à une région si l'observation (ou le segment) de la ressource sur l'entité vérifie la condition avec un Indice de satisfaction de la condition, intrinsèque ou temporelle, supérieur à zéro.

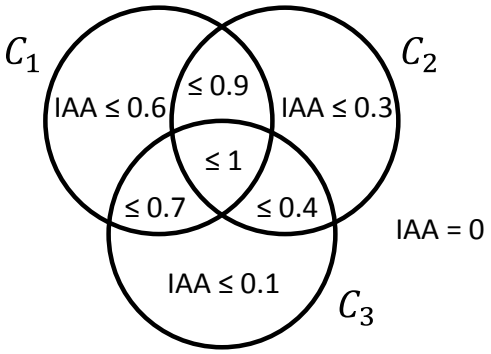


Figure 31. Mise en œuvre des conditions d'activation

Si seules les alertes avec une IAA supérieure à 0.5 sont notifiées (voir section 4.6.1.1), il est nécessaire que les entités vérifient, au moins partiellement, la condition c_1. Notons que la condition c_1 est *indispensable* pour deux raisons : son poids dans la situation de l'alerte, et la qualité minimale requise des alertes détectées instanciant la situation d'alerte. Ainsi, une situation d'alerte peut avoir au maximum une condition indispensable. Les conditions indispensables permettent donc de grouper les entités. Si des symptômes chez les hommes doivent attirer l'attention des utilisateurs mais pas chez les femmes, la condition peut se définir $c_1 :< sexe, =, homme >$. Grouper par condition est obtenu par l'expressivité des situations d'alerte dans TEMPAS.

4.5.3.2 Indice de confiance d'une alerte (ICA)

Définition 25 : <u>Indice de confiance d'une alerte (ICA)</u>

L'indice de confiance d'une alerte exprime la confiance dans les observations et les segments utilisés lors de l'évaluation d'une situation d'alerte

Exemple

Un ICA de 1 après l'évaluation de la situation d'alerte de canicule sur M. DUPONT (l'entité) exprime que les observations et segments utilisés sont récents et fiables

Un ICA de 0.2 après l'évaluation de la situation d'alerte de canicule sur M. MARTIN (l'entité) exprime que les observations ou segments utilisés ne sont pas tous récents et ne sont pas tout à fait fiables

La formule (12) montre le calcul de l'Indice de confiance de l'alerte qui est une valeur entre zéro et un calculée à partir des indices de confiance des conditions d'activation intrinsèques qui composent la situation d'alerte et leur importance respective.

Notons que l'ICA est dépendant de l'ICC qui est dépendant de l'ISC. Par conséquent l'ICA peut seulement être égal à un si toutes les conditions sont satisfaites au moins partiellement. Une condition intrinsèque qui n'est pas satisfaite réduit l'IAA et l'ICA.

Soit a une situation d'alerte exprimée comme l'addition de k conditions d'activation, où un poids est associé à chaque condition exprimant son importance vis-à-vis de la situation d'alerte

$$a = +^{k}\ (c,\rho)$$

$$\sum_{i=0}^{k} \rho_i = 1$$

Soit P la somme des importances des conditions intrinsèques **(12)**

$$P = \sum_{i=0}^{k} \rho_i : c_i \ est\ une\ condition\ intrinsèque$$

Calcul de l'indice de confiance de l'alerte (ICA) :

$$ica(a) = \frac{1}{P}\sum_{i=0}^{k}(icc(c_i)) * \rho_i$$

4.5.4 Instant d'évaluation

Les indices de qualité qui expriment la fiabilité des observations à partir de leur fraîcheur (ICA, l'ICC, l'ICO, et l'ICT) sont dépendants de l'instant de l'évaluation de la situation d'alerte. Supposons une situation d'alerte définie avec une seule *condition d'activation intrinsèque* qui évalue si la « diurèse » se trouve dans *l'état* de « polyurie » (plus de 2.5 litres par jour). La Figure 32 illustre trois instants d'évaluation différents, τ_1, τ_2, et τ_3 pour lesquels le système utilise la même observation pour évaluer la situation d'alerte. L'IAA est le même dans les trois instants puisque la valeur observée est la même. En revanche, l'ICA se réduit au fur et à mesure que l'instant d'évaluation s'éloigne de l'observation.

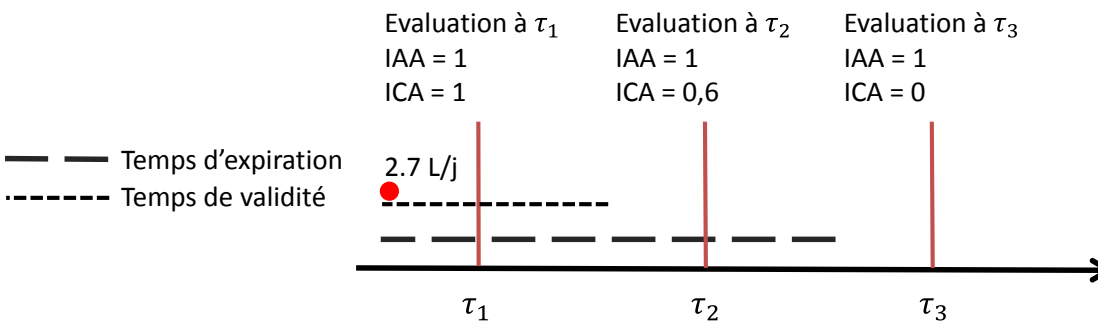


Figure 32. Indice de confiance de l'alerte à trois instants différents

4.5.5 Synthèse de la section

Cette section a introduit neuf indices de qualité calculés sous forme ascendante, depuis l'utilisation des données (observations) jusqu'à la détection de l'alerte (Tableau 15). Une alerte détectée associe deux indices de qualité, l'indice d'applicabilité et l'indice de confiance. L'indice d'applicabilité exprime dans quelle mesure une entité est concernée par une situation d'alerte. Il se calcule à partir des degrés de satisfaction des conditions d'activation par les observations de l'entité utilisées lors de l'évaluation de la situation d'alerte. L'indice de confiance exprime la confiance qu'il est possible d'avoir dans les observations et les segments utilisés lors de l'évaluation de la situation d'alerte. Cette confiance s'appuie sur la fraîcheur des observations par rapport à l'instant d'évaluation.

Si une situation d'alerte est définie à partir de plusieurs conditions d'activation, quelques-unes entre elles peuvent ne pas être satisfaites lors de l'évaluation de la situation d'alerte sur une entité. Toutefois, l'entité peut quand même être très concernée par la situation d'alerte grâce à l'importance des conditions satisfaites. L'indice de confiance de l'alerte est dépendant de l'indice d'applicabilité puisque si une condition n'est pas satisfaite, la confiance est réduite. Le Tableau 17 montre l'interprétation des alertes détectées selon l'IAA et l'IC

Tableau 17. Interprétation des alertes détectées à partir des indices de qualité

IAA	ICA	Interprétation
1	1	Toutes les conditions d'activation sont satisfaites et les observations et segments sont fiables
1	< 1	Toutes les conditions d'activations sont satisfaites et les observations ou les segments ne sont pas complètement fiables
< 1	1	Toutes les conditions d'activation sont satisfaites, au moins partiellement, et les observations et segments sont fiables
< 1	< 1	Quelques conditions d'activation sont satisfaites, au moins partiellement. Il n'est pas possible de déterminer si les observations ou segments utilisés sont fiables

L'IAA et l'ICA utilisés sont exploitables par les utilisateurs, par exemple, pour réduire les faux-positifs, les faux-négatifs, prioriser les situations d'alerte, etc. TEMPAS est un système d'alertes adaptable par les utilisateurs, qui d'une manière ou d'une autre, participent à la « préparation de connaissances » (section 4.4). Ainsi, toute adaptation en temps réel (ajouter, modifier, ou effacer des conditions d'activation des situations d'alerte ; ajouter des valeurs linguistiques pour mieux définir les situations d'alerte ; modifier les ensembles flous (trapézoïdaux) et les fonctions d'appartenance correspondantes pour calculer le degré de certitude de l'utilisation d'une valeur linguistique pour représenter une valeur numérique ; prolonger ou raccourcir le temps de validité ou d'expiration ; etc.) influe sur les prochains calculs des valeurs des indices de qualité.

4.6 Politique de notification

La *politique de notification* est définie pour chaque situation d'alerte et permet de définir une stratégie pour gérer les alertes détectées. La finalité de ce mécanisme est d'obtenir une notification adaptée selon les utilisateurs et les contextes. Une sur-notification ou une absence de notification pourrait être préjudiciable.

TEMPAS applique deux types de contrôle de notification : un contrôle sur les alertes détectées lors de la même évaluation d'une situation d'alerte, et un autre en comparant une alerte accessible avec les précédentes. Les deux contrôles ont lieu après l'évaluation des situations d'alerte (Figure 33).

Définition 26 : Alerte accessible

Une alerte accessible est une alerte construite à partir d'une ou plusieurs alertes détectées.

Figure 33. Notification des alertes

4.6.1 Filtrage d'alertes détectées par une évaluation

Une situation d'alerte est définie à partir de plusieurs conditions d'activation, chacune peut être construite à l'aide des opérateurs comme « ≤ », « = », ou « ≥ » et d'une valeur linguistique l. Une condition construite avec un opérateur « ≤ » ou « ≥ » peut être satisfaite par plusieurs valeurs linguistiques. Une valeur numérique v peut

être représentée ambigument par deux valeurs linguistiques l_0 et l_1 avec des indices linguistiques (IL ou ILR) supérieures à zéro et inférieures à un.

Reprenons le graphe modificateur de la Figure 30 et le Tableau 16 de la section 4.5.1.4 où le calcul l'ILR montre qu'une personne de 16 ans, mesurant 1m72 et pesant 76 Kg est considérée comme étant « obèse ». Supposons que la condition d'activation est la suivante $< poids, \geq, 'normal' >$, qui est satisfaite si la personne est considérée comme « normale » ou « obèse » selon son « poids », sa « taille », et son « âge ». Le Tableau 18 montre les deux résultats qui satisfont la condition d'activation.

Tableau 18. Deux résultats pour la même vérification de conditions d'activation

Valeur Ling. Age	Valeur Ling. Taille	Poids	Valeur Ling Poids	Fonction d'appartenance	C_{Age}	C_{Taille}	C_{Poids}
Jeune	Normale	76 Kg	Normale	$\mu_{normale-normale}(76)$	0.5	0.2	0.67
Jeune	Normale	76 Kg	Obèse	$\mu_{normale-obèse}(76)$	0.5	0.2	0.33
Jeune	Grande	76 Kg	Normale	$\mu_{grande-normale}(76)$	0.5	0.8	1
Adulte	Normale	76 Kg	Normale	$\mu_{normale-normale}(76)$	0.5	0.75	0.67
Adulte	Normale	76 Kg	Obèse	$\mu_{normale-obèse}(76)$	0.5	0.75	0.33
Adulte	Grande	76 Kg	Normale	$\mu_{grande-normale}(76)$	0.5	0.25	1

Valeurs Lings. du chemin		M			Produit	Résultat 1	
		C_{Age}	C_{Taille}	C_{Poids}		ILR	Valeur ling.
Jeune-Normale-Normale	Σ	0.5	0.2	0.67	0.067	0.84325	'Normale'
Jeune-Grande-Normale		0.5	0.8	1	0.4		
Adulte-Normale-Normale		0.5	0.75	0.67	0.25125		
Adulte-Grande-Normale		0.5	0.25	1	0.125		

Valeurs Lings. du chemin		M			Produit	Résultat 2	
		C_{Age}	C_{Taille}	C_{Poids}		ILR	Valeur ling.
Jeune-Normale-Obèse	Σ	0.5	0.2	0.33	0.033	0. 15675	'Obèse'
Adulte-Normale-Obèse		0.5	0.75	0.33	0.12375		

Une deuxième condition $< tendance\ glycemie, =, '\text{à}\ la\ hausse' >$ est satisfaite si la tendance de la glycémie est à la hausse. La Figure 34 montre les résultats d'un algorithme de segmentation de séries chronologiques. Le premier résultat retourne deux *segments* avec une durée de deux jours et un jour respectivement. Le deuxième résultat retourne un seul segment d'une durée de trois jours. Nous partons du principe que les deux résultats (les trois segments) sont exploitables lors de la détection d'alertes. Le Tableau 19 liste les résultats dont la valeur linguistique associée au segment est une « tendance à la hausse ». La fonction d'appartenance utilise l'*angle* du segment et retourne le degré de certitude d'utiliser la valeur « à la hausse » pour représenter le segment.

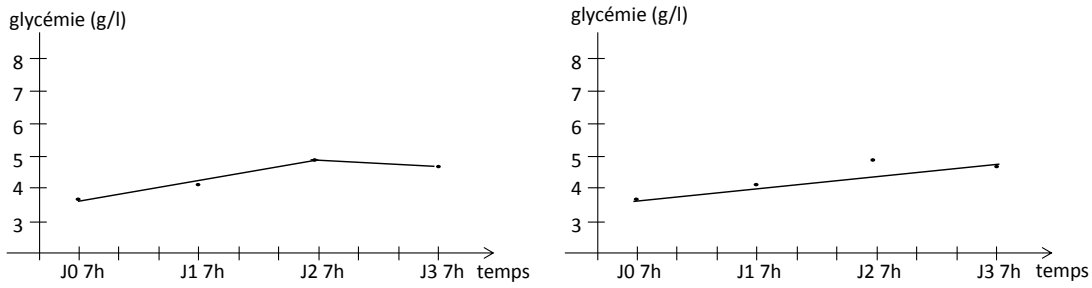


Figure 34. Deux tendances sur la même série chronologique de la glycémie

Tableau 19. Deux résultats pour la segmentation d'une série chronologique

Résultat 1					
Segment	**Angle(°)**	**Valeur Ling.**	**Fonction d'appartenance**	**$C_{Tendance\ Glycémie}$**	**IL**
[J0 7h – J2 7h]	21.3	A la hausse	$\mu_{\text{à } la\ hausse}(21.3)$	1	1
[J2 7h – J3 7h]	-10.20	A la hausse	$\mu_{\text{à } la\ hausse}(-10.20)$	0	0
Résultat 2					
[J0 7h – J3 7h]	11.3	A la hausse	$\mu_{\text{à } la\ hausse}(11.3)$	0.8	0.8

Plus il y a de conditions formant une situation d'alerte, plus il y a de combinaisons possibles entre les résultats satisfaisant les conditions, et plus il y a d'alertes détectées. Le Tableau 20 résume les résultats satisfaisant les deux conditions d'activation précédentes ainsi que les quatre combinaisons possibles. Chacune des combinaisons est une alerte détectée.

Tableau 20. Satisfaction des conditions d'activation

Condition		
$< poids, \geq, 'normal' >$		
Observation	**Val ling.**	**ISC**
76 kg	Normale	0. 84325
76 kg	Obèse	1
Condition		
$< tendance\ glycemie, =, '\text{ à } la\ hausse' >$		
Segment	**Val ling.**	**ISC**
[J0 7h – J2 7h]	A la hausse	1
[J0 7h – J3 7h]	A la hausse	0.8

ID	Observation Poids – Val Ling.	Segment Glycémie – Val Ling.	IAA
Ad_1	76 Kg - Normale	[J0 7h – J2 7h] - A la hausse	0.92
Ad_2	76 Kg - Normale	[J0 7h – J3 7h] - A la hausse	0.82
Ad_3	76 Kg - Obèse	[J0 7h – J2 7h] - A la hausse	1
Ad_4	76 Kg - Obèse	[J0 7h – J3 7h] - A la hausse	0.9

La politique de notification applique deux filtres et une agrégation pour réduire le nombre d'alertes détectées. D'abord, le filtre par qualité, le filtrage par expressivité et l'agrégation par nombre.

4.6.1.1 Filtrage par qualité

Notons que le Tableau 20 montre des résultats avec un IAA supérieur à zéro, ce qui signifie des alertes qui concernent une entité. Toutefois, même si l'IAA d'une alerte détectée est supérieure à zéro, elle ne devrait pas être considérée comme une alerte.

Définition 27 : Filtrage par qualité

Le filtrage par qualité fixe deux seuils, un pour l'indice d'applicabilité de l'alerte (IAA) et un autre pour l'indice de confiance de l'alerte (ICA). Si les valeurs des indices dépassent les seuils, l'alerte détectée est considérée comme telle.

Le filtrage par qualité ignore des alertes détectées dont les indices ne dépassent pas les seuils propres à la situation d'alerte. Ainsi, un seuil de 0.9 pour l'IAA sur les alertes du Tableau 20 réduit le nombre d'alertes détectées à trois. Puisque les alertes sont toujours nombreuses, la politique de notification applique un deuxième filtre.

4.6.1.2 Filtrage par expressivité

Le filtrage par expressivité s'applique sur toutes les alertes détectées qui dépassent une qualité minimale requise. Ce filtrage choisit, si possible, une seule alerte détectée qui traduit le mieux la situation d'alerte qui a été évaluée. Il est divisé en deux sous-filtrages le premier par expressivité linguistique et le deuxième par longévité de la tendance.

4.6.1.2.1 Filtrage par expressivité linguistique

Définition 28 : Filtrage par expressivité linguistique

Le filtrage par expressivité linguistique choisit les alertes détectées dont les valeurs linguistiques expriment le mieux la situation d'alerte :

- celle dont la valeur linguistique est la plus inférieure et la condition d'activation intrinsèque est formée par l'opérateur « ≤ »,

- celle dont la valeur linguistique est la plus supérieure et la condition est formée par l'opérateur « ≥ »,

- ou celle dont la valeur linguistique est la même que celle utilisée dans la condition formée par l'opérateur « = ».

Exemple

Si la condition « température ≥ normale » est satisfaite par une température de 41 °C représentée par les valeurs linguistiques « élevée » et « très élevée », le filtrage choisit la valeur « très élevée »

Si la condition « tension systolique ≤ normale » est satisfaite par une tension de 95 mm Hg représentée par les valeurs linguistiques « Hypotension » et « normale », le filtrage choisit la valeur « Hypotension »

Le filtrage par expressivité garde non seulement les alertes les plus expressives, mais indirectement celle avec un IAA plus élevé. Sur l'exemple du Tableau 20, les alertes « Ad_3 » et « Ad_4 » sont choisies à la place des alertes « Ad_1 » et « Ad_2 » parce que l'indice de satisfaction de la condition intrinsèque (ISCI) favorise déjà l'expressivité linguistique dans son calcul dans la formule (8).

Supposons que les valeurs linguistiques pour représenter le poids sont « maigre », « normal », « surpoids », « obèse », et « obèse extrême ». À partir des alertes détectées dans le Tableau 21, le filtrage par expressivité choisit les alertes détectées « Ad_7 » et « Ad_8 » plutôt que les alertes « Ad_5 » et « Ad_6 » parce l'opérateur de la condition construite avec le « poids » est « ≥ » et que la valeur linguistique est « obèse » ≥ « surpoids ».

Tableau 21. Satisfaction des conditions d'activation avec des IAA égales

ID	$< poids, \geq, 'normale' >$	$< tendance\ glycemie, =, '$ à $la\ hausse' >$	IAA
	Observation Poids – Val Ling.	**Segment Glycémie – Val Ling.**	
Ad_5	120 Kg - surpoids	[J0 7h – J2 7h] - A la hausse	1
Ad_6	120 Kg - surpoids	[J0 7h – J3 7h] - A la hausse	0.9
Ad_7	120 Kg – Obèse	[J0 7h – J2 7h] - A la hausse	1
Ad_8	120 Kg – Obèse	[J0 7h – J3 7h] - A la hausse	0.9

4.6.1.2.2 Filtrage par longévité de la tendance

Définition 29 : Filtrage par longévité de la tendance

Le filtrage par longévité de la tendance choisit les alertes détectées dont il n'existe pas un segment, autre que celui associé à la tendance, qui approxime la totalité des observations approximées par le segment associé à la tendance.

Les tendances sont des valeurs linguistiques associées aux segments. Un segment est l'approximation d'un ensemble d'observations d'une ressource observable. Selon l'algorithme de segmentation utilisé, des ensembles d'observations peuvent être approximés par deux segments différents. Dans quelques cas, toutes les observations approximées par un segment peuvent l'être aussi par un autre différent.

A partir des alertes détectées listées dans le Tableau 21, le filtrage par longévité de la tendance choisit les alertes détectées « Ad_6 » et « Ad_8 » au lieu des alertes « Ad_5 » et « Ad_7 » puisque le segment qui approxime les observations entre « [J0 7h – J3 7h] » approxime aussi celles entre « [J0 7h – J2 7h] ». Ce choix est fait bien que l'IAA des alertes « Ad_5 » et « Ad_7 »soit supérieur à celui des alertes « Ad_6 » et « Ad_8 » étant donné que les alertes sur lesquelles s'applique le filtrage par longévité de la tendance ont déjà passé le filtrage par qualité minimale exigée.

La Figure 35 montre une série chronologique découpée en trois segments. Le Tableau 22 contient deux alertes détectées. Dans ce cas, les deux alertes détectées sont candidates à la notification puisque les segments approximent des observations différentes.

Tableau 22. Satisfaction des conditions d'activation avec des segments dont l'intersection est vide

ID	$< poids, \geq, 'normale' >$	$< tendance\ glycemie, =, '\text{ à } la\ hausse' >$	IAA
	Observation Poids – Val Ling.	**Segment Glycémie – Val Ling.**	
Ad_9	120 Kg – Obèse	[J0 7h – J1 7h] - A la hausse	1
Ad_10	120 Kg – Obèse	[J2 7h – J3 7h] - A la hausse	1

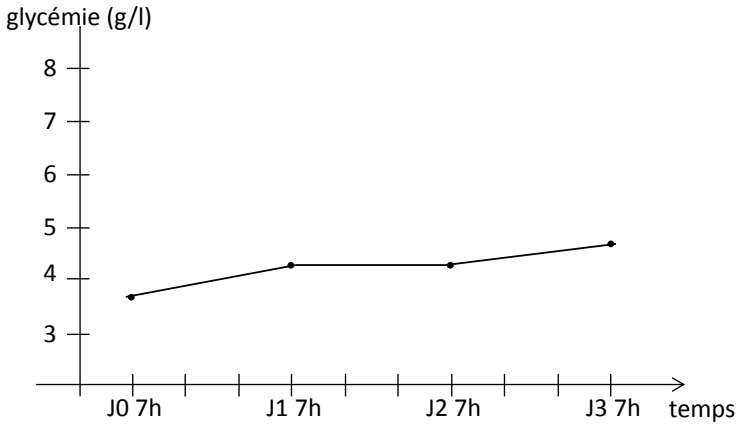


Figure 35. Trois segments sur la série chronologique de la glycémie

4.6.1.3 Agrégation par détection

Définition 30 : Agrégation par détection

L'agrégation par détection construit une seule alerte à partir de toutes les alertes détectées ayant passé les filtrages par qualité et par expressivité.

Plusieurs alertes détectées peuvent passer les filtrages par qualité et par expressivité. C'est le cas pour les deux alertes dans le Tableau 22. Si le nombre de conditions d'activation utilisées pour définir une situation d'alerte augmente, cela sera aussi probablement le cas pour le nombre d'alertes détectées et filtrées.

Comme toutes les alertes détectées sont issues de la même évaluation d'une situation d'alerte, nous avons choisi de notifier une seule *alerte accessible* qui regroupe dans un seul *message* toutes les informations déclenchant l'alerte, au lieu de notifier chacune des alertes détectées. Une *alerte accessible* est accompagnée d'un *message* construit à l'aide les *observations* et les *segments* des alertes détectées

ordonnées chronologiquement. Le *message* de l'alerte accessible construit à partir des alertes détectées dans le Tableau 22 est « M. Dupont pèse 120 Kg, il est obèse. Une tendance à la hausse de la glycémie a eu lieu de J0 à 7h à J1 à 7h et une autre de J2 à 7h à J3 à 7h ».

L'IAA et l'ICA en plus d'être utilisés pour filtrer des alertes, expriment des informations exploitables par les utilisateurs, par exemple, pour prioriser des tâches associées aux alertes, comme le fait de surveiller le patient. L'IAA et l'ICA de l'alerte accessible sont l'IAA et l'ICA les plus élevés parmi les alertes détectées. Ainsi, l'IAA de l'alerte accessible construit à partir des alertes détectées dans le Tableau 22 est égale à 1.

4.6.2 Notification par comparaison avec des alertes déjà notifiées

Nous différencions deux types de notification, la notification implicite et la notification explicite. La notification implicite rend accessible une alerte aux utilisateurs tandis qu'une alerte explicite en plus de la rendre accessible, fait l'objet d'un signalement. La *comparaison avec des alertes déjà notifiées* est liée à la notification explicite.

Définition 31 : Cycle de vie de l'alerte

Le cycle de vie de l'alerte est un mécanisme qui permet de contrôler la notification explicite d'alertes accessibles.

Le cycle de vie de l'alerte compare l'alerte accessible (si elle existe) avec celles déjà notifiées concernant la même situation d'alerte et la même entité. Deux nouveaux concepts ont été définis afin de fixer quatre scénarii possibles : le temps de validité de l'alerte, et l'état de l'alerte.

4.6.2.1 Temps de validité et état de l'alerte

Définition 32 : Temps de validité de l'alerte

Le temps de validité de l'alerte exprime combien de temps après qu'une situation d'alerte a été identifiée, elle n'est plus d'actualité.

Le temps de validité est propre de la situation d'alerte. Les alertes accessibles concernant la même situation d'alerte ont le même temps de validité. En principe, il est possible d'agir sur une alerte accessible seulement pendant qu'elle est actuelle. Une fois le temps de validité écoulé, il est théoriquement trop tard pour la traiter. Un temps de validité infini exprime qu'il n'est jamais trop tard pour agir sur l'alerte puisqu'elle sera toujours d'actualité.

Définition 33 : Etat de l'alerte

L'état de l'alerte exprime la manière dont l'utilisateur peut interpréter une alerte accessible.

Trois états ont été définis pour les alertes accessibles : *nouvelle*, *traitée*, et *périmée*. L'état *nouvelle* exprime qu'une alerte vient de devenir accessible (après l'évaluation d'une situation d'alerte). L'état *traitée* exprime qu'un utilisateur a utilisé l'alerte accessible. L'état *périmée* exprime que le temps de validité de l'alerte s'est écoulé et qu'aucun utilisateur ne l'a utilisée.

4.6.2.2 Scénarii de notification

Le choix de notifier, ou non, explicitement est dépendant de la finalité de l'alerte. Quatre scénarii ont été définis à partir du temps de validité et de l'état de l'alerte afin de déterminer si une alerte accessible fait l'objet d'un signalement (notification explicite). Chaque scenario compare une alerte qui vient de devenir accessible avec la, ou les, dernière(s) alerte(s) précédemment accessible(s). Si c'est la première alerte accessible concernant une situation d'alerte et une même entité alors la notification est explicite. Sinon, un des quatre scénarii de la Figure 36 s'applique.

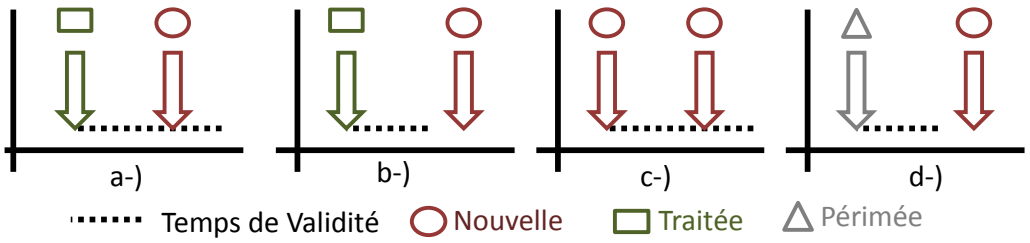


Figure 36. Scenarii du cycle de vie de l'alerte

- Scenario *nouvelle-pendant-traitée* (Figure 36 a) : ce scenario spécifie s'il faut notifier, ou non, explicitement une alerte *nouvelle* qui vient de devenir accessible pendant le temps de validité d'une autre alerte dont l'état est *traitée.*
- Scenario *nouvelle-après-traitée* (Figure 36 b) : ce scenario spécifie s'il faut notifier, ou non, explicitement une alerte *nouvelle* qui vient de devenir accessible après le temps de validité de la dernière alerte dont l'état est *traitée.*
- Scenario *nouvelle-pendant-nouvelle* (Figure 36 c) : ce scenario spécifie s'il faut notifier, ou non, explicitement une alerte *nouvelle* qui vient de devenir accessible pendant le temps de validité d'une alerte dont l'état est *nouvelle.*
- Scenario *nouvelle-après-périmée* (Figure 36 d) : ce scenario spécifie s'il faut notifier, ou non, explicitement une alerte *nouvelle* qui vient de devenir accessible après le temps de validité de la dernière alerte dont l'état est *périmée.*

Enrichir un scenario permet de mieux définir le comportement des notifications explicites lorsque des alertes deviennent accessibles consécutivement. Tous les scénarii sont enrichis avec les indices de qualité de l'alerte afin de mieux spécifier quand la notification explicite doit avoir lieu. Les scénarii *nouvelle-pendant-traitée* et

nouvelle-pendant-nouvelle sont enrichis en fonction du système qui met à jour les propriétés des alertes accessibles plus récentes avec celles de l'alerte accessible la plus ancienne ou les propriétés des alertes accessibles plus anciennes avec celles de l'alerte accessible la plus récente. Finalement le scenario *nouvelle-pendant-traitée* est aussi enrichi par héritage d'état dont la dernière alerte accessible hérite de l'état de l'alerte accessible précédente.

Définition 34 : Enrichissement par qualité

L'enrichissement par qualité exprime s'il faut notifier explicitement une alerte si l'indice d'applicabilité ou de confiance de l'alerte augmente ou diminue.

Exemple

Imaginons une situation d'alerte simple définie avec une seule condition d'activation $< température, =, 'élevée' >$. Une première évaluation de la situation d'alerte mène à utiliser la dernière observation connue de la température qui est de 38°C. Une alerte avec un IAA de 0.9 et un ICA de 0.8 est notifiée explicitement aux infirmières. Une heure après, le système évalue encore une fois la situation d'alerte qui utilise une température de 38.5°C et qui rend accessible une nouvelle alerte avec un IAA de 1 et un ICA de 0.9. Cela signifie que la deuxième alerte concerne complètement le patient, et que l'observation utilisée est plus proche de l'instant d'évaluation. La Figure 37 illustre l'exemple cité.

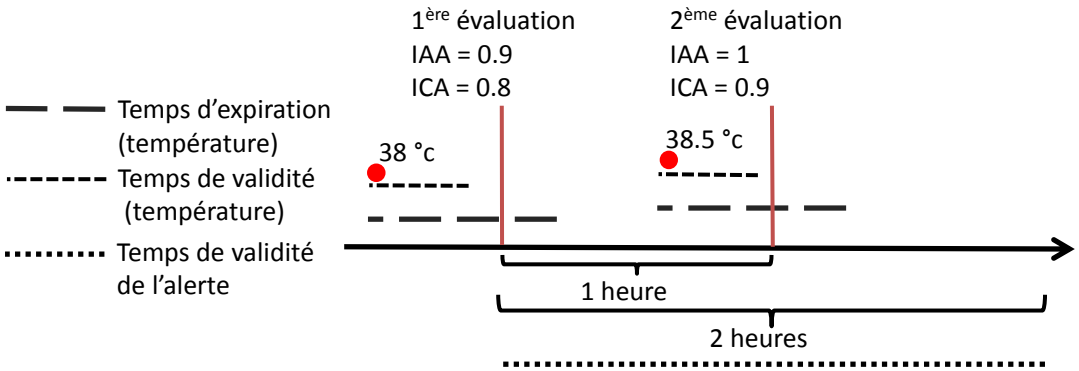


Figure 37. Deux alertes consécutives

Exemple

Supposons que la **première notification explicite** a conduit les infirmières à administrer du paracétamol au patient. Elles souhaiteront être alertées pour savoir si c'était le **médicament indiqué** et donc si la température ne monte pas. Les infirmières spécifient donc pour le scenario ***nouvelle-pendant-traitée*** la notification doit être **explicite si l'IAA augmente**. A partir de l'exemple illustré dans la Figure 37, l'alerte issue de **la deuxième évaluation devrait être notifiée explicitement** aux infirmières puisque le patient est plus concerné par l'alerte, ce qui se traduit par : la température n'a pas baissé.

Imaginons que la **première alerte n'a pas été traitée**. Elle n'a pas été utilisée donc aucune action n'a été prise parce que dans la plupart des cas la température baisse. Les infirmières définissent donc que la **notification doit être explicite** dans les scénarii ***nouvelle-pendant-nouvelle*** et ***nouvelle-après-périmée*** si **l'IAA a augmenté.**

Définition 35 : Enrichissement par « écrasement »

L'enrichissement par « écrasement » remplace les propriétés des alertes accessibles plus récentes avec celles de l'alerte accessible la plus ancienne ou les propriétés des alertes accessibles plus anciennes avec celles de l'alerte accessible la plus récente.

Les scénarii *nouvelle-pendant-traitée* et *nouvelle-pendant-nouvelle* vérifient si une alerte accessible dont l'état est *nouvelle* se trouve pendant le temps de validité d'une alerte déjà notifiée. La Figure 38 montre un exemple de quatre alertes consécutives dont les deux du milieu sont devenues accessibles pendant le temps de validité de la première. En revanche, la quatrième est devenue accessible après le temps de validité de la première, mais pendant le temps de validité de la deuxième.

L'enrichissement par « écrasement » met à jour des propriétés d'une alerte dont l'état est *nouvelle* afin de contrôler la notification explicite pour les scénarii *nouvelle-pendant-traitée* et *nouvelle-pendant-nouvelle*. Par exemple, l'alerte accessible suite à la quatrième évaluation devrait être notifiée explicitement tandis que celles de la deuxième et troisième évaluation non.

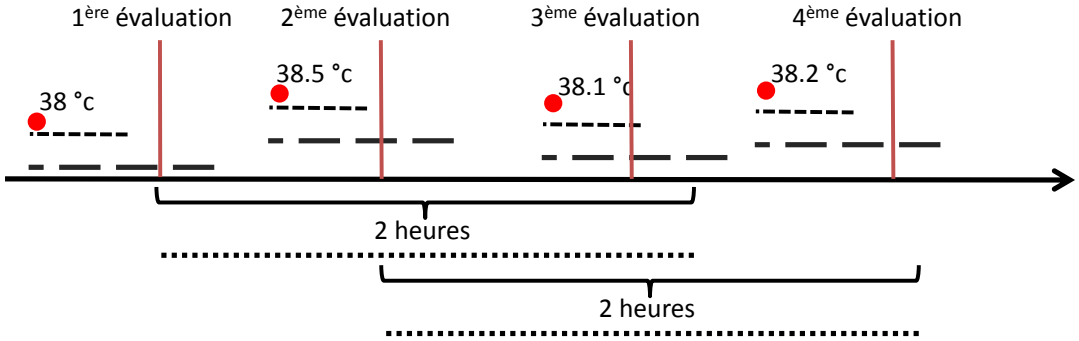


Figure 38. Quatre alertes consécutives

L'enrichissement peut se baser sur l'information la plus ancienne ou la plus récente. La Figure 39, et la Figure 40 illustrent les deux cas. Dans le premier cas, les alertes les plus récentes mettent à jour ses propriétés avec l'alerte la plus ancienne. Dans le deuxième, les alertes les plus récentes mettent à jour ses propriétés par rapport à l'alerte la plus récente. Les seules propriétés qui peuvent être mises à jour sont celles qui jouent un rôle dans le cycle de vie de l'alerte : le temps de validité, l'indice d'applicabilité de l'alerte et l'indice de confiance de l'alerte.

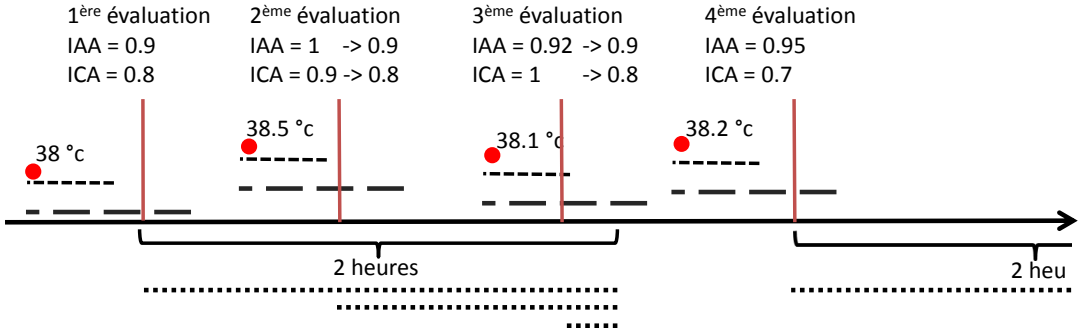


Figure 39. Enrichissement en se basant sur les données anciennes

Exemple

Supposons que la notification doit être explicite si **l'indice d'applicabilité augmente** dans le scenario ***nouvelle-après-périmée*** puisque les infirmières attendent une baisse de la température sans médicament. Si cela n'arrive pas, une **notification explicite** doit avoir lieu. L'enrichissement du **temps de validité** en se basant sur les informations **anciennes**, et l'enrichissement de l'**indice d'applicabilité** en se basant sur les informations **récentes** dans le scenario ***nouvelle-pendant-nouvelle*** génèrent une **notification explicite** lors de la **quatrième évaluation** puisque à cet instant-là, les temps de validité des alertes accessibles après la première, la deuxième et la troisième évaluation ont été dépassés et leur état n'est plus *nouvelle* mais *périmée*. Ceci provoque donc le scenario ***nouvelle-après-périmée*** où il a été défini une notification explicite.

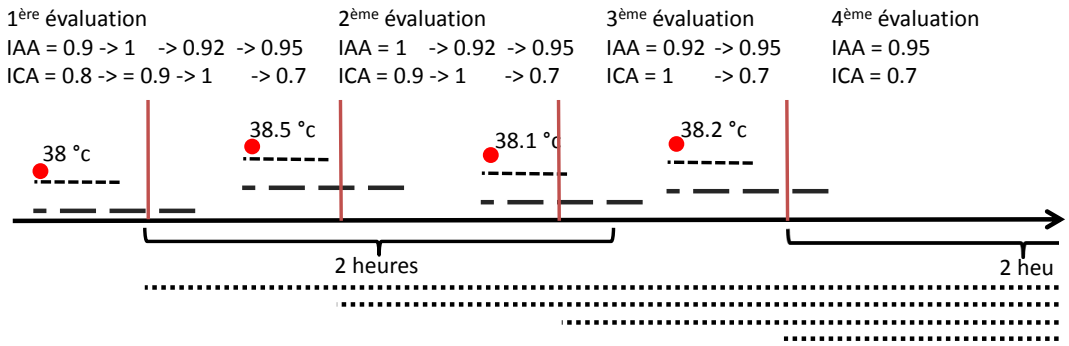


Figure 40. Enrichissement en se basant sur les données récentes

Définition 36 : <u>Enrichissement par état</u>

L'enrichissement par état hérite de l'état de l'alerte accessible notifiée précédemment

L'enrichissement par état sert à différencier les notifications explicites du scenario *nouvelle-pendant-traitée* avec celles du scenario *nouvelle-pendant-nouvelle*. Supposons que l'alerte accessible lors de la première évaluation dans la Figure 38 a été traitée et que la deuxième reste nouvelle. L'alerte accessible lors de la troisième évaluation se trouve dans deux scénarii, *nouvelle-pendant-traitée* et *nouvelle-pendant-nouvelle*. Si les deux scénarii ne sont pas enrichis de la même façon, une notification explicite peut avoir lieu lors de la troisième évaluation et perturber inutilement les utilisateurs. Selon la finalité de l'alerte, cela peut être inutile.

L'enrichissement par état s'applique seulement au scenario *nouvelle-pendant-traitée*. Ainsi, toute alerte qui est devenue accessible pendant le temps de validité d'une alerte dont l'état est « traitée » change automatiquement son état à « traitée ». Si l'enrichissement par état a lieu, l'alerte devenue accessible lors de la troisième évaluation se trouve seulement dans le scenario « nouvelle pendant traitée » et non dans « nouvelle pendant nouvelle ».

4.6.3 Synthèse de la section

Cette section a présenté une politique de notification qui permet de contrôler la notification d'alertes aux utilisateurs. La notification peut être implicite ou explicite des alertes. La notification implicite rend accessible une alerte aux utilisateurs, tandis que la notification explicite en plus de la rendre accessible attire l'attention des utilisateurs.

Selon les observations utilisées lors de l'évaluation d'une situation d'alerte définie en utilisant multiples conditions d'activation, des nombreuses alertes peuvent être détectées. Le premier contrôle s'applique sur ces alertes détectées afin d'éliminer autant que possible, les alertes de mauvaise qualité ou les faux-positifs et faux-négatifs et de les agréger dans une seule alerte accessible.

Le deuxième contrôle s'applique sur ces alertes accessibles afin de réduire la notification explicite d'alertes inutiles qui attirent inutilement l'attention des utilisateurs. La notification explicite est gérée par le cycle de vie de l'alerte. Celui-ci défini quatre scénarii à l'aide du temps de validité de l'alerte, l'état de l'alerte et les alertes déjà notifiées concernant la même situation d'alerte et la même entité. Les scénarii sont enrichis par qualité afin de réduire les notifications explicites, par « écrasement » pour faciliter le passage d'un scenario à un autre, et par état dans le but d'éviter des ambigüités.

4.7 Synthèse du chapitre

Ce chapitre a présenté TEMPAS, un système d'alertes qui permet de définir des situations d'alerte en utilisant nombreuses conditions d'activation. Les conditions d'activation sont intrinsèques ou temporelles. Les conditions intrinsèques expriment avec des valeurs linguistiques l'état (normal, élevé, etc.) ou la tendance (à la baisse, stable, à la hausse, etc.) d'une ressource (température), tandis que les conditions temporelles expriment qu'une condition intrinsèque est vérifiée avant ou en même temps qu'une autre.

Les situations d'alerte sont adaptables au contexte grâce aux informations disponibles afin de mieux définir la situation d'alerte : les valeurs linguistiques utilisées pour définir l'état ou la tendance d'une ressource observable ; la fréquence des observations ; l'utilisation des nouvelles ressources pour définir d'autres conditions d'activation ou pour modifier le calcul des valeurs linguistiques d'autres ressources.

L'évaluation des situations d'alerte sur une entité mène à utiliser les observations des ressources observables afin de conclure si la situation d'alerte a, ou non, lieu. Cette évaluation vérifie donc les conditions d'activation qui composent la situation d'alerte. Ce processus calcule neuf indices de qualité dont deux sont attribués aux alertes détectées, l'indice d'applicabilité de l'alerte (IAA) et l'indice de confiance de l'alerte (ICA). Le premier indice exprime dans quelle mesure une entité est concernée par une alerte, alors que le deuxième traduit la fraîcheur des données utilisées.

Finalement, une politique de notification est définie pour chaque situation d'alerte et est appliquée aux alertes détectées instanciant la situation d'alerte sur une entité. L'objectif de la politique de notification est de contrôler la sur-notification d'alertes. Nous différencions deux types de notification : la notification implicite où l'alerte est accessible, et la notification explicite où l'alerte est accessible et fait l'objet d'un signalement aux utilisateurs. La politique de notification compte deux contrôles. Le premier contrôle s'applique sur les alertes détectées lors de l'évaluation d'une situation d'alerte afin de créer une seule alerte accessible (un vrai-positif) avec un message qui décrit la situation d'alerte. Le message est construit avec des valeurs linguistiques propres au contexte de l'alerte. Un filtrage par qualité ainsi qu'un filtrage par expressivité suivi d'une agrégation permettent de construire une seule alerte accessible. Le deuxième contrôle s'applique aux alertes qui viennent de devenir accessibles afin de déterminer s'il est nécessaire de faire une notification explicite (la signaler aux utilisateurs), ou si la notification implicite suffit. Pour cela, il est nécessaire d'évaluer quatre scenarii correspondants au cycle de vie de l'alerte. Chaque scenario est enrichi afin de mieux spécifier quand la notification explicite doit avoir lieu.

Nous présentons dans le chapitre 6 un exemple complet récapitulatif, depuis la préparation de connaissances jusqu'à la notification des alertes pour la surveillance de la dialyse à domicile.

5 Implémentation et intégration du moteur d'alertes TEMPAS

5.1 Introduction 129
5.2 Couche des données 131
5.3 Evaluation des alertes 135
5.4 Intégration de TEMPAS avec FSD 142
5.5 Synthèse 164

5.1 Introduction

Le moteur d'alertes de TEMPAS a été conçu pour détecter des alertes indépendamment du domaine d'application. Un middleware a été développé pour faciliter l'intégration du moteur avec les systèmes d'information qui s'appuient sur des modèles de données relationnels. Il a été implémenté et intégré à *Futura Smart Design*, un système de gestion d'informations de santé développé par Calystene[1]. Dans le cadre de cette thèse nous avons développé :

- les interfaces graphiques et la logique-métier pour la création, l'affichage, le stockage, la modification et la suppression des situations d'alerte et des connaissances associées ;
- les services responsables de l'évaluation des alertes, la récupération des alertes détectées et la notification des alertes accessibles
- un middleware pour l'intégration du moteur d'alertes avec d'autres systèmes d'information fondés sur une base de données centralisée et relationnelle
- le moteur d'alertes qui détermine si une situation d'alerte a lieu.

Nous présentons Futura Smart Design et TEMPAS. La couche des données est présentée dans la section 5.2. L'évaluation des alertes est traitée dans la section 5.3, et finalement, l'intégration est expliquée dans la section 5.4.

5.1.1 Futura Smart Design

Futura smart design (FSD) est un système de gestion d'informations de santé. Parmi les domaines fonctionnels gérés, on trouve la prescription, la pharmacie, la planification, l'organisation des soins, la gestion du bloc opératoire, la gestion des lits, les urgences et la facturation. Ces services génèrent de nombreuses données susceptibles d'être utilisées dans la définition de situations d'alerte intéressantes pour des utilisateurs.

[1] http://futura-smart-design.catalyzis-group.com/

La solution Futura Smart Design est une application web internet riche dont l'architecture est orientée services. L'application cliente est animée grâce au Framework Silverlight et exécutée par un navigateur web. Toutefois, les services peuvent être exposés comme des services web grâce à soap\wsdl afin d'être exploités par d'autres applications clientes. Les services sont développés avec des technologies faisant partie du Framework .NET. Les données sont stockées dans une base de données relationnelle et un framework y accède pour la correspondance avec des objets relationnels utilisés dans la logique-métier.

Futura Smart Design a été construit pour fournir une interface utilisateur très ergonomique et intuitive pour faciliter la prise en main de la solution.

Comme la plupart des systèmes d'information, FSD fournit des situations d'alerte pré-paramétrées sur des variables spécifiques similaires à « si température supérieure à 39 °C». L'interaction des utilisateurs était restreinte à : modifier le seuil de comparaison ou désactiver l'alerte. Le projet TEMPAS a permis d'ajouter des situations d'alerte paramétrées par les utilisateurs, définies par des valeurs linguistiques qui expriment l'état et la tendance d'une ressource observable, multi-paramètres, formées par des données hétérogènes et évaluées sur des groupes de patients respectant un critère spécifique. TEMPAS ajoute aussi une gestion souple des alertes détectées. Il est possible de définir une politique de notification en se basant sur divers critères contextuels et de qualité.

5.1.2 TEMPAS

TEMPAS est composé de trois parties : le middleware d'intégration, la détection de tendances, et le moteur d'alertes. Les fonctionnalités des deux premières parties sont facultatives à condition que le système maître les fournisse lui-même. Les trois parties ont été développées en Prolog et font partie du même exécutable.

Le *middleware* joue le rôle de pont entre un système d'information et le moteur d'alertes TEMPAS. Il connecte bi-directionnellement les deux systèmes grâce à une base de données relationnelle. Ainsi, le moteur d'alertes exploite les données de FSD afin de détecter des alertes et les rendre accessibles à FSD pour qu'il puisse les récupérer afin de les notifier aux utilisateurs. L'interaction faite au niveau des données facilite l'intégration avec d'autres systèmes d'information.

La détection des tendances n'est pas une fonctionnalité courante dans les systèmes d'information. Dans le but de concevoir des situations d'alerte plus expressives, TEMPAS incorpore une fonctionnalité pour la détection de tendances indépendamment du domaine d'application.

Finalement, le *moteur* d'alertes, le cœur de TEMPAS, est chargé de déterminer si un ensemble de données convient à une situation d'alerte. Le cas échéant, deux indices de qualité sont attribués à chaque alerte détectée : l'indice d'applicabilité et l'indice de confiance.

5.2 *Couche des données*

L'intégration commence avec la couche des données. Deux aspects sont abordés : la définition d'un modèle de données qui représente les connaissances préparées par les utilisateurs ainsi que les situations d'alerte qu'ils définissent et la traduction des données hétérogènes qui se trouvent quelque part dans la base de données de FSD (système maître) dans un format unique exploitable lors de l'évaluation des situations d'alerte.

5.2.1 Schéma de données de TEMPAS

Le schéma de la base de données relationnelle est exploité par le middleware qui facilite l'intégration de TEMPAS avec le système maître (Figure 41). Sur le schéma de la Figure 41 nous avons omis les noms d'associations pour plus de lisibilité. Les situations d'alerte, les données impliquées dans sa définition et nécessaires lors de son évaluation sont représentées dans le schéma de la base de données. Il représente aussi les alertes détectées susceptibles d'être notifiées aux utilisateurs. Le middleware crée une base de données temporelle exploitable par le moteur d'alertes à partir du schéma.

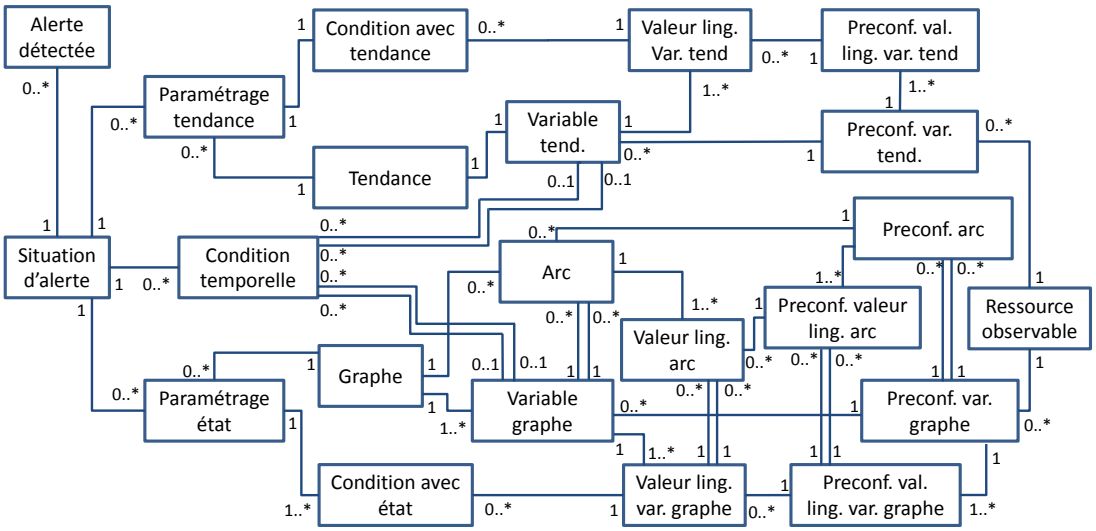


Figure 41. Schéma de la base de données

5.2.2 Séparation des modèles

Rappelons-nous que les situations d'alerte de TEMPAS sont adaptables au contexte parce qu'il est possible d'utiliser de nombreuses informations pour mieux définir les situations d'alerte. Ces informations se trouvent dans plusieurs tables du modèle de données du système d'information maître et doivent être centralisées dans une seule table représentant les *ressources observables* du modèle de données de TEMPAS. La correspondance est réussie grâce à la séparation des modèles illustrée dans la Figure 42.

Les liens entre les entités du modèle de FSD et la table des ressources observables sont des références faibles (absence de clé étrangère) puisque, à la

différence des lignes continues, elles n'expriment pas l'intégrité référentielle, c'est-à-dire que la clé primaire de l'entité de FSD n'est pas une clé étrangère dans la table des ressources observables.

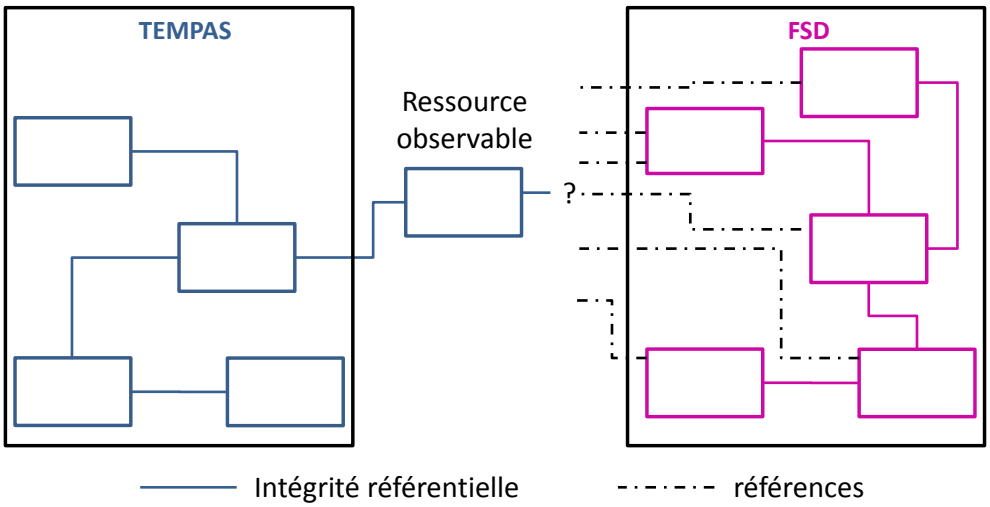


Figure 42. Séparation des modèles

La Figure 43 montre comment les informations dans le modèle de données de FSD sont référencées par des *ressources observables* utilisées pour la définition des *situations d'alerte*.

Exemple

Les « plaquettes » et les « triglycérides » sont des « résultats de laboratoire » dans FSD et des *ressources observables* pour TEMPAS.

Le « poids », le « pouls » et la « taille » sont des « paramètres du patient » dans FSD et des *ressources observables* pour TEMPAS.

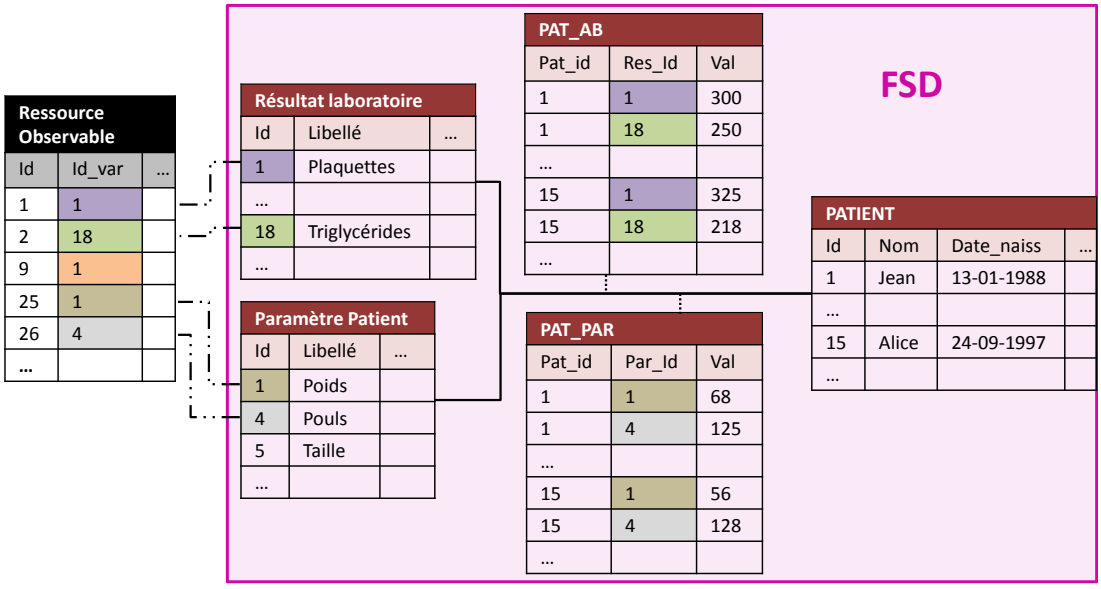


Figure 43. Références faibles

5.2.3 Modélisation des observations

Dans TEMPAS, une *observation* est liée à une *ressource observable* et à une *entité* (un patient). Elle a deux attributs, un pour la *valeur observée*, et un autre pour le *temps d'observation*. Grâce à l'utilisation des *vues* (Figure 44), le middleware sait comment récupérer les observations des ressources utilisées dans la définition des situations d'alerte afin de les transformer postérieurement dans un modèle exploitable par le moteur d'alertes.

Vue_pat_ab			
Entity_id	Var_id	value	timestamp
1	1	300	01-07-2014 17h45
1	18	250	01-08-2014 18h20
…			
15	1	325	01-07-2014 17h45
15	18	218	01-08-2014 18h20
…			

Vue_pat_par			
Entity_id	Var_id	value	timestamp
1	1	68	06-08-2014 18h45
1	4	125	06-08-2014 18h45
…			
15	1	56	06-08-2014 18h55
15	4	128	06-08-2014 18h55
…			

Vue_pat_age		
Entity_id	Var_id	value
1	1	24
…		
15	1	14
…		

Figure 44. Vues contenant les observations

Les vues projettent l'identifiant de l'*entité* (patient), l'identifiant de la *ressource observable*, la *valeur mesurée*, et l'*horodatage*. L'horodatage est facultatif si la valeur mesurée est toujours d'actualité. C'est le cas pour toutes les *ressources observables* qui expriment une dimension temporelle et qui se calculent à partir de la date courante puisque dans ce cas, le système d'information ne stocke pas les valeurs qui représentent la ressource observable mais des dates qui permettent de les calculer.

Exemple

L'« âge » d'un patient, le « nombre de jours depuis la dernière ordonnance » et le « nombre de jours depuis la dernière intervention » sont toujours d'actualité.

5.2.4 Accès aux observations

La relation entre la table ressources observables et les vues se trouve dans la Figure 45. Les vues fournissent l'accès aux observations d'une ressource pour une entité. Si la ressource, dans le modèle de FSD, disparaît, la vue ne contiendra pas d'observations qui la concernent et le moteur d'alertes ne détectera pas d'erreurs. Toutefois, la ressource observable continuera à être présente dans la base parce qu'il n'y a pas d'intégrité référentielle en cascade vers la table *ressource observable* du modèle de TEMPAS. Des triggers ont été déployés dans le système de gestion de base de données de FSD pour :

- effacer en cascade ;
- ajouter des nouvelles *ressources observables*, par exemple lorsqu'un nouveau « score patient » est ajouté ;
- et mettre à jour des informations des ressources observables comme le « libellé », ou les valeurs « minimale » et « maximale » mesurables.

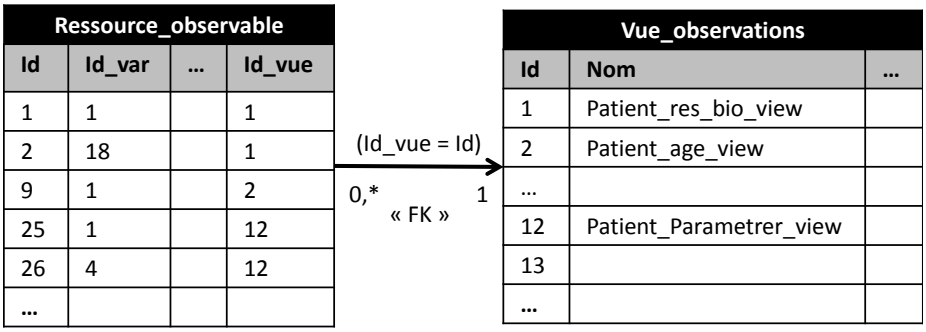


Figure 45. Relation ressources observables et vues

Les utilisateurs du système d'information peuvent utiliser toutes les ressources observables pour concevoir des situations d'alerte. Pour rendre des nouvelles *ressources accessibles* aux utilisateurs il suffit de :

- créer la vue qui contient les observations de la ressource dans le format indiqué ;
- ajouter la vue dans la table des vues des observations ;
- ajouter une ligne dans la table de ressources observables faisant le lien avec la vue qui contient les observations ;
- et déployer les triggers pour effacer en cascade, ajouter des nouvelles ressources dont les observations sont déjà dans la vue créée, et mettre à jour les informations de la ressource.

5.2.5 Groupe d'entités

Le modèle de données de TEMPAS a été enrichi pour créer des *groupes d'entités* (patients) pour cibler la population concernée. Le principe utilisé pour accéder aux patients faisant partie d'un *groupe* est similaire de celui utilisé pour accéder aux observations des ressources observables.

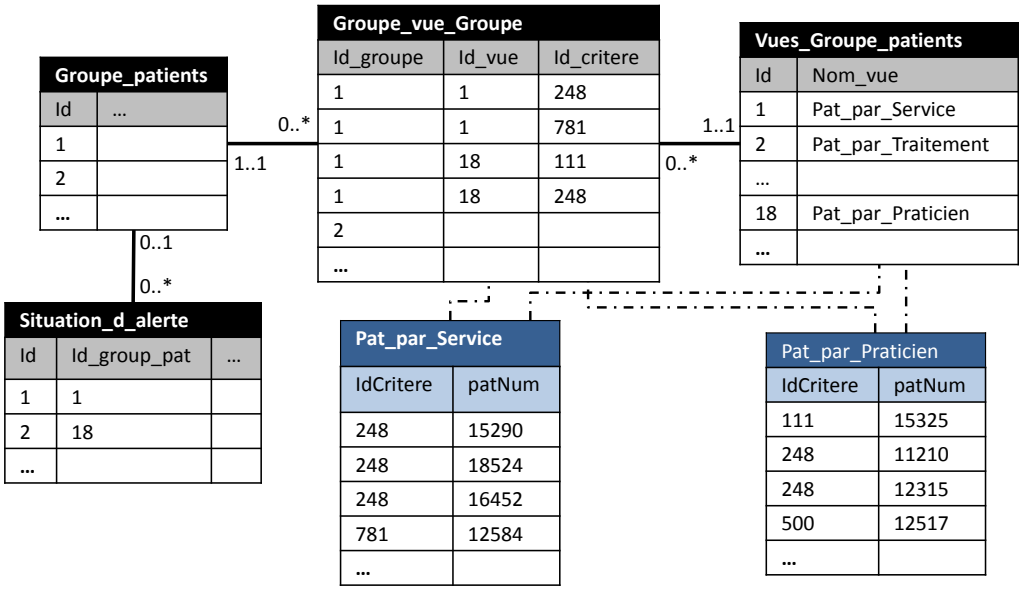


Figure 46. Groupes de patients

La Figure 46 montre la liaison entre les *situations d'alerte* et les *groupes de patients*. Les groupes de patients sont formés à l'aide de vues qui regroupent les patients selon des critères spécifiques propres au système d'information maître. Les groupes de patients sont calculés dynamiquement.

Exemple

La vue *Pat_par_service* contient les patients qui se trouvent dans un service hospitalier spécifique (*IdCritere*) lorsqu'elle est consultée.

Le patient est concerné par des situations d'alerte différentes selon le service hospitalier où il se trouve. Les groupes de patients, en plus d'aider à la contextualisation des situations d'alerte, améliorent la performance de leur évaluation.

5.3 Evaluation des alertes

L'évaluation d'une alerte consiste à évaluer une situation d'alerte sur une entité, ou sur un ensemble d'entités. Le middleware crée une base de données temporelle avec les connaissances préparées par les utilisateurs qui sont utilisées pour la définition de la situation d'alerte à évaluer, soit : les *conditions intrinsèques et temporelles* définissant la situation d'alerte ; les *valeurs linguistiques* et les *ensembles flous* respectifs des *ressources observables* ainsi que leur *temps de validité* et leur *temps d'expiration ;* et les informations pour la détection des tendances.

5.3.1 Détection des tendances

La détection de tendances est faite par un algorithme ascendant (bottom-up) qui détecte les k segments les plus représentatifs qui représentent une série chronologique formée par n observations. Les valeurs observées sont numériques.

5.3.1.1 Critères pour la sélection des observations

Trois critères saisis par l'utilisateur permettent de sélectionner les observations : la durée de la période maximale $maxT$ qui contient les observations de la série chronologique, la durée souhaitée $normalT$ de la série chronologique, et le nombre $minimal$ (ou $maximal$) de points $normalP$ qui doit former la série chronologique. Le seul critère obligatoire est $maxT$ qui sert à restreindre les observations lors de l'absence de $normalT$ ou $normalP$. La série chronologique résultante est celle avec une durée inférieure ou égale à $normalT$, qui est composée au $minimum$ (ou au $maximum$) par $normalP$ observations comprises dans la période $[\tau - maxT, \tau]$ où τ est l'instant d'évaluation. La Figure 47 montre les séries chronologiques représentées par des segments connexes obtenus en appliquant un $maxT$ de 48h, un $normalT$ de 24h et un $normalP$ de 8 observations sur des ensembles d'observations différentes.

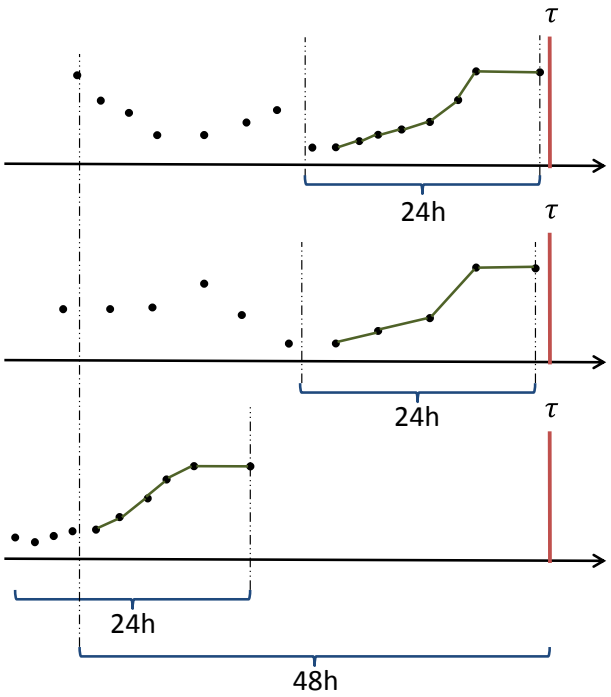


Figure 47. Sélection d'observations pour la détection de tendances

5.3.1.2 Normalisation des propriétés des observations

L'algorithme de détection de tendances trouve k segments avec des longueurs semblables. La longueur du segment est la distance bidimensionnelle entre les deux observations qui le forment. L'algorithme normalise les propriétés des observations, la valeur observée et l'horodatage.

La *normalisation* est une fonction qui retourne une valeur entre 0 et 1. Ainsi, le plus grand changement dans n'importe lequel des deux axes est au maximum égal à 1. Le Tableau 23 montre la normalisation de cinq observations de « température » dont la *valeur minimale réelle* et la *valeur maximale réelle* ont été définies à 36°C et 40°C.

La normalisation dans l'axe X est faite en deux étapes. Le système calcule la *distance temporelle*, soit la *différence* en secondes entre l'horodatage de chaque observation avec l'horodatage de l'observation la plus ancienne dans la série chronologique à approximer. Ensuite la *distance temporelle* est divisée par $normalT$ exprimée elle aussi en secondes. Si $normalT$ n'est pas définie, la normalisation est faite avec $maxT$. L'algorithme de détection de tendances de TEMPAS estime que deux observations séquentielles sont distantes au minimum d'une « seconde ».

La normalisation dans l'axe Y est également faite en deux étapes. Le système calcule la *distance en valeur*, soit la différence entre la *valeur observée* et la *valeur minimale réelle observable* pour la *ressource*. Ensuite, il divise la *distance en valeur* par la différence entre la *valeur maximale réelle* et la *valeur minimale réelle* de la *ressource*.

Tableau 23. Normalisation des observations composant une série chronologique

Horodatage	Valeur
04/05/2013 23h15	38.5
04/05/2013 23h35	38.8
04/05/2013 23h45	39.1
05/05/2013 0h15	38.6
05/05/2013 0h27	39.2

⇨

Horodatage	Valeur
0	2.5
1200	2.8
1800	3.1
3600	2.6
4320	3.2

⇨

Horodatage	Valeur
0	0.625
0.28	0.7
0.47	0.775
0.83	0.65
1	0.8

5.3.1.3 Segmentation de la série chronologique

L'algorithme ascendant a deux étapes. La première étape est la *segmentation* et consiste à segmenter la série chronologique en $n-1$ segments. Pour chaque segment, le système calcule l'*indice de confiance de la tendance* (ICT) introduit dans le chapitre précédent. La deuxième étape est la *fusion* qui trouve et fusionne les deux segments consécutifs dont le *coût de fusion* est le « plus bas ». Le *coût de fusion* est la longueur du segment résultat de la fusion. Ainsi, le coût le « plus bas » est celui dont le segment résultant est le plus « court ». Le pseudocode de l'algorithme est présenté dans (13).

5.3.1.4 Déterminer la tendance

La tendance est la *valeur linguistique* assignée à un segment à partir de ses propriétés. TEMPAS utilise l'angle du segment pour déterminer la tendance associée au segment. L'angle dépend des unités utilisées pour représenter l'*horodatage* et de la *valeur observée* des observations approximées par le segment.

Exemple

Un segment qui représente un changement de 37.7°C à 38.2°C sur une période de 8 heures a une pente de 0.5/8, soit un angle de 3.57°.

Un segment représentant un changement de 99.86°F à 100.76°F sur une période de 28800 secondes a une pente de 0.9/28800, soit un angle de 0.00179°.

Toutefois, les deux segments ont le même comportement puisque 37.7°C = 99.86°F, 38.2°C = 100.76°F, et 8 heures = 28800 secondes, mais 3.57° est très différent de 0.00179°

```
01 /*
02 TS: une série chronologique (p1, p2,…, pn).
03 TS(i): le point i dans TS = pi.
04 TS[i:j]: une sous-série chronologique formée par
05            tous les points entre pi et pj.
06 P.t:l'horodatage du point p.
07 p.v:la valeur numérique représentée par le point p.
08 Segs:une approximation linéaire à morceaux de la série chronologique
09 Segs(i): le segment i dans Segs.
10 Seg.CI: l'indice de confiance de la tendance.
11 Seg.ps: le point du début du segment.
12 Seg.pe: le point du fin du segment.
13 vt: le temps de validité de la ressource.
14 et: le temps d'expiration de la ressource.
15 */
16
17 // retourne l'indice de confiance calculé à partir de deux points
18 // t1 <= t2
19 Algo double = calculCI(t1,t2)
20 if(t2-t1 < vt) return 1;
21 else if(vt < t2-t1 < et)
22     return (et-(t2-t1))/(et-vt);
23 else return 0;
24
25 // retourne un segment linéaire formée par deux points
26 Algo Seg = createSeg(p_s,p_e)
27 Seg.ps = p_s;
28 Seg.pe = p_e;
29 Seg.CI = calculCI(p_s.t,p_e.t);
30
31 // retourne la segmentation d'une série chronologique
32 Algo Segs = segmentation(TS)
33 for i = 1:1:TS.length
34     Segs=Segs.Add(createSeg(TS(i), TS(i+1));
35
36 // retourne le segment résultant de la fusion de deux segments
37 Algo Seg = mergeSeg(s1,s2)
38 s3.ps = s1.ps;
39 s3.pe = s2.pe;
40 s3.CI = s1.CI * s2.CI;
41
42 // retourne les k segments qui représentent la série chronologique
43 Algo Segs = findBestSegs(TS,k)
44 Segs = segmentation(TS);
45 while (Segs.length > k )
46 //trouve les deux segments consécutifs dont
47 //le coût de fusion est l'inférieur
48     (s1,s2) = findBestPair(Segs);
49     s3 = mergeSeg(s1,s2);
50     delete(Segs,s1);
51     delete(Segs,s2);
52     add(Segs,s3);
```

(13)

Après normalisation, la pente des deux segments précédents est 0.125, soit un angle de 7.125°. Ainsi, déterminer la tendance associée à un segment formé par des observations normalisées n'avantage pas la *distance temporelle* ou la *distance en valeur*. Les tendances sont donc indépendantes des unités des observations donc du domaine d'application.

5.3.2 Sélection des Observations

Le middleware a accès aux observations grâce à l'utilisation des vues. Lorsque la *condition d'activation* concerne l'*état* d'une ressource, le middleware transforme la dernière observation immédiatement antérieure à l'instant d'évaluation τ. Dans le cas des tendances, le middleware transforme toutes les observations qui accomplissent les critères $maxT$, $normalT$ et $normalP$ pour former la série chronologique sur laquelle les tendances sont détectées.

Le middleware utilise des *modèles de requêtes* pour construire des requêtes dynamiques qui récupèrent les observations des vues. Dans (14) un *modèle de requêtes* pour récupérer les observations et former une série chronologique est présenté. Au total, neuf *modèles de requêtes* ont été définis pour SQL Server.

```
SELECT TOP ?
DATEDIFF(SECOND,
        (SELECT MAX(horodatage) FROM ?
                        WHERE entity_id = ?
                        AND var_id = ?
                        AND horodatage <= ?),
        horodatage),
value,
horodatage
FROM ?
WHERE entity_id= ?
AND var_id = ?                                                    (14)
AND DATEADD(SECOND, ?, horodatage) >=
        (SELECT MAX(horodatage) FROM ?
                WHERE entity_id = ?
                AND var_id = ?
                AND horodatage <= ?)
AND horodatage <= ?
AND DATEADD(SECOND, ?, horodatage) >= ?
ORDER by horodatage DESC

[NormalP,NomVue,IdPat,IdVar,InstantEvaluation,NomVue,IdPat,IdVar,NormalT,NomV
ue,IdPat,IdVar,InstantEvaluation,InstantEvaluation,MaxT,InstantEvaluation]
```

5.3.3 Evaluation d'une situation d'alerte

TEMPAS est déployé comme un exécutable (~760 Ko). La détection d'alertes est déclenchée lors de l'appel de la fonction *evaluateAlertSituation* (15). Cela implique la création d'une base temporelle exploitable par le moteur d'alertes, la détection des tendances (si nécessaire), l'enrichissement de la base temporelle avec les tendances, la détermination le cas échéant d'une situation d'alerte et, si besoin, le stockage des alertes détectées dans la base de données du système d'information maître.

```
evaluateAlertSituation(TemporalDataBaseDirectory,
                AlertSituationId, EntityId, EvaluationTime)          (15)
```

5.3.3.1 Trouver toutes les solutions

Le moteur d'alerte a été développé en Prolog. L'avantage réside dans le raisonnement descendant de la formule (16) et le prédicat de base *findall* (qui fait partie de l'ISO standard). Le but de soumettre un *objectif* est de déterminer si la déclaration représentée par l'*objectif* est vraie à partir de la base de connaissances définie par des règles et des faits. Ainsi, prouver qu'un *objectif* est satisfait mène à prouver les prédicats qui le conditionnent. Si les prédicats peuvent s'inférer, l'*objectif* original est satisfait. Prolog s'appuie sur l'*hypothèse du monde clos*, où un prédicat est considéré comme faux s'il n'est pas possible d'inférer qu'il est vrai. Comme un fait peut être inféré par plusieurs règles, l'*objectif* peut être satisfait par différents chemins. Le prédicat *findall* permet de trouver toutes les solutions qui prouvent un *objectif*. C'est-à-dire que l'on peut déterminer qu'une déclaration est vraie grâce aux solutions trouvées.

Règles et faits	*Règles et faits*	*Règles et faits*		
A <- B (1)	*A <- B (1)*	*A <- B (1)*		
B <- C (2)	*B <- C (2)*	*A <- C (2)*		
C (3)		*B (3)*		
	objectif A	*C (4)*		
objectif A	*↓ par (1)*	*objectif A*	*objectif A*	**(16)**
↓ par (1)	*objectif B*	*↓ par (1)*	*↓ par (2)*	
objectif B	*↓ par (2)*	*objectif B*	*objectif C*	
↓ par (2)	*objectif C*	*↓ par (3)*	*↓ par (4)*	
objectif C	*↓*	*Vrai*	*Vrai*	
↓ par (3)	*Faux (impossible*			
Vrai	*d'inférer C)*			

De manière synthétique, l'inférence descendante utilisée par Prolog dans le but de détecter des alertes instances d'une situation d'alerte est : *prouver* les conditions d'activation — une condition est prouvée si certaine(s) valeur(s) linguistique(s) peu(ven)t être inférée(s) ; prouver les valeurs linguistiques — une valeur linguistique est prouvée lorsque il est possible d'inférer qu'une valeur numérique appartient à une plage.

D'autre part, Prolog est un langage utilisé dans l'un des produits logiciels de Calystene que nous réutilisons partiellement.

Un exemple de l'inférence descendante pour la détection d'alertes lors de l'évaluation d'une situation d'alerte définie comme *« poids = 'obèse' et tendance de la tension systolique ≤ 'stable' »* se trouve dans (17). La variable « poids » est dépendante de la variable « taille » et la variable « taille » est indépendante des autres variables. Le *moteur d'alertes* se sert du prédicat *findall* de Prolog pour trouver toutes les solutions qui satisfont la situation d'alerte. Chaque solution est une alerte détectée, laquelle est formée par la combinaison des valeurs linguistiques attribuées aux observations et

segments. Dans l'exemple, seul l'indice d'applicabilité de l'alerte (IAA) est associé aux alertes détectées.

Faits : *var(poids, 80), var(taille,170),* *tend(tension,[(17h10,129) – (19h17,90)],-30),* *tend(tension,[(17h10,129) – (18h34,125)],-5)*		
detecterAlertes([poids = 'obèse', tension ≤ 'stable'], Alertes)		
poids = 'obèse'		
var(taille,X), *var(poids,Y),* *0 ≤ X ≤ 150,* *60 ≤ Y ≤ 300*	*var(taille,X),* *var(poids,Y),* *140 ≤ X ≤ 175,* *69 ≤ Y ≤ 300*	*var(taille,X),* *var(poids,Y),* *168 ≤ X ≤ 230,* *78 ≤ Y ≤ 300*
Echec	*Succès* *X = 170,* *Y = 80*	*Succès* *X = 170,* *Y = 80*
tension ≤ 'stable'		
tension = 'stable'	*tension = 'décroissant'*	
tend(tension,T,-Z), *-10 ≤ Z ≤ 10*	*tend(tension,T,-Z),* *-90 ≤ Z ≤ -4*	
Succès *T = [(17h10,129) – (18h34,125)],* *Z = -5*	*Succès* *T = [(17h10,129) – (18h34,125)],* *Z = -5*	*Succès* *T = [(17h10,129) – (19h17,90)],* *Z = -30*
Alertes		
[[poids,80,'obèse'],[taille,170,'normale'], *[tension, [(17h10,129) – (18h34,125)],'stable'], IAA = 0.6]*		
[[poids,80,'obèse'],[taille,170,'grande'], *[tension, [(17h10,129) – (18h34,125)],'stable'], IAA = 0.43]*		
[[poids,80,'obèse'],[taille,170,'normale'], *[tension, [(17h10,129) – (18h34,125)], 'décroissant'], IAA = 0.27]*		
[[poids,80,'obèse'],[taille,170,'grande'], *[tension, [(17h10,129) – (18h34,125)], 'décroissant'], IAA = 0.1]*		
[[poids,80,'obèse'],[taille,170,'normale'], *[tension, [(17h10,129) – (19h17,90)], 'décroissant'], IAA = 0.69]*		
[[poids,80,'obèse'],[taille,170,'grande'], *[tension, [(17h10,129) – (19h17,90)], 'décroissant'], IAA = 0.51]*		

(17)

5.3.3.2 Observations manquantes

Le moteur d'alertes calcule l'indice linguistique révisé (ILR) qui exprime le degré d'exactitude de la valeur linguistique l_0 pour représenter la valeur numérique v_0 sous le contexte c étant donné l'existence d'autres valeurs linguistiques $l_1, \dots, l_{n+1}$ représentant les valeurs numériques $v_1, \dots, v_{n+1}$ issues des ressources $r_1, \dots, r_{n+1}$. La

modification des valeurs linguistiques est représentée à l'aide du graphe modificateur. Pour calculer l'ILR, il est nécessaire de trouver le chemin du graphe le plus important qui aboutit à la ressource r_0. Le chemin le plus important est déterminé par la somme la plus élevée des poids des arcs formant le chemin. Cependant, cette approche présente un problème dès que des observations manquent puisque le moteur échoue lors de l'inférence des valeurs linguistique l_0. Notre solution permet de trouver le chemin le plus important en tenant compte des valeurs manquantes.

Définition 37 : Chemin le plus important

Le chemin le plus important est celui dont la somme des poids des arcs formant le chemin dont les observations sont connues est la plus grande.

5.3.3.3 Persistance des alertes

Les alertes détectées sont stockées dans la base de données et accessibles au système maître qui applique la politique de notification. Toutefois, deux évaluations consécutives de la même situation d'alerte sur le même patient peuvent utiliser exactement les mêmes observations.

Exemple

Soit la surveillance d'un patient « dialysé » à domicile. Après la dialyse, l'infirmière libérale envoie le « poids » et la « tension » du patient au système d'information utilisé par le médecin traitant. Supposons une situation d'alerte qui évalue l'état du « poids » et de la « tension ». Si la situation d'alerte est évaluée à 18h et ensuite à 21h, des doublons sont susceptibles d'être générés.

Le middleware compare les observations et les segments qui forment l'*alerte détectée* sur un patient avec celles des *alertes détectées* précédemment instanciant la même situation d'alerte et évaluées sur le même patient. L'*alerte détectée* est stockée dans la base de données si :

- Il n'existe pas une alerte détectée formée par les mêmes observations et segments
- Il existe une alerte détectée formée par les mêmes observations et segments dont l'état est autre que « traitée »

5.4 Intégration de TEMPAS avec FSD

Nous avons présenté comment TEMPAS récupère et exploite les observations. Dans cette section, nous présentons le rôle du système d'information maître (FSD) qui intègre TEMPAS et fournit une nouvelle fonctionnalité à ses utilisateurs. L'intégration aborde quatre aspects :

- comment est réalisé le lien entre FSD et TEMPAS pour évaluer une situation d'alerte ;
- quand se produit l'évaluation des situations d'alerte;

- comment les utilisateurs définissent les situations d'alerte ;
- et quand la récupération d'alertes détectées doit avoir lieu.

5.4.1 Configuration du système d'alertes

La configuration du système commence par la création d'un fichier XML **(18)** qui contient des informations pour le déclenchement de l'évaluation des situations d'alerte. Dans le Tableau 24 sont représentés les attributs des éléments composant le fichier de configuration, les valeurs qu'ils peuvent prendre et la signification de ces valeurs.

```
<TEMPAS>
        <AlertDetector Activated="true" TemporalLaunch="300"
        DetectorPath="C:/web/AlertDetection/" DetectorName="FSDTempas"
        FileSeed="DetectAlerts" ODBC="FSD"                                  (18)
        User="admin" pwd="adminadmin" WorkDataBaseLocation="Context/"
/>
</TEMPAS>
```

Tableau 24. Fichier de configuration

Attribut	Valeurs	Signification
Activated	True/false	'True' pour activer l'évaluation des situations d'alerte
TemporalLaunch	Entier positif	Nombre de secondes à atteindre avant la prochaine évaluation périodique des situations d'alerte
DetectorPath	String	Chemin complet où se trouve déployé TEMPAS
DetectorName	String	Nom sous lequel a été déployé TEMPAS
FileSeed	String	Patron du nom des fichiers qui permettent de déclencher l'évaluation d'une situation d'alerte
ODBC	String	Connecteur ODBC utilisé par le middleware de TEMPAS afin d'interagir avec la base de données
User	String	Nom de l'utilisateur se connectant à la base de données
Pwd	String	Mot de passe de connexion à la base de données
WorkDataBase	String	Chemin relatif de stockage de la base de données temporelle

FSD et TEMPAS sont déployés sur le même serveur. FSD appelle la fonction d'évaluation d'une situation d'alerte en passant par les processus du système local qui *exécutent* TEMPAS. Les arguments pour l'évaluation de la situation d'alerte sont les valeurs des attributs du fichier de configuration (19). Les trois derniers arguments correspondent à *l'identificateur de la situation d'alerte* qu'il faut évaluer, l'*identifiant de l'entité* sur laquelle il faut évaluer la situation d'alerte et l'*instant d'évaluation* représenté par un horodatage. Pour évaluer une situation d'alerte sur tous les patients faisant partie du *groupe de patients* ciblé par la situation d'alerte, il faut remplacer l'*identifiant du patient* par *0*.

Exemple

Le système local appelle la fonction d'évaluation de la situation d'alerte identifiée par 2015 sur le patient identifié par 15290 le 3 janvier 2012 à 15h40

FSDTempas – 'C:/web/AlertDetection/' 'Context/' 'FSD' 'admin' 'adminadmin ***'2015' '15290' '03-01-2012 15:40:00'*** **(19)**

5.4.2 Politique d'évaluation

La politique d'évaluation définit quand l'évaluation des situations d'alerte doit avoir lieu. Nous différencions trois types d'évaluation, l'évaluation *instantanée*, *événementielle* et *périodique*. La politique d'évaluation au travers du *buffer d'évaluation* contrôle et priorise l'appel des évaluations des situations d'alerte.

5.4.2.1 Evaluation instantanée

Une *évaluation instantanée* est celle qui se passe à un instant précis donné par l'utilisateur. Dans FSD, ce type d'évaluation se produit en deux cas: chaque fois qu'une situation d'alerte vient d'être créée ou modifiée ou immédiatement suite à une évaluation déclenchée par un utilisateur. Dans le premier cas, l'instant d'évaluation est l'horodatage de création ou de modification de la situation d'alerte. Dans l'autre cas, il correspond à l'horodatage de l'instant présent.

5.4.2.2 Evaluation événementielle

L'*évaluation événementielle* se produit lorsque des événements prédéfinis dans le système maître sont détectés, le but étant que chaque fois qu'un *événement* est détecté, les situations d'alerte liées à l'événement s'évaluent. L'*arrivée* de nouvelles données dans le système maître considérées comme des observations dans le modèle de données de TEMPAS déclenche l'évaluation événementielle.

Exemple

L'arrivée d'une nouvelle température ou d'une nouvelle administration de médicament sont des événements qui déclenchent l'évaluation événementielle.

A différence de la vision globale de TEMPAS où l'utilisateur a le contrôle, l'évaluation événementielle est gérée par les concepteurs/développeurs du système d'information maître et non par les utilisateurs/concepteurs d'alertes.

Les événements du système maître sont détectés grâce à la *proxification* des services que nous entendons comme l'ajout des fonctionnalités qui n'interfèrent pas avec la logique-métier des services eux-mêmes. La *proxification* est réussie grâce au pattern de conception *fabrique* et au patron d'architecture *inversion de contrôle*. Chaque fois qu'un service est appelé, le patron *fabrique* crée une nouvelle instance du service et l'enregistre dans le *containeur injecteur de dépendances* qui permet d'appliquer l'*inversion de contrôle*. Le containeur retourne ainsi une instance du service proxifié.

Le développeur d'un service qui potentiellement manipule des données sources des alertes, appelle une méthode virtuelle *SubmitChanges* avant d'enregistrer les données dans la base de données (20). Le service qui implémente la méthode *SubmitChanges* est enregistré dans le containeur injecteur de dépendance (21). Le service proxifié se comporte différemment du service original lorsque la méthode *SubmitChanges* est appelée. Il enregistre les données dans la base de données comme le service original l'aurait fait et, seulement après, si la détection d'alertes est activée, un **événement TEMPAS** est déclenché *(TriggerTempasEvents)* (22).

```
public virtual void SubmitChanges(object dto,
                                  Type type,DomainOperation op)
{
        …
        Ctx.SaveChanges(…);
        …
}
```

(20)

```
<unity xmlns="http://schemas.microsoft.com/practices/2010/unity">
      <container>
        <extension type="Interception"/>
        <register type="Services.ISignesVitaux, Services"
        mapTo="Services.SignesVitaux, Services">
                <constructor>
                        <param name="userId"/>
                </constructor>
                <interceptor type="VirtualMethodInterceptor"/>
                <interceptionBehavior type="Tempas.Behaviors,Tempas"/
               >
        </register>
      </container>
</unity>
```

(21)

```
public IMethodReturn Invoke(IMethodInvocation method,
                        GetNextInterceptionBehaviorDelegate next)
{
     IMethodReturn result = next()(method, next);
     if(TempasActivated && method.MethodBase.Name == "SubmitChanges")
     {
          TriggerTempasEvents(method);
     }
     return result;
}
```

(22)

L'événement TEMPAS a associé deux informations de base, le type d'opération sur la base de données (*create, read, update, delete*) et le DTO (*Data Transfer Object*) utilisé pour transférer les données.

Les DTO transfèrent deux informations importantes pour **récupérer les situations d'alerte** à évaluer, l'*identifiant du patient*, et l'*identifiant de la variable*. La structure des DTO est dépendante des données à transférer, ainsi, les identifiants sont transmis sous des *propriétés.* Le fichier de configuration permet de récupérer ces

informations des différents DTO sous le nom d'*EntityIdProperty* et *VariableIdProperty* (23).

```
<TEMPAS>
  <MonitoredTypes>
    <Type FullName="DTO.SigneVitalDTO, Model"
    EntityIdProperty="PatientId"
    VariableIdProperty="ConstantId"/>
    <Type FullName="DTO.MedicamentDTO, Model"
    EntityIdProperty="IdPatient"
    VariableIdProperty="MedicamentId"/>
  </MonitoredTypes>
</ TEMPAS >
```

(23)

L'*identifiant du patient* est utilisé pour récupérer les situations d'alerte ciblant le patient identifié. Ensuite, l'*identifiant de la variable* est utilisé pour filtrer les situations d'alerte dont la variable fait partie des connaissances qui permettent de l'évaluer. La variable transférée par un DTO correspond à une *ressource observable* dans le modèle de TEMPAS. La correspondance est exprimée dans le modèle de données de TEMPAS dans la Table *vue_observations* qui contient les *vues* utilisées pour récupérer les observations d'une variable (Figure 48). Ainsi, ce sont les *identifiants des ressources* qui sont utilisés pour filtrer les situations d'alerte.

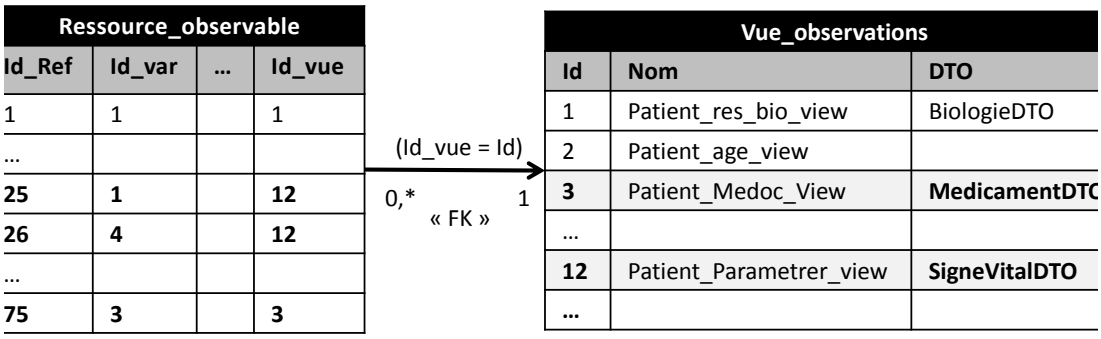


Figure 48. Correspondance entre variables et ressources

5.4.2.3 Evaluation périodique

Lors de la définition d'une situation d'alerte, les utilisateurs définissent l'*urgence* de la situation d'alerte et facultativement la *périodicité* souhaitée avec laquelle elle doit être évaluée. A la différence des évaluations événementielles, l'évaluation périodique d'une situation d'alerte évalue la situation d'alerte sur tous les patients faisant partie du groupe de patients ciblé par l'alerte. Le processus chargé de l'évaluation périodique récupère toutes les situations d'alerte dont la *dernière évaluation* est antérieure à l'*évaluation prévue* et les évalue puis il calcule quand la prochaine évaluation périodique aura lieu.

Le processus utilise les propriétés *PeriodicEvaluation* et *LastEvaluation* associées aux *situations d'alerte* pour récupérer les situations d'alerte qui doivent être évaluées à un instant τ. *PeriodicEvaluation* exprime la périodicité souhaitée avec laquelle la

situation d'alerte doit être évaluée. Elle peut être modifiée par les utilisateurs. *LastEvaluation* exprime quand la situation d'alerte a été évaluée sur tout le groupe de patients. Elle est mise à jour par le système chaque fois que l'*évaluation périodique* ou l'*évaluation instantanée* ont lieu. Le processus utilise *la propriété Urgence* pour avantager l'évaluation des situations d'alerte qui représentent un intérêt majeur pour les utilisateurs.

Exemple

Nous avons défini quatre *urgences* « prioritaire », « haute », « normale », et « basse » associés aux valeurs 1, 2, 3 et 4.

Le calcul du temps d'attente avant la prochaine évaluation périodique collective prend en compte :

- la déviation standard de *PeriodicEvaluation* des situations d'alerte,
- la moyenne,
- la *PeriodicEvaluation* la plus petite parmi toutes les situations d'alerte avec une *urgence* majeure,
- et le nombre de secondes attribué à *TemporalLaunch* du fichier de configuration dans le Tableau 24.

Le pseudocode du calcul de la prochaine évaluation est présenté dans (24). Deux conditions priment pour ce calcul : le temps d'attente avant la prochaine évaluation collective ne peut pas être inférieur au temps défini dans la configuration du système d'alertes *TemporalLaunch* ; une situation d'alerte avec une urgence de 1 (« prioritaire ») ne peut pas dépasser deux fois la durée de la *PeriodicEvaluation* sans être évaluée.

Une première comparaison utilise la déviation standard proportionnelle à la dispersion des *PeriodicEvaluation*. Si les *PeriodicEvaluation* ne sont pas très dispersées, le *temps d'attente* est le minimum entre la *moyenne* et la *PeriodicEvaluation* la plus petite parmi les situations d'alerte les plus urgentes. Le Tableau 25 montre le résultat du calcul du temps d'attente en présence de quatre situations d'alerte. Une urgence égale à 1 exprime une urgence « prioritaire ».

```
Algo double = tempsAttendre()
// Récupère une liste avec la durée de PeriodicEvaluation et la
// respective urgence des situations d'alerte,
// [[p1,u1],…,[pi,ui],…[pn,un]]
P = getPeriodicEvaluation();

// Calcule la moyenne de la premier colonne, soit PeriodicEvaluation
U = moyenne(P(1)) ;

// Calcule la déviation standard
O = stdev(P(1)) ;

// Récupère les éléments avec l'urgence la plus élevée
//[[pa,1],…,[pj,1],…[pm,1]]
I = PlusUrgents(P) ;

// Récupère la valeur minimal de la premier colonne
M = min(I(1)) ;

// Récupère le temps minimal à attendre pour deux évaluations
// consécutives
L = getTemporalLaunch() ;

//Temps à attendre
T = 0 ;

//Données très disperses
If O > M
      T = 2*M
Else
      If M <= U
            T = M
      Else
            T = U

If T < L
      T = L

Return T
```

(24)

Tableau 25. Exemples de calcul du temps d'attente

Situation d'alerte	EXEMPLE 1		EXEMPLE 2		EXEMPLE 3	
	Pér.	Urg.	Pér.	Urg.	Pér.	Urg.
A	2000	3	400	3	300	3
B	500	2	350	2	350	2
C	600	1	320	1	450	1
D	1800	1	600	1	600	1
Moyenne	1225		417,5		425	
Déviation standard	679,61		109,17		114,56	
Min(plus urgent)	600		320		450	
TemporalLaunch	300		300		300	
Temps d'attente	**1200**		**320**		**425**	

5.4.2.4 Buffer d'évaluation

Le buffer d'évaluation est un mécanisme qui contrôle l'appel des évaluations des situations d'alerte. L'évaluation instantanée, l'évaluation événementielle, et l'évaluation périodique passent par le *buffer d'évaluation* qui *réduit* les évaluations répétitives.

Le buffer contient toutes les situations d'alerte qui doivent être évaluées dans la *liste de situations d'alerte à évaluer*. L'insertion d'une nouvelle évaluation peut se faire au milieu de la *liste* selon l'*urgence* de la situation d'alerte et le temps d'attente des situations devant être évaluées. En règle générale, « plus urgente est une situation d'alerte, plus tôt elle doit être évaluée ». Cependant, « l'arrivée constante des nouvelles évaluations des situations d'alerte d'urgence 1 ne doit pas retarder indéfiniment l'évaluation d'une situation d'alerte qui n'est pas urgente ».

La *fenêtre d'évaluation* est un troisième critère, en plus de l'*urgence* de la situation d'alerte et du *critère temporel*, qui exprime où, dans la *liste de situations d'alerte à évaluer*, il n'est pas possible d'insérer de nouvelles évaluations. Elle correspond aux premiers éléments de la liste d'évaluation. L'urgence des situations d'une situation d'alerte est définie par les utilisateurs. En revanche, la taille de la fenêtre et le critère temporel (t) sont récupérés depuis le fichier de configuration (25).

```
<TEMPAS>
        <Buffer WindowSize="20" TimeCriteria="5"/>
</TEMPAS>
```
(25)

La fenêtre est réduite au fur et à mesure de l'évaluation des situations d'alerte. Une fois sa taille minimale atteinte, elle est recalculée. La Figure 49 montre ce comportement.

Pour l'insertion d'une nouvelle situation d'alerte, le buffer parcourt une à une les situations d'alerte à évaluer *au-delà* de la fenêtre. L'insertion est possible si la situation arrivante est plus urgente que la situation en attente et que cette attente est courte. Le temps d'attente est court si la différence temporelle entre l'instant d'insertion dans la *liste d'évaluation* de la situation en attente et la situation à insérer est inférieure à la *valeur de comparaison d'attente courte* du Tableau 26.

La *valeur de comparaison d'attente courte* est dépendante de l'*urgence* des situations d'alerte et du critère temporel t obtenu du fichier de configuration (25). Une urgence de 1 exprime que la situation d'alerte est très urgente, c'est-à-dire prioritaire.

Tableau 26. Attente courte

Urgence		Arrivant						
		1	**2**	**…**	$\boldsymbol{i}$	**…**	$\boldsymbol{n-1}$	$\boldsymbol{n}$
En attente	**1**		t	…	$(i-1)*t$	…	$(n-2)*t$	$(n-1)*t$
	2			…	$(i-2)*t$	…	$(n-3)*t$	$(n-2)*t$
	..				…	…	…	…
	$\boldsymbol{i}$					…	$\big((n-1)-i\big)*t$	$(n-i)*t$
	…						…	…
	$\boldsymbol{n-1}$							t
	$\boldsymbol{n}$							

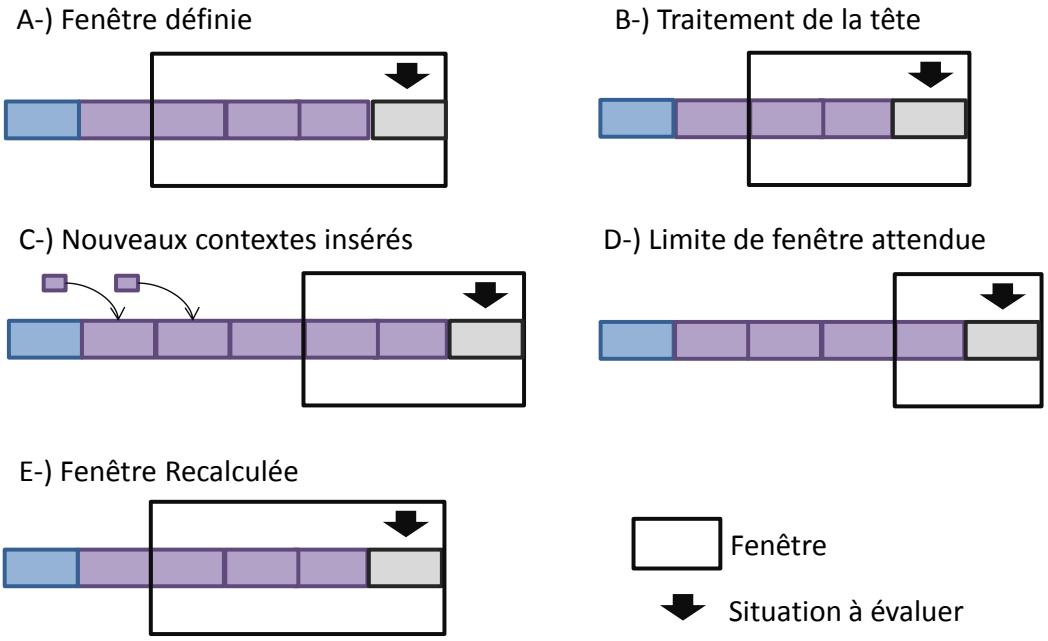


Figure 49. Fenêtre d'évaluation

5.4.3 Exploitation d'informations qui ne sont pas générées par le système maître

Rappelons que les situations d'alerte sont définies en utilisant des conditions d'activation construites avec des ressources observables. Les ressources observables sont récupérées du système d'information maître. Cependant, les ressources observables disponibles peuvent ne pas permettre de définir complétement les situations d'alerte qui intéressent les utilisateurs. L'amélioration que nous proposons utilise des informations issues d'autres sources afin de les présenter comme des *ressources observables* lors de la définition des situations d'alerte.

Nous utilisons des *services web* afin de remplir la base de données avec des informations externes comme par exemple les « données météorologiques » pour

repérer un contexte caniculaire ou d'hydrométrie anormale. Ces informations peuvent être assurées par différents services classés localement par un *consommateur de services* par *qualité de service*. Le *consommateur* essaie de récupérer les informations en faisant appel au service web avec la plus haute qualité. Si le service est défaillant, le « consommateur de services météorologiques » utilise le prochain service dans la liste de « services météorologiques ». Un *orchestrateur* récupère les informations des *consommateurs de services* afin de les stocker dans la base de données pour les rendre exploitables par TEMPAS. Une table contient les variables météorologiques récupérées par le *consommateur de services*, et une autre contient les *valeurs mesurées*. Une *vue* fait le lien entre tous les patients concernés par les valeurs des variables météorologiques afin de les exploiter comme des *observations* lors de l'évaluation des situations d'alerte. Les variables météorologiques ont été ajoutées à la table Ressources Observables responsable de la séparation de modèles.

Exemple

La « température extérieure » utilisée pour définir des situations d'alerte correspond à Grenoble, mais l'observation est attachée à un patient et non la ville. La vue qui permet à TEMPAS d'exploiter les observations de la « température extérieure » contient la température extérieure de la ville où se trouve le patient lors de la consultation du service météorologique.

‹ « Dupont », « Température extérieure », « 12/01/2014 17h », 10°› est une observation

Le concepteur/développeur du système d'information maître est responsable de rapatrier les informations qui peuvent être utiles aux utilisateurs lors de la définition des situations d'alerte. Si cela se fait à l'aide de services web, il est nécessaire de :

- trouver les services web ;
- créer les objets pour récupérer les informations à partir de WSDL ;
- définir une interface pour le consommateur de services ;
- créer le consommateur de services et y enregistrer les services connus par ordre de *qualité de service* ;
- enregistrer le consommateur de services dans l'orchestrateur d'informations externes ;
- définir la couche de données correspondante pour stocker et associer les informations aux patients (les alertes sont détectées sur des patients) ;
- et finalement ajouter les variables dans la table de ressources observables.

5.4.4 Alertes détectées

Des alertes sont éventuellement détectées lors de l'évaluation des situations d'alerte. Dans cette section, nous détaillons les processus de notification des alertes aux utilisateurs ainsi que son traitement.

5.4.4.1 Buffer de notification

La récupération des alertes est un processus qui permet aux applications clientes de récupérer les alertes détectées lors des évaluations des situations d'alerte. La

récupération est *périodique* et *paramétrable* dans l'application cliente par les utilisateurs. Afin d'appliquer la *politique de notification* de TEMPAS et réduire les lectures dans la base de données, nous avons implémenté un *buffer de notification*. Le buffer récupère périodiquement les alertes détectées de la base de données, applique la politique de notification de TEMPAS, et rend les alertes *accessibles* aux utilisateurs. Le processus est illustré dans la Figure 50. La fréquence de récupération est définie à *t*. Les flèches indiquent la récupération des situations d'alerte.

Le *buffer de notification* récupère toutes les alertes dont l'état est « nouvelle » et toutes les alertes détectées dans les « trois derniers jours ». Les alertes « nouvelles » sont celles qui doivent être traitées par un utilisateur tandis que les alertes des trois derniers jours offrent un historique des alertes détectées sur les patients.

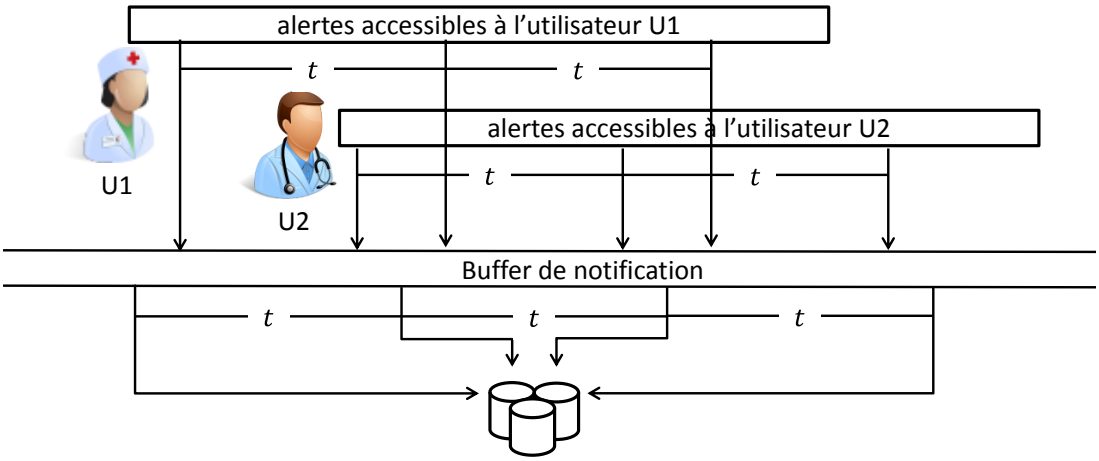


Figure 50. Récupération des alertes détectées

L'utilisateur peut accéder uniquement aux alertes détectées résultat de l'évaluation des situations d'alerte dont le groupe de destinataires comprend l'utilisateur, et qui ont été détectées sur un patient auquel l'utilisateur a le droit d'accès.

5.4.4.2 Politique de notification de TEMPAS

La politique de notification a été présentée dans la section 4.6. Avant de l'appliquer, le système change l'état « nouvelle » des alertes détectées qui ont dépassé leur temps de validité par l'état « périmée ».

Les alertes détectées lors de l'évaluation d'une situation d'alerte son filtrées par qualité, ensuite par expressivité, et agrégées dans une seule alerte *accessible*. Les alertes accessibles sont comparées avec les alertes accessibles issues des détections précédentes afin de conclure si la notification doit être *explicite*. La notification explicite est de la responsabilité de l'application cliente.

5.4.4.3 Groupe de destinataires

Pendant la configuration d'une situation d'alerte, l'utilisateur peut spécifier quels utilisateurs doivent être notifiés des alertes détectées. Dans FSD, les critères pour la création des groupes sont le *rôle* de l'utilisateur, le *service* hospitalier auquel l'utilisateur est attaché, et des utilisateurs spécifiques.

Puisque le nombre de groupes d'utilisateurs est potentiellement plus petit que le nombre d'alertes détectées, le *buffer de notification* récupère d'abord les groupes auxquels l'utilisateur fait partie et retourne les alertes dont le groupe de destinataires de la situation d'alerte coïncide, mais seulement après avoir vérifié *le droit d'accès*. S'il n'a pas le droit d'accès au dossier du patient concerné par l'alerte, la notification ne lui parvient pas.

En revanche, la récupération d'alertes détectées à notifier à un utilisateur ne prend pas en compte le rôle avec lequel l'utilisateur s'est connecté à l'application cliente ni le service hospitalier où il se trouve.

5.4.4.4 Traitement des alertes

Les alertes notifiées aux utilisateurs dont l'état est « nouvelle » peuvent être traitées par les utilisateurs. *Traiter* une alerte accessible change l'état à « traitée » de « toutes les *alertes détectées* issues de la même détection qui ont été utilisées pour former l'*alerte accessible* ». Le changement d'état affecte les alertes notifiées aux autres utilisateurs faisant partie du même groupe de destinataires.

Le changement d'état est une mise à jour faite par l'application cliente suite à une action faite par « un » utilisateur. Si le changement a été pris en compte, le buffer de notification rendra accessible l'alerte avec l'état « traitée ». Il se peut que l'état de l'alerte indiqué par le *buffer de notification* ne soit pas le même que l'état changé par l'utilisateur. Si c'est le cas après deux récupérations, la mise à jour a *échouée*. Dans ce cas, l'application cliente doit afficher l'état récupéré et non l'état changé pour que l'utilisateur puisse se rendre compte de l'échec et si nécessaire faire une nouvelle tentative.

Le Tableau 27 illustre le changement réussi (en haut) et échoué (en bas) de l'état d'une alerte. La fréquence de récupération est définie à 5 minutes. La réussite ou l'échec de la mise à jour se joue à la minute 10 où le buffer de notification a eu le temps de récupérer le nouvel état de l'alerte.

Tableau 27. Changement d'état d'une alerte

Minute	Déclencheur	Action	Etat dans App. cliente	Etat dans la BD
0	Buffer de notification	Lire la BD	-	nouvelle
3	Application Cliente	Lire le buffer	-	nouvelle
5	Buffer de notification	Lire la BD	-	nouvelle
6	Application Cliente	Changer état	traitée	
8	Application Cliente	Lire le buffer	traitée	nouvelle
10	Buffer de notification	Lire la BD	-	traitée
13	Application Cliente	Lire le buffer	traitée	traitée

Minute	Déclencheur	Action	Etat dans App. cliente	Etat dans la BD
0	Buffer de notification	Lire la BD	-	nouvelle
3	Application Cliente	Lire le buffer	-	nouvelle
5	Buffer de notification	Lire la BD	-	nouvelle
6	Application Cliente	Changer état	traitée	
8	Application Cliente	Lire le buffer	traitée	nouvelle
10	Buffer de notification	Lire la BD	-	nouvelle
13	Application Cliente	Lire le buffer	nouvelle	nouvelle

5.4.5 Application Cliente

L'application cliente est utilisée par les utilisateurs du système d'alertes afin de configurer les situations d'alerte et visualiser les alertes détectées. FSD est un système d'information de santé qui gère plusieurs domaines fonctionnels. Chaque domaine fonctionnel est présenté aux utilisateurs par des *composants fonctionnels*. L'utilisateur a un espace de travail où il *dépose* les composants dont il a besoin. Les composants et l'espace de travail communiquent entre eux grâce aux messages échangés avec le pattern *publish/subscribe*. Chaque composant fait appel aux services afin de gérer une partie de la logique-métier du domaine fonctionnel. Sept composants et un démon ont été développés dans le but de faciliter la *définition* des situations d'alerte et la *gestion des alertes accessibles*. Chaque composant possède deux vues : une vue consultation et une vue édition.

5.4.5.1 Gestion des variables

Le composant « configuration de variables » permet aux utilisateurs de préconfigurer les *valeurs linguistiques* qui expriment l'état ou la tendance d'une ressource observable. Une même ressource peut être préconfigurée plusieurs fois avec différentes valeurs linguistiques afin de mieux définir le contexte de l'alerte.

Exemple

Les *valeurs linguistiques* pour représenter l'état de l'« âge » peuvent être « enfant », « jeune », « adulte », « âgé » ou encore « nouveau-né », « nourrisson », « bébé ».

La Figure 51 montre la vue consultation des valeurs linguistiques du pouls selon l'âge du bébé. Dans ce composant l'utilisateur définit les valeurs minimale et maximale mesurables pour la ressource observable, les valeurs linguistiques à utiliser, le temps de validité, et le temps d'expiration.

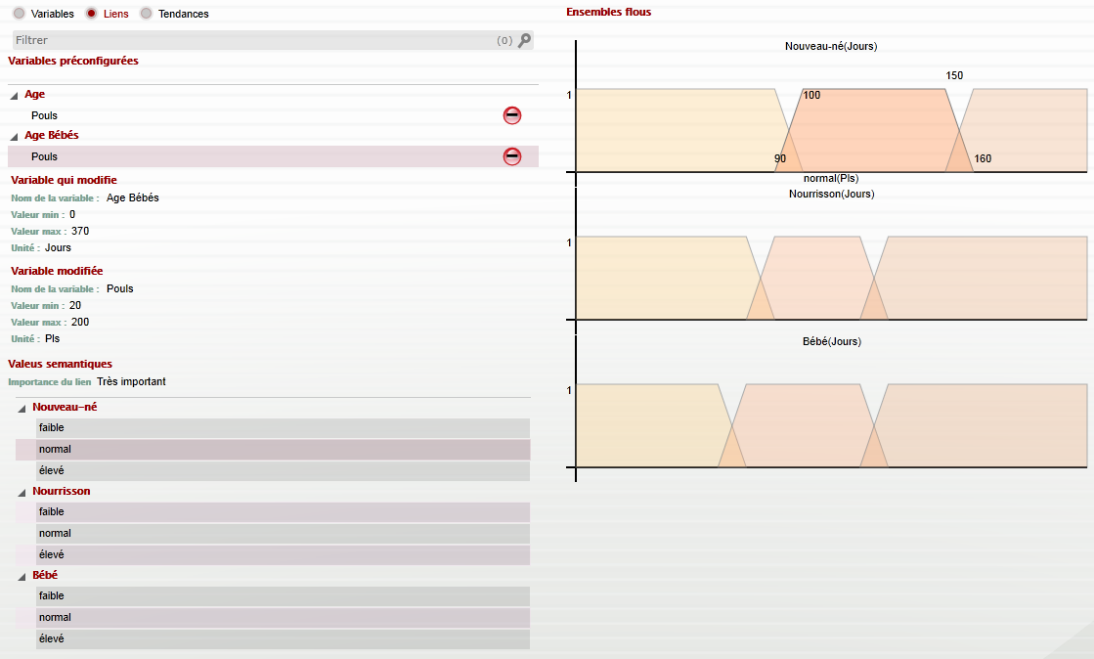


Figure 51. Composant : gestion des variables

5.4.5.2 Groupe de patients

Le composant « groupe de patients » permet de regrouper des patients selon deux critères, les patients attachés à un service hospitalier, et les patients suivis par un médecin. Le service est une des parties de la division *logique* de l'établissement. Si le patient se trouve dans l'hôpital ou chez lui, il peut être attaché au même service. La Figure 52 montre la vue consultation du composant. L'utilisation explicite de groupes de patients réduit l'évaluation inutile des situations d'alerte.

5.4.5.3 Groupe de destinataires

Le composant « groupe de destinataires » permet de créer des groupes d'utilisateurs soit en spécifiant les utilisateurs et leur rôle ou la structure logique à laquelle les utilisateurs sont attachés. La structure logique est une hiérarchie à quatre niveaux commençant par la société, les pôles, les services, et les unités fonctionnelles. La Figure 53 montre le vue édition du composant. Le groupe de destinataires a pour objectif d'envoyer les alertes détectées seulement aux utilisateurs qui sont concernés.

Exemple

Le groupe est formé par les utilisateurs attachés à l'unité fonctionnelle de la « maternité », par ceux avec le rôle de « surveillante », et par l'utilisateur « Medim Andrée ».

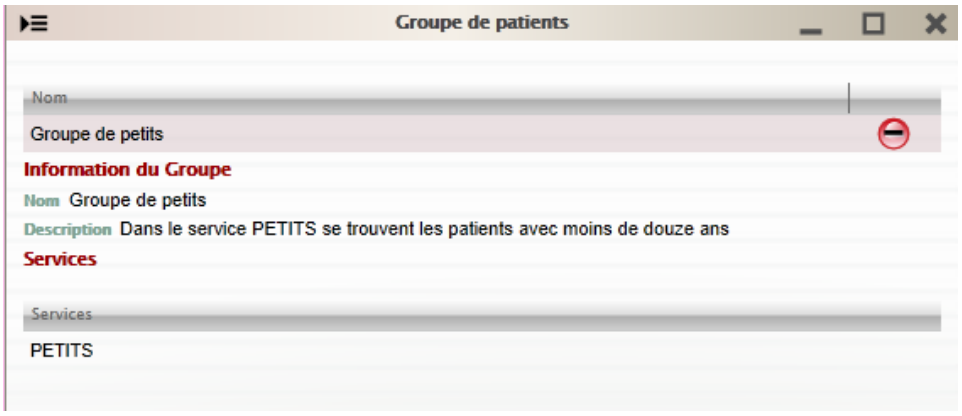


Figure 52. Composant : groupe de patients

Figure 53. Composant : groupe de destinataires

5.4.5.4 Equation de l'alerte

Le composant « équation d'alerte » sert à *instancier* les ressources observables préconfigurées dont les valeurs linguistiques expriment l'état de la ressource. Sa fonction la plus importante est de définir les *graphes modificateurs* pour le calcul des valeurs linguistiques. La Figure 54 montre la vue édition du composant « équation d'alerte ». Le composant permet de :

- regrouper les ressources observables à utiliser pour la définition d'une situation d'alerte ;
- ajouter des arcs pour le calcul des valeurs linguistiques ;
- et redéfinir les fonctions d'appartenance représentant les ensembles flous associés aux valeurs linguistiques.

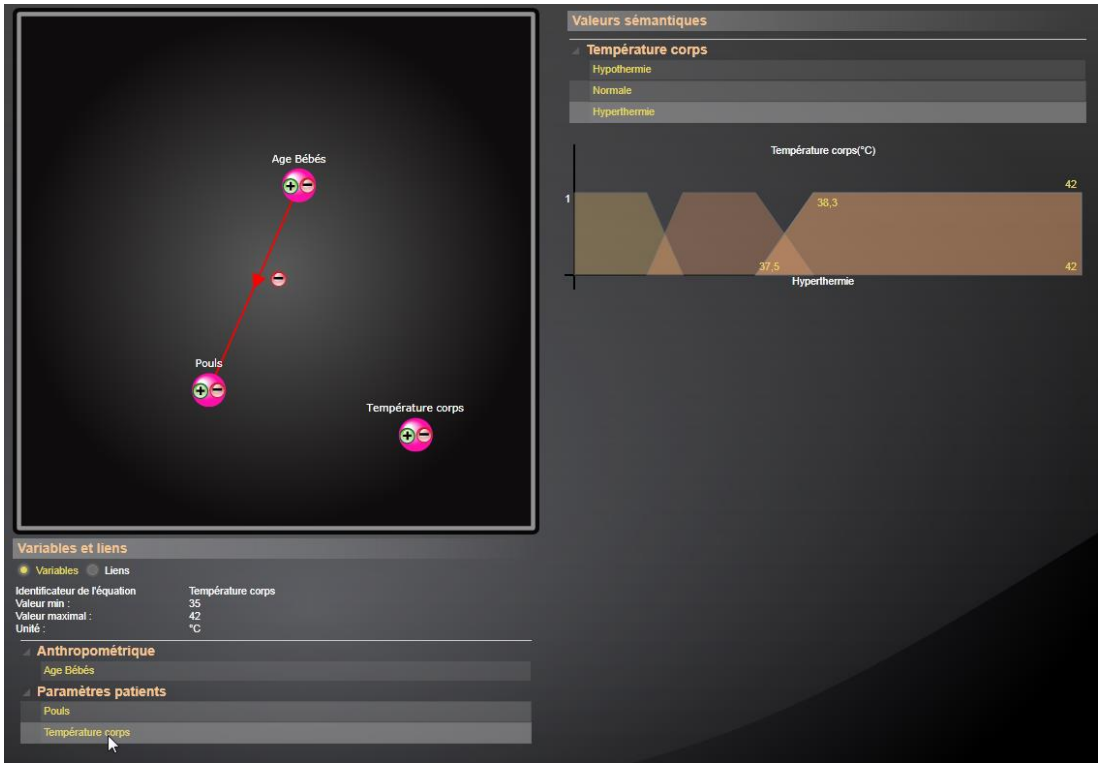


Figure 54. Composant : équation de l'alerte

5.4.5.5 Tendance

L'instanciation des ressources observables dont les *valeurs linguistiques* expriment la tendance est réalisée par le composant « Tendance ». Le composant permet de définir les fonctions d'appartenance représentant les ensembles flous associés aux valeurs linguistiques. Dans le cas des tendances, les fonctions d'appartenance sont définies entre « moins quatre-vingt-dix » et « quatre-vingt-dix degrés » puisque la *valeur linguistique* qui exprime la tendance est calculée par rapport à l'angle du segment qui approxime les observations.

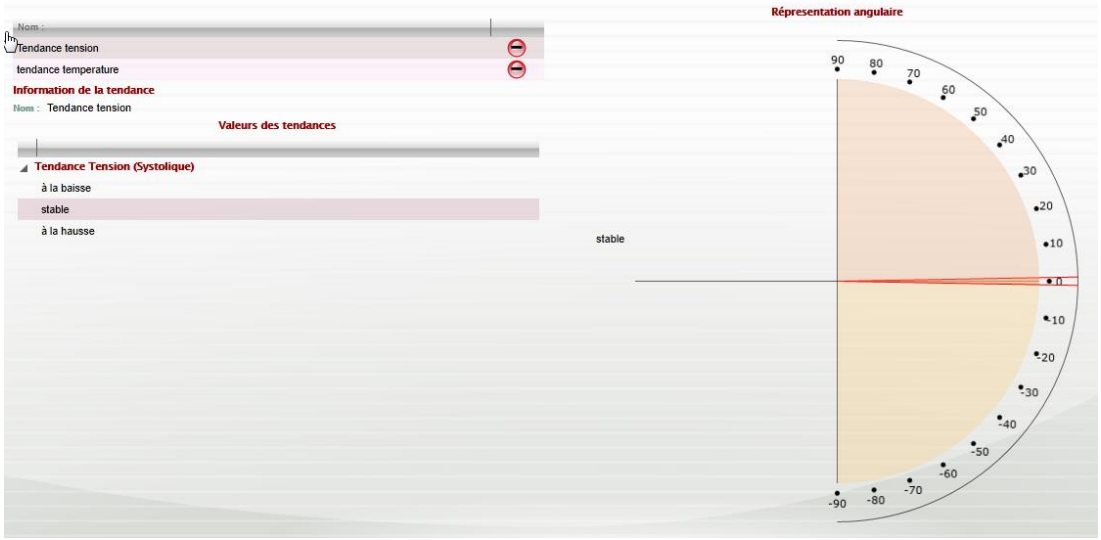


Figure 55. Composant : tendances

5.4.5.6 Alerte

Le composant « alerte » est le cœur de l'application cliente dont la fonction est la contextualisation de la *situation d'alerte*.

Chaque condition d'activation a un poids exprimé dans une valeur entre 1 et 10 où 10 est très important. Le poids est normalisé de sorte que la somme des poids est égale à 1. Dans le cas des tendances, il est nécessaire de définir la *période* dans laquelle les tendances doivent être détectées. Le nombre d'observations à utiliser peut être limité. Finalement, l'utilisateur définit le nombre de segments à détecter. Chaque segment détecté est utilisé pour la vérification de la condition d'activation.

Exemple

Utiliser toutes les mesures (observations) des dernières 41 heures pour calculer la tendance de la tension systolique.

Les *conditions d'activation temporelles* sont illustrées dans la Figure 56. Les conditions temporelles permettent de définir qu'un segment ou une observation, commence avant ou en même temps qu'un(e) autre(e).

La Figure 57 montre la définition des *conditions d'activation intrinsèques* dans le composant « alerte ».

Ce événement est	≤	que ce cet autre	Imp.
Tendance Température corps	≤	Tendance Tendance Tension (Systolique)	1

Figure 56. Conditions temporelles

Figure 57. Conditions intrinsèques

L'utilisateur peut définir des conditions d'activation avec des valeurs numériques. Dans ce cas, le système crée lui-même les *valeurs linguistiques*, les *ensembles flous* associés, et les *fonctions d'appartenance* correspondantes. Ceci est invisible aux yeux des utilisateurs. La Figure 58 montre ce cas. Dans la fenêtre du haut, la condition a été définie comme *« pouls ≤ 40 »*, dans celle du bas, *« pouls ≥ 120 »*. Les opérateurs relationnels courants dans ces conditions de base sont « ≤ » et « ≥ ».

L'avantage de créer automatiquement les fonctions d'appartenance est l'exploitation des indices de qualité lors de la détection d'alertes. Toutefois, l'utilisateur peut ne pas vouloir utiliser la logique floue pour la détection d'alertes. Dans ce cas, il peut définir des conditions de façon « habituelle » sous forme de seuils numériques et fixer *l'indice d'applicabilité* à 1. Les seuils de filtrage sont ajustés à l'aide des curseurs (Figure 59).

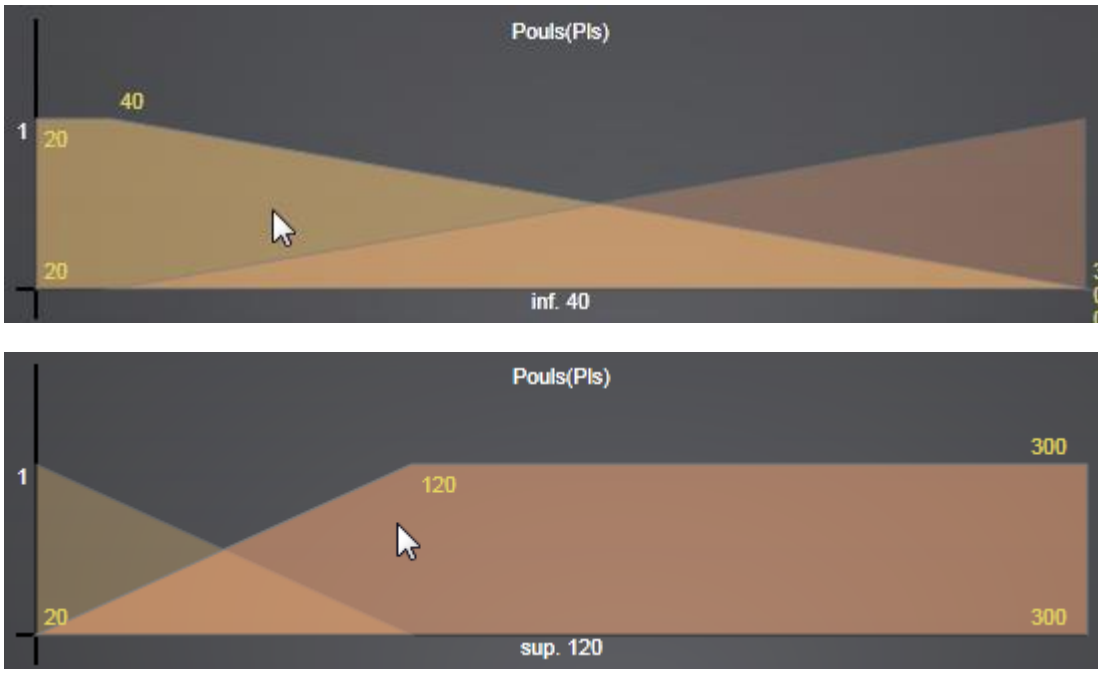


Figure 58. Conditions avec des valeurs numériques

Figure 59. Seuils de filtrage

Pendant la configuration de la situation d'alerte, les utilisateurs définissent le *groupe de patients* ciblés par l'évaluation de la situation d'alerte ainsi que le *groupe d'utilisateurs* à notifier lors de la détection d'alertes. Par défaut, une situation d'alerte est évaluée sur tous les patients et notifiée à l'utilisateur qui l'a créée.

Dans le paramétrage de la situation d'alerte, les utilisateurs assignent une *urgence* à la situation d'alerte, définissent si elle est ou non *active* et la *périodicité* d'évaluation dans le cas où l'évaluation est périodique. La Figure 60 montre la configuration et le paramétrage de la situation d'alerte. Nous avons défini quatre urgences « prioritaire », « haute », « normale », et « basse ». L'urgence sert à prioriser l'*évaluation* des situations d'alerte (politique d'évaluation).

Figure 60. Configuration et paramétrage de la situation d'alerte

Finalement, les utilisateurs définissent le cycle de vie de l'alerte présenté dans la politique de notification. La Figure 61 illustre le cycle de vie par défaut qui s'applique sur toutes les nouvelles situations d'alerte dans FSD.

5.4.5.7 Liste d'alertes

La fonctionnalité du composant *liste d'alertes* est de lister toutes les alertes qui sont *accessibles* à l'utilisateur connecté. Les alertes sont ordonnées par le temps restant avant de devenir périmées et groupées par le patient concerné. Par défaut, le composant affiche les alertes dont l'état est « nouvelle ». Toutefois, l'utilisateur peut visualiser rapidement les alertes selon leur état sur un des quarts de la rondelle (Figure 62). Un champ textuel *filtre* les alertes affichées par :

- état,
- nom ou prénom du patient,
- nom de l'alerte ou message à afficher,
- les conditions ayant activé l'alerte,
- la date de détection,
- et les alertes détectées l'« heure », le « jour », ou la « semaine » en cours.

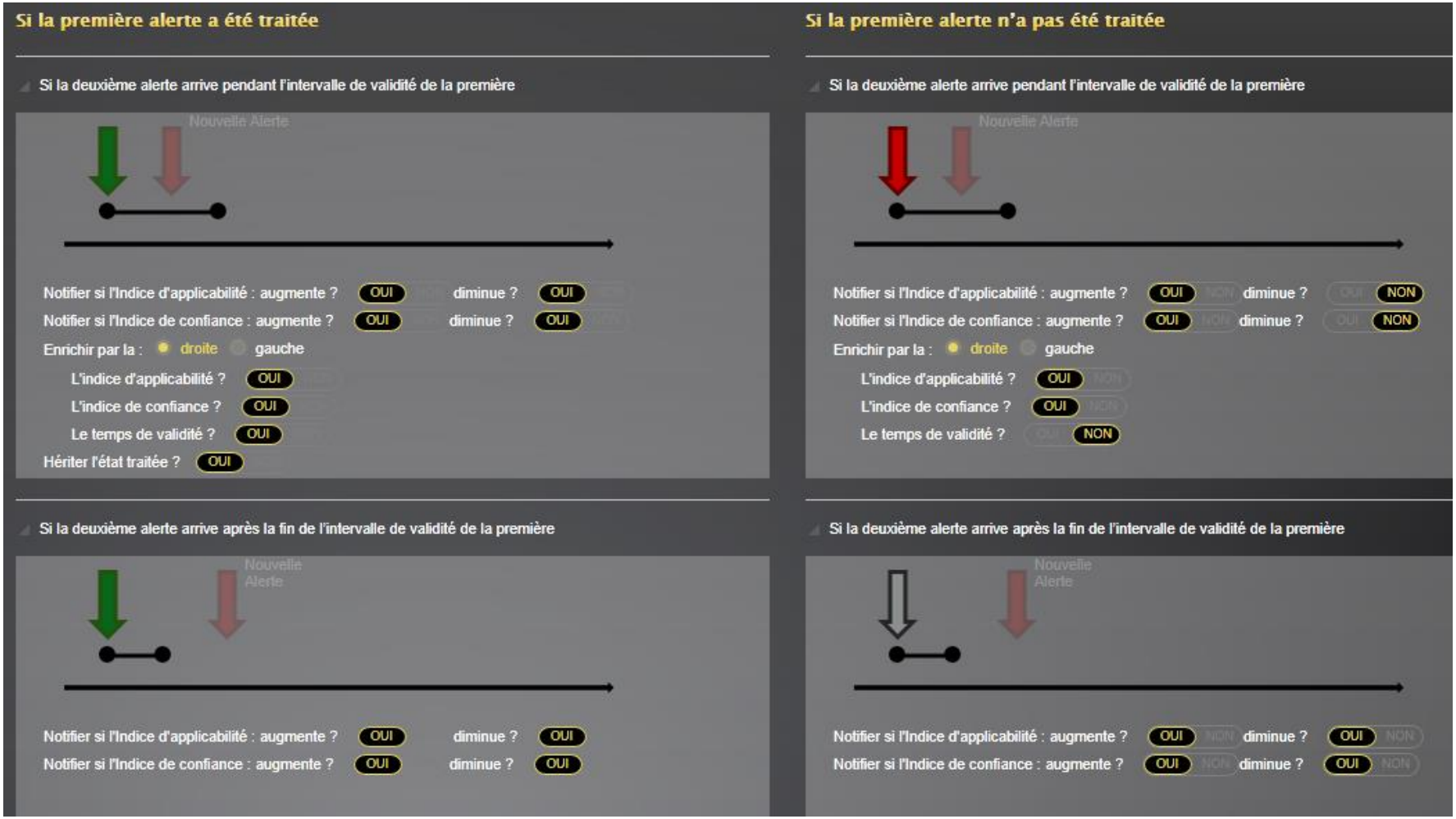


Figure 61. Cycle de vie de l'alerte

Deux curseurs sont utilisés pour filtrer les alertes selon le *ratio d'applicabilité* ou de confiance. Ce filtrage priorise les alertes qui dépassent « plus largement » la *qualité minimale exigée* de la politique de notification d'une situation d'alerte. La formule du calcul du ratio est présentée dans la formule (26) où $\propto$ représente la qualité minimale exigée, soit pour l'applicabilité soit pour la confiance, pour la situation d'alerte. Le filtrage par ratio s'applique sur toutes les alertes accessibles sauf celles dont l'urgence de la situation d'alerte correspondante est *prioritaire*.

Figure 62. Composant : liste d'alertes

$$ratio = \begin{cases} \frac{IAA - \propto}{1 - \propto} & \propto \neq 1 \\ 1 & \propto = 1 \end{cases} \quad \textbf{(26)}$$

Un état « vue » a été introduit dans l'application cliente afin de réduire la charge visuelle de la liste d'alertes. Une alerte avec un état « vue » est une alerte « nouvelle » que ne s'affiche pas par défaut. Cela permet aux utilisateurs de prioriser la visualisation des alertes « nouvelles ». Néanmoins, une alerte « vue » reste une alerte « nouvelle » qui se périme si elle n'est pas « traitée ».

En plus de lister les alertes accessibles pour l'utilisateur connecté, ce composant autorise l'utilisateur à changer l'état d'une alerte. Une alerte est « traitée » seulement si un utilisateur a changé son état. Le changement d'état peut se faire alerte par alerte, ou en sélectionnant plusieurs alertes en même temps.

5.4.5.8 Démon de récupération d'alertes

Le daemon récupère les alertes accessibles à l'utilisateur connecté *périodiquement*. Deux paramétrages sont disponibles : activer ou désactiver le démon pour récupérer ou arrêter la récupération d'alertes, et ajuster la périodicité.

Le démon utilise un service qui récupère les alertes du *buffer de notification*. Les alertes accessibles ont une propriété *NotificationExplicite* pour exprimer si la notification doit être explicite ou seulement implicite après application de la politique

de notification. La notification explicite est faite dans l'espace de travail de l'utilisateur. Elle se manifeste comme un *volet couleur magenta* qui couvre tout l'écran et se positionne au-dessus des composants en clignotant moins de deux secondes pendant lesquels l'activité courante de l'utilisateur est bloquée. La notification implicite se manifeste principalement avec le composant *liste d'alertes*. Cependant, d'autres composants peuvent souscrire à la notification implicite.

Exemple

Le composant *plan de service* illustre la division physique de l'établissement de santé. Il contient les « lits » libres et occupés. Un lit « occupé » est assigné à un patient. Si une alerte a été détectée sur le patient qui occupe le lit affiché, une « icône d'alerte clignotante » est dessinée à côté du lit.

5.5 Synthèse

Nous avons présenté l'intégration de TEMPAS à Futura Smart Design. La première étape consiste à rendre les informations de FSD exploitables par le moteur d'alertes de TEMPAS. Un modèle de données ainsi qu'un middleware pour la lecture, l'écriture et la transformation des données ont été conçus à ce propos.

TEMPAS est formé par le middleware et le moteur d'alertes qui déterminent quand une situation d'alerte se produit à un instant donné. La détection de tendances est indépendante du domaine d'application et trouve les k segments les plus représentatifs qui approximent une série chronologique et leur assigne des valeurs linguistiques à partir de leur inclination. Le moteur d'alertes développé en Prolog trouve toutes les solutions possibles qui satisfont une situation d'alerte et les stocke dans la base de données via le middleware.

L'intégration avec le système FSD a été paramétrée par des fichiers de configuration. Nous avons défini une politique d'évaluation dans laquelle FSD fait appel à TEMPAS pour évaluer une situation d'alerte. Nous avons présenté les composants de l'application FSD destinés à la préparation des connaissances. L'un des composants est responsable de l'affichage et de la gestion des alertes.

6 Exemple récapitulatif

6.1 Introduction 165
6.2 Situations d'alerte 165
6.3 Autres systèmes d'alertes 166
6.4 Préparation des connaissances 166
6.5 Définition des situations d'alerte 170
6.6 Evaluation des situations d'alerte 172
6.7 Notification des alertes 175
6.8 Adaptation des alertes 178
6.9 Synthèse 181

6.1 Introduction

Dans ce chapitre nous présentons un exemple récapitulatif qui permet d'illustrer le fonctionnement de TEMPAS. Deux situations d'alerte concernant la dialyse péritonéale continue ambulatoire (DPCA) ont été définies : une première situation d'alerte qui notifie les utilisateurs lorsque le patient est sous-dialysé et une deuxième concernant une émergence remarquée par le patient lorsqu'il observe le cathéter utilisé pour introduire le dialysat. Nous présentons les résultats des évaluations des situations d'alerte sur quatre patients ainsi que les alertes notifiées au personnel qui suit les patients. Nous présentons une adaptation des situations d'alerte pour prendre en compte le contexte du patient et celui des utilisateurs.

6.2 Situations d'alerte

La dialyse péritonéale est une technique de suppléance rénale qui permet au patient de gérer son traitement à domicile. Des situations d'alerte peuvent se produire après qu'un patient a été dialysé. Nous présentons deux situations d'alerte que nous avons définies avec TEMPAS.

« Pour réaliser la dialyse, une solution appelée dialysat est introduite dans la cavité péritonéale par le biais d'un cathéter […]» (CHU de Liège, 2010).

6.2.1 Sous-dialyse

La première situation d'alerte concerne la sous-dialyse qui représente une surcharge liquidienne (trop de liquide dans l'organisme). Elle se caractérise par la prise soudaine de poids et une augmentation de la tension systolique. Nous prenons la sous-dialyse comme exemple parce qu'elle est plus courante que la sur-dialyse.

6.2.2 Surveillance du cathéter

Les patients dialysés à domicile doivent prendre des précautions spéciales telles que la surveillance quotidienne du cathéter à l'occasion de la douche ou du bain (Association de Dialyse Varoise (A.DI.VA.), 2010). Cinq signes différents peuvent apparaître lors de cette surveillance : la formation d'une croûte, un saignement, une rougeur, la peau granuleuse, ou de l'écoulement. Une peau granuleuse ou de l'écoulement nécessitent une attention immédiate de la part du corps médical. En revanche, les trois autres sont des signes attendus, qui deviennent une alerte s'ils se produisent de manière répétée. Cette situation d'alerte se nomme « émergence répétitive ».

6.3 Autres systèmes d'alertes

Des systèmes d'information pour le télé suivi à domicile comme ap'Telecare de (TMM Software, 2013) proposent des alertes simples qui envoient des notifications lorsque des seuils, personnalisés pour chaque patient, sont dépassés. Ainsi, si la tension systolique dépasse 150 mm Hg ou si le poids du patient dépasse 71 Kg, une alerte est notifiée aux utilisateurs en charge du suivi du patient.

Diatélic DP est un système de télémédecine pour le suivi et la prévention en dialyse péritonéale (Diatelic, 2011). Le système fourni des alertes expertes concernant le domaine de la dialyse comme la sous-dialyse. Le moteur d'alertes est basé sur des modèles stochastiques dont la finalité est la prévention. *« Une aide au diagnostic est proposée grâce à un système sophistiqué de détection précoce de troubles de l'hydratation. De plus, un système d'alertes permet d'attirer l'attention des médecins sur les patients présentant un risque d'aggravation de leur état de santé. Les alertes et le résumé des évènements de la journée sont transmis aux spécialistes pour faciliter le suivi. »*. Il est possible de prendre en compte d'autres facteurs afin d'améliorer les alertes. Ceci nécessite du temps et d'argent (environ 60k€). *« Leur conception est bien plus difficile que pour les systèmes d'alertes classiques car elle nécessite d'abord une collaboration étroite entre l'expert médecin et l'informaticien expert dans la modélisation utilisant ces outils mathématiques. Il faut du temps pour modéliser l'expertise car avant de la modéliser il faut d'abord la comprendre. Il faut du temps, de la disponibilité des experts et des données qui sont la matière première, pour mettre en œuvre, tester et mettre au point le modèle. »*

6.4 Préparation des connaissances

Dans cette section nous présentons les connaissances qui permettent à TEMPAS de déterminer qu'une situation d'alerte se produit.

6.4.1 Ressources observables

6.4.1.1 Sous-dialyse

La sous-dialyse se caractérise par une prise de poids et par une augmentation de la tension. Nous identifions quatre ressources observables utilisées en télémédecine

pour le suivi des patients dialysés : le « poids de référence », le « poids », le « pourcentage de changement de poids (PCP) », et la « tension systolique ».

Le « poids de référence » est une caractéristique de chaque patient. Il est fixé par un médecin après une consultation du patient. Le « poids » et la « tension systolique » sont mesurés après chaque dialyse et envoyés au système d'information par une tablette ou un téléphone intelligent. Le « pourcentage de changement de poids » est calculé à partir du « poids » après dialyse du patient et son « poids de référence ». Le Tableau 28 illustre trois exemples du calcul du « pourcentage de changement de poids ».

Tableau 28. Exemples Pourcentage de changement de poids

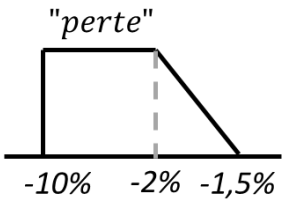
Patient	Poids de référence	Poids	Pourcentage changement de poids
1	75	77	((77-75)/75)*100 = 2.67%
2	70	77	((77-70)/70)*100 = 10%
3	60	57	((57-60)/60)*100 = -5%

Parmi les quatre ressources que nous avons présentées, seulement deux suffisent pour définir une situation d'alerte dans le cas d'une sous-dialyse, le « pourcentage de changement de poids » et la « tension ». Le Tableau 29 et le Tableau 30 récapitulent la spécification de ces deux ressources observables dans le contexte de la télémédecine. Dans ce contexte d'application, l'échantillonnage peut se faire une ou deux fois par jour entre trois et cinq fois par semaine.

Tableau 29. Ressource observable – Pourcentage de changement de poids (PCP)

Temps de validité	Temps d'expiration	Valeur minimal	Valeur maximale
3 jours	7 jours	-10%	10%

Valeurs linguistiques		
« Perte »	1	[-10%, -1.5%]
« Stable »	2	[-2%, 2%]
« Prise »	3	[1.5%, 10%]

Représentation graphique	Fonction d'appartenance
	$\mu_{perte}(x) = \begin{cases} 0 & si \quad x < -10 \; ou -1{,}5 \leq x \\ 1 & si \quad -10 \leq x \leq -2 \\ \dfrac{-1{,}5 - x}{0{,}5} & si \quad -2 < x < -1{,}5 \end{cases}$

Exemple récapitulatif

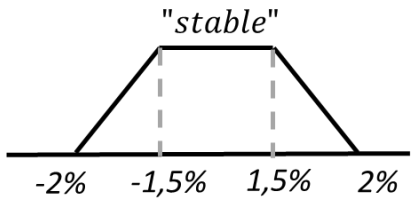
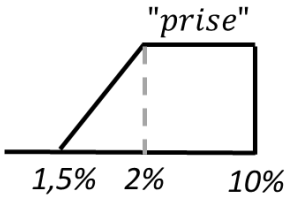
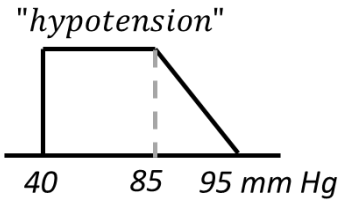
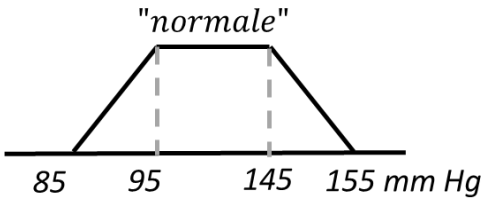
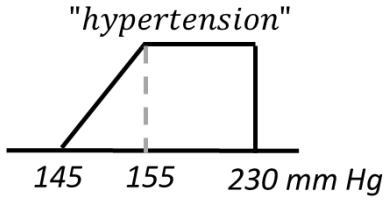
	$\mu_{stable}(x) = \begin{cases} 0 & si \quad x \leq -2 \; ou \; 2 \leq x \\ \frac{x+2}{0,5} & si \quad -2 < x < -1,5 \\ 1 & si \quad -1,5 \leq x \leq 1,5 \\ \frac{2-x}{0,5} & si \quad 1,5 < x < 2 \end{cases}$
	$\mu_{prise}(x) = \begin{cases} 0 & si \quad x \leq 1,5 \; ou \; 10 < x \\ \frac{x-1,5}{0,5} & si \quad si \; 1,5 < x < 2 \\ 1 & si \quad 2 \leq x \leq 10 \end{cases}$

Tableau 30. Ressource observable – Tension

Temps de validité	Temps d'expiration	Valeur minimal	Valeur maximale
3 jours	7 jours	40 mm Hg	230 mm Hg
Valeurs linguistiques			
« Hypotension »	1	[40 mm Hg, 95 mm Hg]	
« Normale »	2	[85 mm Hg, 155 mm Hg]	
« Hypertension »	3	[145 mm Hg, 230 mm Hg]	
Représentation graphique		**Fonction d'appartenance**	
		$\mu_{\substack{hypo \\ tension}}(x) = \begin{cases} 0 & si \quad x < 40 \; ou \; 95 \leq x \\ 1 & si \quad 40 \leq x \leq 85 \\ \frac{95-x}{10} & si \quad 85 < x < 95 \end{cases}$	
		$\mu_{normale}(x) = \begin{cases} 0 & si \quad x \leq 85 \; ou \; 155 \leq x \\ \frac{x-85}{10} & si \quad 85 < x < 95 \\ 1 & si \quad 95 \leq x \leq 145 \\ \frac{155-x}{10} & si \quad 145 < x < 155 \end{cases}$	
		$\mu_{\substack{hyper \\ tension}}(x) = \begin{cases} 0 & si \quad x \leq 145 \; ou \; 230 < x \\ \frac{x-145}{10} & si \quad si \; 145 < x < 155 \\ 1 & si \quad 155 \leq x \leq 230 \end{cases}$	

Pour définir les ensembles flous associés aux valeurs linguistiques, l'utilisateur spécifie seulement l'ordre et la plage de valeurs de chaque valeur linguistique. Ces

informations suffisent à TEMPAS pour déterminer la fonction d'appartenance correspondante.

6.4.1.2 Emergence répétitive

L'émergence répétitive est définie à partir des observations de la surveillance du cathéter. Le patient peut communiquer une ou plusieurs cases correspondant à l'émergence. La valeur observée est un score qui correspond à la somme des poids des observations cochées par le patient. Le Tableau 31 liste le poids de chaque observation. Si le patient coche tous les cas de figure, le score est de 18.

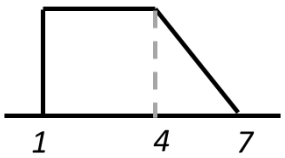
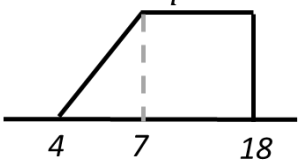
Tableau 31. Score émergence

Observation	Poids
Croûte	1
Saignement	1
Rougeur	2
Peau granuleuse	7
Ecoulement	7
Score Maximal	**18**

Exemple

Si le patient observe la formation d'une croûte et de la rougeur, la valeur observée de l'« émergence » est 3.

Tableau 32. Ressource observable – Emergence

Temps de validité	Temps d'expiration	Valeur minimal	Valeur maximale
3 jours	7 jours	1	18
Valeurs linguistiques			
« Peut attendre »	1	[1, 7]	
« Est impérative »	2	[4, 18]	
Représentation graphique		**Fonction d'appartenance**	
"*peut attendre*" 		$\mu_{peut\ attendre}(x) = \begin{cases} 0 & si \quad x < 1\ ou\ 7 \leq x \\ 1 & si \quad 1 \leq x \leq 4 \\ \frac{7-x}{3} & si \quad 4 < x < 7 \end{cases}$	
"*est impérative*" 		$\mu_{est\ impérative}(x) = \begin{cases} 0 & si \quad x \leq 4\ ou\ 18 < x \\ \frac{x-4}{3} & si \quad si\ 4 < x < 7 \\ 1 & si \quad 7 \leq x \leq 18 \end{cases}$	

6.4.2 Equation d'alerte

L'équation d'alerte regroupe les ressources observables qui interviennent dans la définition d'une situation d'alerte. Elle sert aussi à définir les graphes modificateurs pour le calcul des valeurs linguistiques des ressources. La Figure 63 et la Figure 64 montrent les équations d'alerte utilisées pour définir les deux situations d'alerte, la sous-dialyse et l'émergence répétitive.

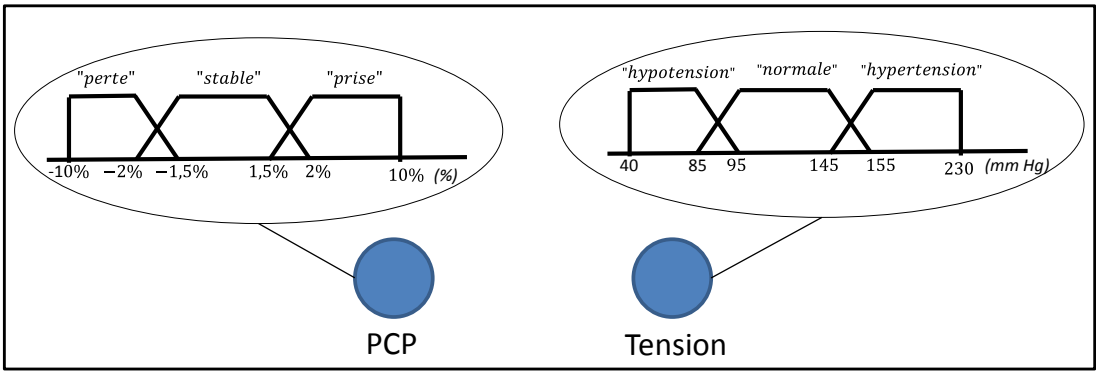


Figure 63. Equation d'alerte – dialyse

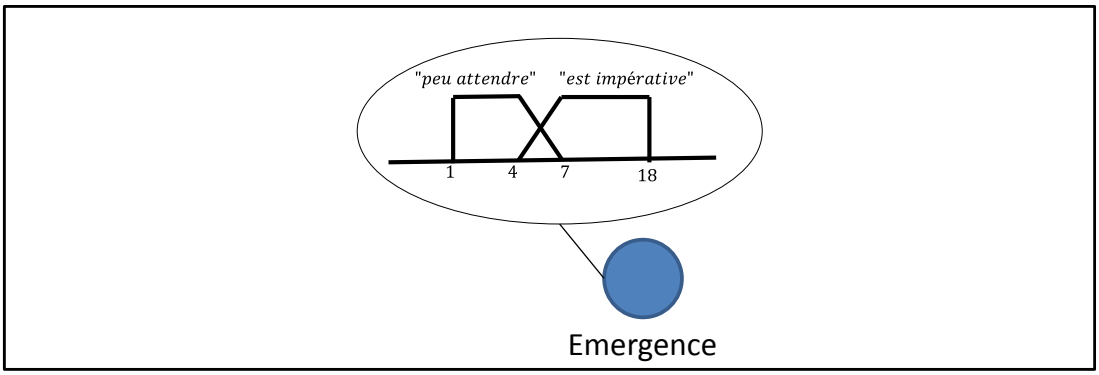


Figure 64. Equation d'alerte – émergence

6.5 Définition des situations d'alerte

Dans cette section nous définissons les situations d'alerte ainsi que la politique de notification associée à chacune des situations précédentes.

6.5.1 Sous-dialyse

La sous-dialyse est définie avec deux conditions d'activation intrinsèques également importantes

$$\text{Sous-dialyse} : (\text{PCP} = \text{« prise »}, 0.5) \\ \text{et } (\text{tension} = \text{« hypertension »}, 0.5)$$

Si une seule ou les deux conditions sont satisfaites, une alerte doit être notifiée aux utilisateurs concernés. Le seuil de filtrage concernant l'indice d'applicabilité de

l'alerte (IAA) est fixé à zéro. Par contre, le seuil de filtrage concernant l'indice de confiance de l'alerte (ICA) est fixé à 0.5 pour éviter les alertes détectées sur des observations « très » anciennes.

Seuil IAA = 0
Seuil IAC = 0.5

Le Tableau 33 montre le cycle de vie des alertes détectées concernant la sous-dialyse. Le temps de validité pour la sous-dialyse a été défini à une journée puisque le « poids » et la « tension » sont envoyés une fois par jour après la dialyse. Le cycle de vie de l'alerte spécifie que les notifications explicites doivent toujours avoir lieu lorsqu'une alerte a été notifiée, elle est « de la journée » et elle n'a toujours pas été traitée. Le traitement d'une alerte suppose la prise de décision correspondante : prévenir le médecin traitant ou l'infirmier.

Temps de validité = 1 jour

Tableau 33. Politique de notification – sous-dialyse

	Scénario			
	nouvelle-pendant-traitée	**nouvelle-pendant-nouvelle**	**nouvelle-après-traitée**	**nouvelle-après-périmée**
Notifier si				
IAA Augmente	Oui	Non	Oui	Oui
IAA Diminue	Oui	Non	Oui	Oui
ICA Augmente	Oui	Non	Oui	Oui
ICA Diminue	Oui	Non	Oui	Oui
Enrichir par la	Droite	Droite		
IAA	Oui	Oui		
ICA	Oui	Oui		
Temps de validité	Non	Non		
Hériter l'état	Oui			

6.5.2 Emergence répétitive

L'émergence répétitive est définie à partir de deux conditions d'activation intrinsèques. La tendance est plus importante puisque ce qui compte c'est la répétition d'une émergence qui « peut attendre ».

Emergence répétitive : (Emergence = « peut attendre », 0.3)
et (tendance de l'émergence ≥ « stable », 0.7)

Nous fixons l'indice d'applicabilité de l'alerte ainsi que l'indice de confiance (ICA) (IAA) à 0.5. Pour qu'une alerte puisse être notifiée, une tendance « stable » ou « à la hausse » de l'émergence doit être détectée.

Seuil IAA = 0,5
Seuil IAC = 0.5

La détection de tendances utilise les observations datant des 20 derniers jours d'observations si et seulement si la série chronologique est composée au minimum de 4 observations.

Le Tableau 34 montre le cycle de vie des alertes détectées concernant l'émergence répétitive. Le temps de validité a été défini à cinq jours puisque c'est le temps moyen d'attente avant de changer le matériel qui cause le problème. Si l'utilisateur a traité une alerte concernant une émergence répétitive, seules les alertes des cinq jours après la notification initiale doivent être notifiées.

Temps de validité = 5 jours

Tableau 34. Politique de notification – sous-dialyse

	Scenario			
	nouvelle-pendant-traitée	**nouvelle-pendant-nouvelle**	**nouvelle-après-traitée**	**nouvelle-après-périmée**
Notifier si				
IAA Augmente	Oui	Oui	Oui	Oui
IAA Diminue	Non	Oui	Oui	Oui
ICA Augmente	Oui	Oui	Oui	Oui
ICA Diminue	Non	Oui	Oui	Oui
Enrichir par la	Gauche	Droite		
IAA	Non	Oui		
ICA	Non	Oui		
Temps de validité	Oui	Non		
Hériter l'état	Non			

Il est possible d'utiliser une ressource observable qui calcule le nombre d'observations dans les derniers « sept jours » avec un score inférieur à 4. Toutefois l'inconvénient d'une telle approche, est la méthode de calcul de l'observation de la ressource puisque les « sept jours » ont été définis dans la vue de la base de données. A la différence du Pourcentage de Changement de Poids (PCP) qui est une mesure « standard » dans la dialyse, le nombre d'émergences consécutives afin de considérer une alerte comme « émergence répétitive » est défini par l'utilisateur. Ainsi, il peut vouloir modifier le nombre de jours (les observations des « dix derniers jours », par exemple) servant à déterminer des émergences répétitives. Il doit être en mesure de le faire sans avoir besoin de créer un vue chaque fois que la période change. En plus de laisser le choix à l'utilisateur de définir la période pertinente, utiliser la tendance permet d'exprimer, avec l'indice de confiance de l'alerte, la distance temporelle entre les observations (Tableau 38).

6.6 Evaluation des situations d'alerte

Dans cette section nous présentons la politique d'évaluation. Nous présentons aussi les données des patients que nous utilisons pour évaluer les situations d'alerte.

Nous utilisons dans la plupart des cas, des données qui déclenchent des alertes, car celles qui n'en déclenchent pas ne montrent pas les fonctionnalités de TEMPAS.

6.6.1 Politique d'évaluation

L'évaluation des situations d'alerte est événementielle. Chaque fois que le patient envoie son poids, sa tension, ou des observations d'émergence du cathéter, le système évalue les situations d'alerte concernées. Mais les situations d'alerte sont aussi évaluées périodiquement, une fois par jour, afin de prendre en compte des alertes qui auraient pu être détectées (le weekend par exemple) et qui n'ont pas pu été traitées.

Exemple

Si une alerte de sous-dialyse a été détectée le vendredi et que personne ne l'a « traitée », elle sera normalement « périmée » le samedi. Or, il est nécessaire de la prendre en compte. Une réévaluation de la situation d'alerte dimanche ou lundi détectera une « nouvelle » alerte à partir des mêmes observations et la notifiera aux utilisateurs concernés (section 5.3.3.3).

6.6.2 Données des patients dialysés

Voici quatre patients dialysés et leur poids de référence respectif : « M. **A**lbert », 63.5 Kg ; « Mme. **B**rigitte », 59 Kg; « M. **C**harles », 99 Kg; et « M. **D**avid », 69 Kg.

Le Tableau 35 liste deux jours de données menant à l'évaluation de la « sous-dialyse » et le Tableau 36 liste onze jours de données menant à l'évaluation de l'« émergence répétitive ». Supposons que les patients sont suivis à distance par le même utilisateur.

Tableau 35. Données – sous dialyse

Patient	Envoyées			Connues		Calculées	
	Horo-datage	Tension systolique (mm Hg)	Poids de référence (Kg)	Horo-datage	Poids mesuré (Kg)	Horo-datage	PCP (%)
« M. **A**lbert »	21/07/14 6h00	153	63.5	26/06/14 9h00	62	21/07/14 6h00	-2.36
« Mme. **B**rigitte »	21/07/14 6h15	151	59	26/06/14 10h00	59.9	21/07/14 6h15	1.52
« M. **C**harles »	21/07/14 7h00	118	99	26/06/14 11h00	100.8	21/07/14 7h00	1.81
« M. **D**avid »	21/07/14 10h25	154	69	26/06/14 14h00	71.2	21/07/14 10h25	3.19
« M. **A**lbert »	22/07/14 6h02	150	63.5	26/06/14 9h00	62.4	22/07/14 6h02	-1.73
« Mme. **B**rigitte »	22/07/14 6h15	153	59	26/06/14 10h00	60.2	22/07/14 6h15	2.03

Exemple récapitulatif

Patient	Envoyées			Connues		Calculées	
	Horo-datage	Tension systolique (mm Hg)	Poids de référence (Kg)	Horo-datage	Poids mesuré (Kg)	Horo-datage	PCP (%)
« M. **C**harles »	22/07/14 6h59	136	99	26/06/14 11h00	106.7	22/07/14 6h59	7.78
« M. **D**avid »	22/07/14 10h30	162	69	26/06/14 14h00	73.1	22/07/14 10h30	5.94

Tableau 36. Données – émergence répétitive

Patient	Horodatage	Observation	Score
« Mme. **B**rigitte »	19/07/14 6h15	Rougeur	2
« Mme. **B**rigitte »	20/07/14 6h10	Rougeur	2
« Mme. **B**rigitte »	21/07/14 6h15	Rougeur	2
« Mme. **B**rigitte »	22/07/14 6h20	Rougeur	2
« Mme. **B**rigitte »	26/07/14 6h10	Rougeur	2
« Mme. **B**rigitte »	27/07/14 6h15	Rougeur	2
« Mme. **B**rigitte »	29/07/14 6h18	Rougeur	2

Le Tableau 37 affiche le résultat des évaluations de la situation d'alerte pour la sous-dialyse. Les évaluations 1, 2 et 3 ont eu lieu tout de suite après que les données ont été connues par le système. La quatrième est une évaluation périodique. La cinquième et dernière évaluation est déclenchée par l'arrivée des données datant de plus d'une semaine. Toutefois, l'évaluation utilise les observations connues les plus récentes de « M. Albert ».

Le Tableau 38 montre les évaluations concernant l'émergence répétitive. Nous supposons que les évaluations de la situation d'alerte « émergence répétitive » sont déclenchées par l'arrivée des nouvelles données de l'émergence. Si la série chronologique utilisée pour détecter les tendances n'a pas plus de quatre observations, aucun segment ni tendance ne sont détectés. Les 3 dernières évaluations génèrent des alertes avec une Indice d'applicabilité de 0.83 parce que le segment utilisé pour déterminer la tendance est composé au moins par un segment dont les observations ont eu lieu entre 3 et 7 jours correspondant respectivement aux temps de validité et d'expiration de la ressource « émergence » dans le contexte d'application. Le segment qui cause cette diminution de confiance est [(22/07/14 06h10, 2), (26/07/14 06h10, 2)].

Tableau 37. Résultat des évaluations de la sous-dialyse

Patient	Horo-datage	Tension syst. (mm Hg)	$\mu_{\substack{hyper\\tension}}$	PCP (%)	μ_{prise}	Horo-datage Evaluation	IAA	ICA	*Eval. #*
« M. **A**lbert »	21/07/14 6h00	153	0.8	-2.36	0	*29/07/14 12h00*	0.25	0.5	*5*

Patient	Horo-datage	Tension syst. (mm Hg)	$\mu_{\substack{hyper \\ tension}}$	PCP (%)	μ_{prise}	Horo-datage Evaluation	IAA	ICA	*Eval. #*
« Mme. **B**rigitte »	21/07/14 6h15	151	0.6	1.52	0.05	*21/07/14 6h15*	0.33	1	*1*
« M. **C**harles »	21/07/14 7h00	118	0	1.81	0.64	*21/07/14 7h00*	0.32	0.5	*2*
« M. **D**avid »	21/07/14 10h25	154	0.9	3.19	1	*21/07/14 10h25*	0.95	1	*3*
« M. **A**lbert »	22/07/14 6h02	150	0.5	-1.73	0	*22/07/14 11h00*	0.25	0.5	*4*
« Mme. **B**rigitte »	22/07/14 6h15	153	0.8	2.03	1	*22/07/14 11h00*	0.9	1	*4*
« M. **C**harles »	22/07/14 6h59	136	0	7.78	1	*22/07/14 11h00*	0.5	0.5	*4*
« M. **D**avid »	22/07/14 10h30	162	1	5.94	1	*22/07/14 11h00*	1	1	*4*

Tableau 38. Résultat des évaluations de l'« émergence répétitive »

Patient	Horo-datage	Score	$\mu_{\substack{peu \\ attendre}}$	Segment	°	μ_{stable}	IAA	ICA	*Eval. #*
« Mme. **B**rigitte »	19/07/14 6h15	2	1	__	__	0	0.33	0.33	1
« Mme. **B**rigitte »	20/07/14 6h10	2	1	__	__	0	0.33	0.33	2
« Mme. **B**rigitte »	21/07/14 6h15	2	1	__	__	0	0.33	0.33	3
« Mme. **B**rigitte »	**22/07/14 6h10**	2	1	[(19/07/14 6h15, 2), (22/07/14 6h10, 2)]	0	1	1	1	4
« Mme. **B**rigitte »	**26/07/14 6h10**	2	1	[(20/07/14 6h10, 2), (26/07/14 6h10, 2)]	0	1	1	0.83	5
« Mme. **B**rigitte »	27/07/14 6h15	2	1	[(21/07/14 6h15, 2), (27/07/14 6h15, 2)]	0	1	1	0.83	6
« Mme. **B**rigitte »	29/07/14 6h18	2	1	[(22/07/14 6h10, 2), (29/07/14 6h18, 2)]	0	1	1	0.83	7

6.7 Notification des alertes

Nous présentons dans cette section le descriptif des notifications des alertes par rapport aux données précédentes, les situations d'alerte, et la politique de notification. Le descriptif des notifications des deux situations d'alerte est présenté dans le Tableau 39. Notons que la notification du 21/07/14 à 7h00 affiche la valeur quantitative de la tension et non la valeur linguistique parce que la valeur linguistique que peut représenter 118 mm Hg est « normal » ce qui ne satisfait pas la condition « si 'hypertension' ».

Exemple récapitulatif

Tableau 39. Notification – sous dialyse

Horodatage	Action	Description
19/07/14 6h15	**Aucune notification**	Un IAA de 0.33 ne suffit pas pour qu'une émergence répétitive puisse être considérée comme telle
20/07/14 6h10	**Aucune notification**	Un IAA de 0.33 ne suffit pas pour qu'une émergence répétitive puisse être considérée comme telle
21/07/14 6h15	**Notification explicite** « Sous-dialyse Mme. Brigitte le 21/07/14 à 6h15 Tension : 151 – 'hypertension' PCP : 1.52 – 'prise' IAA : 0.33 ICA : 1 » **Aucune notification** concernant l'émergence répétitive de « Mme. Brigitte ».	Les deux conditions sont partiellement satisfaites
21/07/14 7h00	**Notification explicite** « Sous-dialyse M. **C**harles le 21/07/14 à 7h00 Tension : 118 PCP : 1.81 – 'prise' IAA : 0.32 ICA : 0.5 »	L'IAA n'est pas très élevé. Il n'y a pas d'urgence à traiter l'alerte : une seule condition est partiellement satisfaite : prise de poids.
21/07/14 7h10	L'utilisateur « traite » l'alerte de sous-dialyse concernant « Mme. Brigitte »	Pour la traiter, l'utilisateur va chercher le compte rendu de la dernière consultation
21/07/14 10h25	**Notification explicite** « Sous-dialyse M. David le 21/07/14 à 10h25 Tension : 154 – 'hyperthermie' PCP : 3.19 – 'prise' IAA : 0.95 ICA : 1 »	Le patient est très concerné par l'alerte
21/07/14 10h50	L'utilisateur « traite » l'alerte de sous-dialyse de « M. David »	Pour la traiter, l'utilisateur va chercher le compte rendu de la dernière consultation

Horodatage	Action	Description
22/07/14 6h10	**Notification explicite** « Emergence répétitive Mme. Brigitte le 22/07/14 à 6h10 Emergence : 2 – 'peut attendre' Tendance émergence : stable entre le 19/07/14 à 6h15 et le 22/07/14 à 6h10 IAA : 1 ICA : 1 »	Le message n'affiche pas le nombre d'observations qui ont été utilisées pour déterminer une « émergence répétitive »
22/07/14 7h00	L'utilisateur traite l'alerte d'émergence répétitive de « Mme. Brigitte ». Le médecin responsable a aussi été notifié de l'alerte.	L'utilisateur commande du nouveau matériel pour « Mme. Brigitte ».
22/07/14 10h00	Le système change l'état de l'alerte de « sous-dialyse M. Charles » à l'état « périmée »	Lors que le système va récupérer les alertes à afficher à l'utilisateur, il « périme » les alertes qui ont dépassé le temps de validité
22/07/14 11h00	**Notification explicite** « Sous-dialyse M. Albert le 22/07/14 à 6h02 Tension : 150 - 'hypertension' PCP : -1.73 IAA : 0.25 ICA : 0.5 » **Notification explicite** « Sous-dialyse Mme. Brigitte le 22/07/14 à 6h15 Tension : 153 – 'hypertension' PCP : 2.03 – 'prise' IAA : 0.9 ICA : 1 » **Notification explicite** « Sous-dialyse M. Charles le 22/07/14 à 6h59 Tension : 136 PCP : 7,78 – 'prise' IAA : 0.5 ICA : 0.5 » **Notification explicite** « Sous-dialyse M. David le 22/07/14 à 10h30 Tension : 162 – 'hypertension' PCP : 5.94 – 'prise' IAA : 1 ICA : 1 »	4 alertes ont été notifiées explicitement en même temps. La notification de « M. Albert » est explicite parce que c'est la première fois qu'elle est détectée. Les notifications de « Mme Brigitte » et « M. David » sont explicites puisque la nouvelle alerte se trouve dans le scénario nouvelle_après_traitée La notification de « M. Charles » est explicite puisque la nouvelle alerte se trouve dans le scénario nouvelle_après_périmée

Exemple récapitulatif

Horodatage	Action	Description
22/07/14 11h05	L'utilisateur priorise les alertes à l'aide de l'IAA. Il commence par « traiter » l'alerte concernant « M. David » et finit par celle concernant « M. Albert ».	
26/07/14 6h10	**Notification implicite** « Emergence répétitive Mme. Brigitte le 26/07/14 à 6h10 Emergence : 2 – 'peut attendre' Tendance émergence : 'stable' entre le 20/07/14 à 6h10 et le 26/07/14 à 6h10 IAA : 1 ICA : 0. 83 »	L'alerte est dans le scenario nouvelle_pendant_traitée. Elle est notifiée implicitement parce que l'ICA n'est pas supérieur à la précédente. Ensuite elle hérite de l'état « traitée » puisque le matériel n'a toujours pas été, en principe, changé à cette date
27/07/14 6h15	**Notification explicite** « Emergence répétitive Mme. Brigitte le 27/07/14 à 6h15 Emergence : 2 – 'peut attendre' Tendance émergence : 'stable' entre le 21/07/14 à 6h15 et le 27/07/14 à 6h15 IAA : 1 ICA : 1 »	L'alerte est dans le scénario nouvelle_après_traitée Le matériel a dû être changé et le problème persiste. L'utilisateur est notifié explicitement
29/07/14 6h18	**Notification explicite** « Emergence répétitive Mme. Brigitte le 29/07/14 à 6h18 Emergence : 2 – 'peut attendre' Tendance émergence : 'stable' entre le 22/07/14 à 6h10 et le 29/07/14 à 6h18 IAA : 1 ICA : 1 »	L'alerte est dans le scénario nouvelle_pendant_nouvelle La notification explicite rappelle encore à l'utilisateur de prendre des mesures
29/07/14 12h00	Aucune notification	Les observations récupérées de « M. Albert » ont déjà généré une alerte dont l'état est « traitée »

6.8 Adaptation des alertes

Dans cette section, nous présentons deux adaptations concernant la sous-dialyse. La situation d'alerte comme telle n'est pas modifiée parce que l'utilisateur est toujours intéressé par les mêmes caractéristiques d'une sous-dialyse, une prise de poids et une hypertension. Toutefois, de nouvelles informations sont disponibles dans le système d'information. Il s'agit d'une caractéristique du patient qui exprime s'il est, ou non cardiaque.

6.8.1 Adaptations du calcul des valeurs linguistiques

Les seuils pour déterminer si un patient est hypertendu varient en fonction de s'il est, ou non, cardiaque. L'équation d'alerte est adaptée afin de prendre en compte cette nouvelle information. La Figure 65 montre la nouvelle équation d'alerte et la modification des valeurs linguistiques. Le Tableau 40 montre les connaissances de cette caractéristique par le système d'information.

Tableau 40. Patients cardiaques

Patient	Cardiaque	Horodatage
« M. **A**lbert »	Oui	26/06/14 9h00
« Mme. **B**rigitte »	Non	26/06/14 10h00
« M. **C**harles »	-	
« M. **D**avid »	-	

6.8.2 Nouvelles informations pour les alertes

Lorsqu'une alerte est notifiée, l'utilisateur récupère des informations du dossier du patient afin de prendre une décision, par exemple, le compte rendu de la dernière consultation. Les utilisateurs gagneraient du temps si l'alerte notifiait le dernier compte rendu ainsi que le poids mesuré et le poids de référence. Le « poids de référence » et le « poids » ont déjà été présentés comme de ressources observables. Le « dernier compte rendu » est aussi une ressource qui peut s'observer chez les patients. Ainsi il peut être exploité comme ressource observable dans une équation d'alerte. En revanche, TEMPAS ne saurait pas évaluer des conditions d'activation formées à partir de cette ressource dont les valeurs observées sont textuelles.

Pour améliorer le message des alertes, il suffit d'ajouter des ressources observables dans l'équation de l'alerte. La dernière observation connue de la ressource est utilisée pour former le message de l'alerte. La Figure 65 montre l'équation d'alerte modifiée. Notons que la situation d'alerte n'a pas changé, et que seule la ressource « cardiaque » affecte les nouveaux résultats des évaluations de la situation d'alerte « sous-dialyse ».

6.8.3 Notifications

Supposons que l'évaluation de la situation d'alerte « sous-dialyse » utilise l'équation d'alerte que nous avons modifiée. Le Tableau 41 liste les alertes notifiées. En surligné, les changements par rapports aux alertes notifiées dans la section 6.7. Nous affichons seulement les quatre alertes détectées le 22/11/14 à 11h. A la différence du « poids de référence », du « poids », et de la tension, les « comptes rendus » sont inventés et succincts.

Exemple récapitulatif

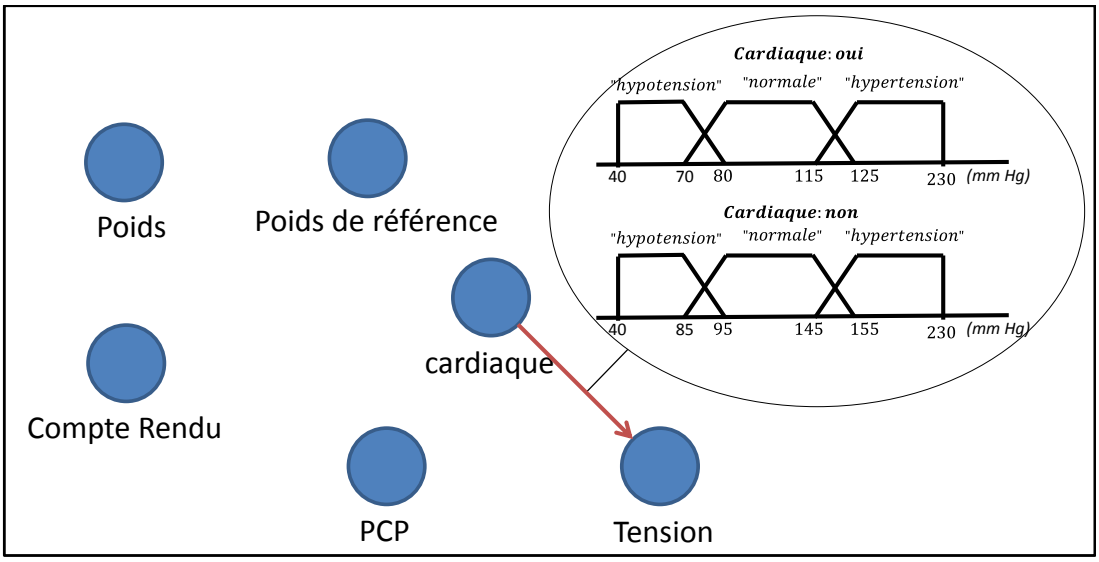


Figure 65. Equation d'alerte modifiée – dialyse

Tableau 41. Notifications adaptées

Horodatage	Action	Description
22/07/14 11h00	« Sous-dialyse M. Albert le 22/07/14 à 6h02 Tension : 150 – 'hypertension' PCP : -1.73 Poids : 62.4 Poids de référence : 63.5 - 26/06/14 9h00 Cardiaque : Oui - 26/06/14 9h00 Compte rendu : 'Il est nécessaire de surveiller ce patient qui est cardiaque' -26/06/14 9h00 IAA : 0.5 ICA : 0.5 »	Le patient est cardiaque donc la tension de 150 est sans doute une « hypertension »
22/07/14 11h00	« Sous-dialyse Mme. Brigitte le 22/07/14 à 6h15 Tension : 153 – 'hypertension' PCP : 2.03 – 'prise' Poids : 60.2 Poids de référence : 59.9 - 26/06/14 10h00 Cardiaque : Non - 26/06/14 10h00 Compte rendu : 'La patiente montre des signes d'amélioration' -26/06/14 10h00 IAA : 0.9 ICA : 1 »	

Horodatage	Action	Description
22/07/14 11h00	**Notification explicite** « Sous-dialyse M. Charles le 22/07/14 à 6h59 Tension : 136 PCP : 7,78 – 'prise' Poids : 106.7 Poids de référence : 99 - 26/06/14 11h00 IAA : 0.5 ICA : 0.5 »	L'observation concernant si le patient est ou non cardiaque est manquante. Dans ce cas, il utilise les fonctions d'appartenance associées à la ressource et non celles associés au lien entre « cardiaque » et « tension ». Le système ne connaît aucune observation du « compte rendu » du patient.
22/07/14 11h00	**Notification explicite** « Sous-dialyse M. David le 22/07/14 à 10h30 Tension : 162 – 'hypertension' PCP : 5.94 – 'prise' Poids : 73.1 Poids de référence : 69 - 26/06/14 11h00 IAA : 1 ICA : 1 »	

6.9 Synthèse

Nous avons présenté deux exemples de situations d'alerte concernant la dialyse péritonéale : la sous-dialyse et l'émergence répétitive. Dans le Tableau 42 , nous dressons un comparatif succinct entre TEMPAS et les systèmes de dialyse présentés en début de chapitre.

Tableau 42. Différences entre les systèmes

Critère	Ap'Telecare	Diatélic	TEMPAS
Objectif principal	Suivi à domicile	Système expert dans le domaine de la dialyse à domicile	Système d'alertes personnalisable indépendant du domaine d'application
Situations d'alerte connues (sous-dialyse, par exemple)	Nombreuses alertes faisant références au même incident	Des alertes détectées très pointues dans le domaine	Il est possible de paramétrer la situation d'alerte, mais il faut trouver le bon paramétrage

Exemple récapitulatif

Critère	Ap'Telecare	Diatélic	TEMPAS
Situations d'alerte personnalisables (émergence répétitive, par exemple), ou des nouvelles données à utiliser dans des situations d'alerte connues	Sur des valeurs numériques et de seuils	Il est possible d'avoir des alertes pointues, mais il faut changer le modèle mathématique afin de générer les alertes utiles. Demande un investissement de temps, de compétences et d'argent	Il est possible de paramétrer la situation d'alerte, mais il faut trouver le bon paramétrage.
Adaptation à l'évolution du patient (nouveau poids de référence, par exemple)	Réadaptation des seuils pour filtrer les alertes détectées à partir du poids du patient	Selon le type de modèle mathématique utilisé. Mais probablement par auto-adaptation	La situation d'alerte ne change pas. La détection d'alertes prend en compte automatiquement les changements des « ressources » impliquées dans la situation d'alerte
Adaptation des situations d'alerte, et des alertes détectées aux préférences de chaque utilisateur	----------	----------	Il est possible de définir autant de situations d'alerte adaptées aux préférences de chaque utilisateur. Elles peuvent toutes utiliser la même équation d'alerte, ou définir une équation d'alerte spécifique aussi aux préférences des utilisateurs pour afficher

Nous avons présenté la définition des situations d'alerte, leur évaluation, ainsi que la notification des alertes détectées. Les situations d'alerte ont été définies dans le cadre de la dialyse à domicile. Les paramétrages ont été définis avec l'aide d'un infirmier en néphrologie.

7 Validation

7.1 Introduction 183
7.2 Indépendance du domaine d'application 183
7.3 Flexibilité et adaptabilité 194

7.1 Introduction

Dans le but de valider nos propositions, deux jeux de tests ont été réalisés. L'objectif du premier est de valider l'*indépendance du domaine d'application*. Le deuxième valide la *flexibilité et adaptabilité* du système.

7.2 Indépendance du domaine d'application

L'objectif de TEMPAS est d'être indépendant du domaine d'application. Les situations d'alerte qu'un utilisateur peut définir sont celles qui sont pertinentes pour lui. L'outil permet de les définir et détecter tandis que la *finalité* des alertes détectées est celle que l'utilisateur veut lui donner.

Nous avons utilisé TEMPAS pour détecter des alertes à partir de données issues des dispositifs de monitoring d'une unité de soins intensifs de nouveaux-nés. Au total, sept paramètres physiologiques ont été surveillés et utilisés pour identifier vingt-six scénarii. Un scénario contient 45 minutes de données de deux ou plusieurs paramètres physiologiques. Huit actions sont envisagées et elles sont évaluées comme actions objectives, valides, neutres, ou inappropriées pour chaque scénario. Chaque situation d'alerte TEMPAS a une finalité associée qui est une action-à-faire parmi les actions possibles dans la validation du système BT-45.

7.2.1 Présentation de BT-45

BT-45 est un prototype qui fait des résumés de 45 minutes afin de démontrer l'efficacité des résumés textuels sur les graphiques lors de la prise de décision (Portet, et al., 2009). BT-45 utilise deux sources de données liées à la surveillance de nouveaux-nés : les *données physiologiques* obtenues en continu par le système *badger 2* ; et les données obtenues discrètement par le système *BabyWatch* et saisies par un *observateur*.

Le système *Badger 2* contrôle :

- (Phys-1) la « fréquence cardiaque » ;
- (Phys-2) la « pression transcutanée d'oxygène » (TcPO2) ;
- (Phys-3) la « pression transcutanée de dioxyde de carbone » (TcPCO2) ;
- (Phys-4) la « saturation artérielle en oxygène » (SaO2) ;
- (Phys-5) la « température périphérique » de l'incubateur ;
- (Phys-6) la « température du nouveau-né » ;

- (Phys-7) et la « pression artérielle ».

BabyWatch est un système qui facilite la saisie de données concernant des nouveaux-nés. Parmi les données récupérées se trouvent les actions faites par le corps médical lors d'une intervention sur un bébé, ce que l'observateur perçoit du comportement du bébé, et les alarmes déclenchées par les dispositifs de monitoring.

Tableau 43. Résultats attendus des tests

Id scénario	Action							Ne rien faire
	Act-A	Act-B	Act-C	Act-D	Act-E	Act-F	Act-G	
Sc-1	**O**	N	I	N	I	I	I	I
Sc-2	V	**O**	N	V	I	N	N	I
Sc-3	I	N	**O**	N	I	I	I	I
Sc-4	V	N	I	**O**	I	I	I	I
Sc-5	I	N	I	N	**O**	I	I	I
Sc-6	N	N	I	V	I	**O**	V	I
Sc-7	V	N	I	V	I	V	**O**	I
Sc-8	N	N	I	N	V	I	I	**O**
Sc-9	**O**	N	I	V	I	I	I	V
Sc-10	V	**O**	N	I	I	N	I	I
Sc-11	I	N	**O**	I	I	I	I	I
Sc-12	N	N	N	**O**	I	I	N	I
Sc-13	I	N	I	V	**O**	I	N	I
c-14	V	N	I	V	N	**O**	V	I
Sc-15	V	N	I	I	I	I	**O**	I
Sc-16	N	N	N	N	I	I	N	**O**
Sc-17	**O**	N	I	I	I	I	V	I
Sc-18	V	**O**	N	N	I	N	N	I
Sc-19	I	N	**O**	N	V	I	I	I
Sc-20	V	N	I	**O**	V	I	N	I
Sc-21	I	N	I	V	**O**	I	I	I
Sc-22	V	N	I	V	V	**O**	I	I
Sc-23	V	N	I	V	I	I	**O**	I
Sc-24	I	I	N	I	I	I	I	**O**
Sc-25	?	?	?	?	?	?	?	V
Sc-26	V	?	?	V	V	?	?	?

O : Action Objectif	**V : Action Valide**	**N : Action Neutre**	**I : Action Inappropriée**	**? : Action Inconnue**

BT-45 a été validé à l'aide de vingt-six scénarii de test. Chaque scénario utilise 45 minutes des données concernant un patient et cible une seule « action objective », des

« actions valides », des « actions neutres », et des « actions inappropriées ». Les *actions-à-faire* sont :

- (Act-A) ajuster la « fraction d'oxygène inspirée » ;
- (Act-B) ajuster la « ventilation en pression positive continue » ;
- (Act-C) extuber ;
- (Act-D) gérer la « température de l'incubateur » ;
- (Act-E) surveiller l'équipement ;
- (Act-F) aspirer ;
- (Act-G) assister la « pression artérielle » ;
- et « ne rien faire ».

Le Tableau 43 liste les résultats attendus des vingt-six scénarii de test utilisés pour valider BT-45. Les scénarii ont été représentés par :

- (Rep-1) une *présentation graphique* des données physiologiques issues de *Badger 2* ;
- (Rep-2) un *résumé rédigé* par trois experts ;
- (Rep-3) et un *résumé généré* automatiquement par BT-45 à partir des données de *Badger 2* et *BabyWatch*.

L'une des trois représentations et un *préambule* du patient étaient fournis aux *utilisateurs testeurs* afin de prendre une ou plusieurs actions. Dans l'idéal, l'action choisie par l'utilisateur est cohérente avec l'« action objective » (**O**) du scénario présenté dans le Tableau 43.

7.2.2 Identification des situations d'alerte TEMPAS

Pour valider l'indépendance du domaine d'application de TEMPAS, nous avons assigné une *finalité* aux situations d'alerte, soit une *action-à-faire*.

Exemple

« Ajuster la fraction d'oxygène inspirée » (Act-A) est l'*action-à-faire* associée à la situation d'alerte Sit-A.

« Assister la pression artérielle » (Act-G) est l'*action-à-faire* associée à la situation d'alerte Sit-G.

Au total, sept situations d'alerte (Act-A - Act-G) ont été définies. Si une alerte est détectée par TEMPAS, cela s'interprète comme une décision à prendre, soit l'*action-à-faire* associée à la situation d'alerte.

Exemple

Si lors de l'évaluation de la situation d'alerte Sit-A, une alerte est détectée, il faut « Ajuster la fraction d'oxygène inspirée ».

Si lors de l'évaluation de la situation d'alerte Sit-C, une alerte est détectée, il faut « assister la pression artérielle »

7.2.3 Récupération des données

Nous avons récupéré les données physiologiques monitorées par *Badger 2*. Chaque variable surveillée est représentée par une ressource observable dans le modèle de TEMPAS.

La « fréquence cardiaque », la « pression transcutanée d'oxygène » (TcPO2), la « pression transcutanée de dioxyde de carbone » (TcPCO2), la « saturation artérielle en oxygène » (SaO2), la « température périphérique » de l'incubateur, la « température du nouveau-né », et la « pression artérielle » monitorés par Badger 2 sont des ressources observables dans le modèle de TEMPAS

Les données ont été prétraitées avant de les rendre exploitables par TEMPAS. Les observations de Badger sont faites par « intervalles temporels ». Un registre contient toutes les valeurs observées séquentiellement chaque seconde pendant l'intervalle de temps associé. En revanche, les observations dans le modèle de TEMPAS sont faites par « instants temporels ». Un registre est la valeur observée à un instant donné. La valeur insérée est la moyenne des valeurs observées pendant chaque minute par *Badger 2*.

7.2.4 Définition des situations d'alerte TEMPAS

La définition des situations d'alerte de TEMPAS a été guidée par le raffinement progressif des situations d'alerte jusqu'à l'absence (si possible) de faux-négatifs et de faux-positifs. Ainsi, lorsqu'une alerte est détectée et notifiée, l'utilisateur sait quelle décision il doit prendre.

Le protocole appliqué sur chacune des sept situations d'alerte TEMPAS est représenté dans la Figure 66.

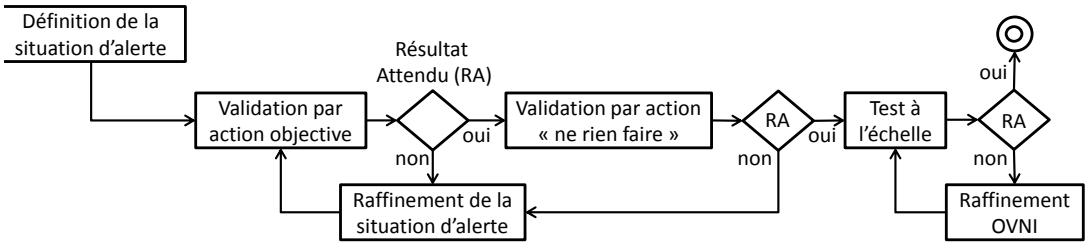


Figure 66. Validation des situations d'alerte

Afin de définir les situations d'alerte, nous avons préparé les *ressources observables* et nous avons construit les *conditions d'activation*. Pour chaque ressource observable nous avons défini :

- les valeurs minimale et maximale mesurables
- les valeurs linguistiques et les ensembles flous respectifs pour représenter l'état et la tendance
- le temps de validité et d'expiration des observations

Par manque d'expertise dans le domaine des nouveaux-nés, nous avons récupéré ces valeurs depuis Wikipédia.

Pour définir les *conditions d'activation* pour chaque situation d'alerte, nous nous sommes servis des trois représentations des scénarii pour la validation de *BT-45*, les graphiques et les deux résumés. Les représentations n'utilisent pas toutes les ressources observables.

Exemple

Si la représentation graphique (Rep-1) d'un scénario utilise seulement deux des sept variables physiologiques monitorées (Phys-1 – Phys-7), nous interprétons cela comme deux conditions d'activation, chacune construite à partir d'une ressource observable.

Une situation d'alerte TEMPAS est associée à une *action-à-faire* qui est l'« action objective » ciblée au moins par trois scenarii de test de validation de BT-45 (Tableau 43). Ceci veut dire qu'au moins trois représentations graphiques, trois résumés faits par des experts, et trois résumés générés automatiquement par BT-45 nous fournissent des informations pour la construction des situations d'alerte.

Dans un premier temps, nous avons choisi d'utiliser les ressources observables issues de Badger 2, car les ressources observables définies par les experts du domaine, y compris celles saisies à l'aide de BabyWatch sont, en général, absentes des systèmes d'information de santé.

7.2.4.1 Validation par « action objective »

La validation par « action objective » a pour but d'assurer les vrais-positifs et d'éviter les faux-négatifs. Les faux-négatifs ne sont pas détectés par le système d'alertes, ils sont donc considérés comme faux. Nous savons qu'il s'agit d'un faux-positif parce que nous connaissons d'avance qu'une alerte doit être détectée.

Nous évaluons la situation d'alerte sur les trois ou quatre scénarii dont l'« action objective » est la *finalité* de la situation d'alerte (Tableau 43). La validation est réussie seulement si l'évaluation de la situation d'alerte résulte de la détection d'une ou plusieurs alertes dont les indices de qualité égalent ou dépassent la qualité minimale fixée à 0.52.

7.2.4.2 Validation par scénarii « ne rien faire »

La validation par scenarii « ne rien faire » cherche à assurer les vrais-négatifs et à éviter les faux-positifs. Ceci est possible parce que nous connaissons d'avance qu'aucune alerte ne doit être détectée. « Ne rien faire » n'est pas une action mais le scenario de test de validation de BT-45 dont l'« action objective », soit la meilleure *action-à-faire,* est « ne rien faire » (Tableau 43). La validation est réussie seulement si l'évaluation de la situation d'alerte ne détecte pas d'alertes, ou détecte des alertes dont les indices de qualité ne dépassent pas la qualité minimale fixée à 0.52.

Pour les situations d'alerte Sit-A ou Sit-C, nous n'avons pas réussi à éviter tous les faux-positifs (Tableau 45).

7.2.4.3 Raffinement des situations d'alerte

Le raffinement d'une situation d'alerte est la modification de connaissances préparées concernant les ressources observables, les conditions d'activation et la détection de tendances. Une situation d'alerte est raffinée si le résultat de la validation par « action objective » ou par scénarii « ne rien faire » n'est pas celui qui est attendu. Une nouvelle validation par « action objective » et par scénarii « ne rien faire » a lieu après chaque raffinement de la situation d'alerte.

Les modifications faites lors de chaque raffinement sont listées dans le Tableau 44.L'adaptation d'une situation d'alerte est faite en temps réel et s'évalue sur de nouvelles données.

Tableau 44. Changements dans les situations d'alerte

Changement	Nombre de changements							
	Sit-A	Sit-B	Sit-C	Sit-D	Sit-E	Sit-F	Sit-G	TOTAL
Changer l'importance des conditions d'activation	2	1	16	7	4	2	2	34
Changer la fonction d'appartenance associée à une valeur linguistique	15	2	0	4	1	0	1	28
Ajouter une condition d'activation	4	2	4	3	2	3	1	19
Effacer une condition d'activation	1	2	4	2	4	0	5	18
Changer le nombre de segments à trouver dans la série chronologique	5	3	3	1	2	0	2	16
Changer la période pour la détection de tendances	1	0	3	2	1	1	1	9
Modifier la condition d'activation	0	2	5	0	2	0	0	9
Annuler un changement	3	0	0	0	0	0	0	3
Plusieurs changements en même temps	4	1	2	3	5	1	1	17

Le Tableau 45 résume le nombre des fois que nous avons raffiné chaque situation d'alerte afin de réduire au maximum les faux-positifs. Les faux-négatifs ont été évités complétement dans tous les cas.

Tableau 45. Nombre de raffinements par scénario

Situation d'alerte	Action associée	Nombre de raffinements	Nombre de faux-positifs après le dernier raffinement
Sit-A	Act-A	35	1

Situation d'alerte	Action associée	Nombre de raffinements	Nombre de faux-positifs après le dernier raffinement
Sit-B	Act-B	13	0
Sit-C	Act-C	37	2
Sit-D	Act-D	22	0
Sit-E	Act-E	21	0
Sit-F	Act-F	7	0
Sit-G	Act-G	13	0

7.2.4.4 Test à l'échelle et raffinement OVNI

Le test à l'échelle d'une situation d'alerte consiste à l'évaluer sur tous les scénarii de test de validation de BT-45, soit 26 évaluations par situation d'alerte.

A différence du raffinement des situations d'alerte de l'étape précédente, le raffinement OVNI (Objective, Valide, Neutre, Inappropriée) prend en compte les alertes détectées et non détectées lors de l'évaluation d'une situation d'alerte dont la finalité est une « action inappropriée » dans le scénario testé (Tableau 43). Ce nouveau facteur nous mène à redéfinir ce que nous considérons comme des faux-positifs. Ainsi proposer à l'utilisateur de faire une action qui est inappropriée représente un faux-positif. Le raffinement OVNI réduit les faux-positifs. Un faux-positif d'une situation d'alerte est une alerte détectée et l'action objective du scénario est « ne rien faire », ou la finalité de la situation d'alerte est une « action inappropriée » dans le scénario testé.

Exemple

A partir du Tableau 43, supposons que la situation d'alerte Sit-E associée à l'action Act-E a détecté au moins une alerte lors de l'évaluation dans les Sc-7 et Sc-8. Dans le premier cas, l'action Act-E est « inappropriée ». Dans le deuxième cas, l'action Act-E est « valide » mais l'action objective est « ne rien faire ». Les deux alertes seront donc considérées comme étant des faux-positifs. Un vrai-négatif est la non détection d'alertes lorsque l'action est inappropriée ou l'action objective du scénario est « ne rien faire ».

7.2.5 Résultats

Nous avons calculé la précision et l'exactitude puis le temps d'évaluation des situations d'alerte.

7.2.5.1 Précision et exactitude

Nous nous sommes focalisés sur deux mesures statistiques, la précision et l'exactitude. La précision exprime le fait que les alertes sont correctement détectées c'est-à-dire qu'une alerte était attendue et le système l'a détectée. L'exactitude exprime que les alertes ont été bien détectées et que des alertes n'ont pas été manquées. L'encart (27) présente les formules de calcul des deux indicateurs.

$$Précision = \frac{nombre\ de\ vrais\text{-}positifs}{nombre\ d'alertes\ détectées}$$

$$Exactitude = \frac{nombre\ de\ vrais\text{-}positifs + nombre\ de\ vrais\text{-}négatifs}{Nombre\ d'évaluations} \quad (27)$$

Nous avons défini trois critères afin d'obtenir des mesures statistiques pour la classification binaire de l'alerte comme étant vrai ou fausse : O, OV, et OVN. Le critère O considère que des alertes doivent se détecter seulement si l'action associée à la situation d'alerte est l'action objective du scénario. Autrement, des alertes ne doivent pas être détectées. Le critère OV, considère que des alertes doivent être détectées seulement si l'action associée à la situation d'alerte est l'action objective ou une action valide du scénario. Finalement, dans le critère OVN, des alertes doivent être détectées seulement si l'action associée à la situation d'alerte est l'action objective, une action valide, ou une action neutre pour le scénario testé. Si l'action est inappropriée, aucune alerte ne doit être détectée. Le Tableau 46 montre le résultat attendu lors de l'évaluation de la situation d'alerte associée à l'action Act-A pour chacune des catégories précédentes. La précision et l'exactitude peuvent être calculées parce que le résultat attendu est connu à l'avance.

Tableau 46. Résultats attendus O, OV, OVN

Id scénario	Action	Résultat attendu		
	A	O	OV	OVN
1	O	**1**	**1**	**1**
2	V	0	**1**	**1**
3	I	0	0	0
4	V	0	**1**	**1**
5	I	0	0	0
6	N	0	0	**1**
7	V	0	**1**	**1**
8	N	0	0	**1**
9	O	**1**	**1**	**1**
10	V	0	**1**	**1**
11	I	0	0	0
12	N	0	0	**1**
13	I	0	0	0
14	V	0	**1**	**1**
15	V	0	**1**	**1**
16	N	0	0	**1**
17	O	**1**	**1**	**1**
18	V	0	**1**	**1**
19	I	0	0	0
20	V	0	**1**	**1**
21	I	0	0	0

Id scénario	Action	Résultat attendu		
	A	O	OV	OVN
22	V	0	**1**	**1**
23	V	0	**1**	**1**
24	I	0	0	0
26	V	0	**1**	**1**

Pour considérer les alertes comme telles, nous avons fait varier la qualité minimale exigée pour l'indice d'applicabilité à 0.5, 0.52, 0.55, 0.6, et 0.8. Finalement, nous avons aussi défini la qualité minimale ajustée à chacune des situations d'alerte. Le Tableau 47 résume les vrais-positifs, les faux-positifs, les faux-négatifs, et les vrais-négatifs pour chacune des catégories O, OV, OVN en variant les filtres pour l'indice d'applicabilité. Un vrai-positif est une alerte détectée avec une qualité au-dessus de la qualité minimale et une alerte était attendue. Un faux-positif est une alerte détectée avec une qualité au-dessus de la qualité minimale et aucune alerte n'aurait dû être détectée. Un vrai-négatif est la non-détection d'une alerte ou la détection d'une alerte avec une qualité au-dessous de la qualité minimale et aucune alerte n'était attendue. Enfin, un faux-négatif est la non-détection d'une alerte, ou la détection d'une alerte avec une qualité au-dessous de la qualité minimale et une alerte était attendue. La précision et l'exactitude dans le Tableau 47 sont calculées avec les résultats des 171 évaluations.

Il n'est pas évident de considérer une alerte qui suggère une « action valide » comme étant une vraie ou une fausse alerte. Le Tableau 48 résume la précision et l'exactitude en utilisant seulement les scénarii dont l'action associée à la situation d'alerte est l'« action objective » ou une « action inappropriée », catégorie O-I. La précision et l'exactitude dans le Tableau 48 sont calculées avec les résultats des 94 évaluations.

Tableau 47. Calcul de la précision et l'exactitude avec les catégories OVNI

	Filtre IAA	Vrais-positifs	Faux-positifs	Faux-négatifs	Vrais-négatifs	Précision	Exactitude
O	0.5	21	41	0	109	34%	76%
	0.52	20	32	1	118	38%	81%
	0.55	19	29	2	121	40%	82%
	0.6	19	26	2	124	42%	84%
	0.8	12	20	9	130	38%	83%
	Ajusté	21	30	0	120	41%	82%
OV	0.5	36	26	15	94	58%	76%
	0.52	34	18	17	102	65%	80%
	0.55	32	16	19	104	67%	80%
	0.6	31	14	20	106	69%	80%
	0.8	23	9	28	111	72%	78%

	Filtre IAA	Vrais-positifs	Faux-positifs	Faux-négatifs	Vrais-négatifs	Précision	Exactitude
	Ajusté	35	16	16	104	69%	81%
OVN	0.5	48	14	50	59	77%	63%
	0.52	45	7	53	66	87%	65%
	0.55	42	6	56	67	88%	64%
	0.6	41	4	57	69	91%	64%
	0.8	32	0	66	73	100%	61%
	Ajusté	45	6	53	67	88%	65%

Tableau 48. Calcul de la précision et l'exactitude avec O-I

Filtre IAA	Vrais-positifs	Faux-positifs	Faux-négatifs	Vrais-négatifs	Précision	Exactitude
0.5	21	14	0	59	60%	85%
0.52	20	7	1	66	74%	91%
0.55	19	6	2	67	76%	91%
0.6	19	4	2	69	83%	94%
0.8	12	0	9	73	100%	90%
Ajusté	21	6	0	67	78%	94%

7.2.5.2 Temps d'évaluation

Le Tableau 49 montre le nombre final de conditions d'activation par situation d'alerte définie et le Tableau 50 le temps d'exécution de chaque scénario sur une moyenne de 28 patients. Au total, 171 évaluations ont pris 28.4 secondes. Le temps d'évaluation est corrélé au nombre de conditions d'activation qui doivent être satisfaites. Le temps d'évaluation des situations Sit-E et Sit-F est supérieur aux autres puisqu'elles sont définies par un nombre plus important de conditions d'activation.

Tableau 49. Conditions d'activation par situation d'alerte

Situation d'alerte	Nombre de conditions d'activation	Exprimant l'état	Exprimant la tendance
Sit-A	2	1	1
Sit-B	2	1	1
Sit-C	4	3	1
Sit-D	3	1	2
Sit-E	8	3	5
Sit-F	6	2	4
Sit-G	1	1	0

Tableau 50. Temps d'exécution

Situation d'alerte	Temps d'exécution (secondes)
Sit-A	2.2
Sit-B	2.3
Sit-C	3.4
Sit-D	4.0
Sit-E	9.2
Sit-F	6.5
Sit-G	0.8
TOTAL	28.4

7.2.6 Synthèse

Les situations d'alerte ont été adaptées par un utilisateur qui n'est pas expert dans le domaine de la santé et qui a réussi, juste en regardant des graphiques et en lissant des résumés, à définir les situations d'alerte. Dans un premier temps, l'adaptation des situations d'alerte a été itérative, suivie des évaluations sur six ou sept scénarii jusqu'à l'obtention de résultats satisfaisants. Ensuite, l'adaptation a été guidée par les résultats obtenus lors de l'évaluation des situations d'alerte sur les vingt-six scénarii. Nous avons calculé la précision et l'exactitude des situations d'alerte selon quatre catégorisations que nous avons définies comme O, OV, OVN, et O-I.

Le filtrage par indice de qualité (IAA) permet d'adapter la précision et l'exactitude des situations d'alerte. En général, il est possible d'améliorer la précision et réduire l'exactitude en augmentant la qualité minimale exigée. Cela signifie que les alertes détectées lors de l'évaluation d'une situation d'alerte sont de vraies alertes mais que certaines ont pu être manquées. Inversement, diminuer la qualité minimale exigée réduit la précision et augmente l'exactitude. Cela signifie que les alertes détectées lors de l'évaluation d'une situation d'alerte peuvent être de fausses alertes mais peu d'alertes ont pu être manquées.

Les adaptations d'une situation d'alerte cherchent à augmenter la précision et l'exactitude. Le filtrage par qualité est adaptable en temps réel, il peut prioriser l'un par rapport à l'autre à un instant précis. L'utilisation d'autres observations, comme celles de BabyWatch pourraient servir à l'amélioration de la définition des situations d'alerte et donc à la réduction des faux-positifs et faux-négatifs. Les résultats obtenus sont, en l'état, assez encourageants.

7.3 Flexibilité et adaptabilité

TEMPAS a été présenté comme un système d'alertes adaptable au contexte facilement utilisable par un utilisateur non informaticien. Nous cherchons à valider la flexibilité du système qui permet l'adaptation à différents contextes d'application en temps réel. Cette validation est faite par des utilisateurs experts dans le domaine de la dialyse à domicile. Nous avons défini un protocole de validation à base de tests de l'application par des utilisateurs. Le protocole est divisé globalement en deux parties. L'objectif de la première partie a été de définir des scénarii de tests interprétables par les utilisateurs experts. La deuxième partie a consisté à récupérer les réactions des utilisateurs pour valider l'application.

7.3.1 Participants de l'expérimentation

Nous avions besoin de participants avec une expertise dans un domaine spécifique. Nous sommes entrés en contact avec Calydial, un établissement de santé spécialisé dans le traitement de patients atteints de maladies rénales chroniques. Deux rendez-vous ont eu lieu : un premier pour définir des situations d'alerte que nous utilisons ensuite pour créer des scénarii de tests, et un deuxième pour que les utilisateurs testent TEMPAS.

Les participants ont été regroupés en deux groupes. Le groupe « définition » et le groupe « expérimentation ». Le premier rendez-vous a eu lieu avec le groupe « définition » formé par le responsable informatique et le responsable télémédecine. Le deuxième rendez-vous a eu lieu avec le groupe « expérimentation » formé par les deux responsables précédents et une infirmière responsable du suivi des patients à domicile.

7.3.2 Définition des situations d'alerte

Nous avons fixé un premier rendez-vous avec le groupe « définition » afin de définir ensemble des situations d'alerte interprétables par tous les participants. Nous leur avons montré les fonctionnalités de TEMPAS et ils ont défini ce qui les intéresse : une situation d'alerte largement connue dans le domaine de la dialyse à domicile et une autre qui répond à leurs besoins propres.

7.3.2.1 Système d'information de Calydial

Calydial pratique la télémédecine. Les patients domiciliés chez eux, ou une infirmière libérale, envoient des informations à Calydial en utilisant une tablette ou un téléphone intelligent. Une application chargée de récupérer les données génère des alertes si les valeurs observées dépassent des seuils prédéfinis, surtout dans le cas du poids ou de la tension systolique. Les informations sont envoyées en moyenne une fois par jour et ceci une ou cinq fois par semaine.

7.3.2.2 Présentation de TEMPAS

Nous nous sommes servis d'une vidéo pour montrer aux utilisateurs quels types d'alertes pouvaient être définis avec TEMPAS. La situation d'alerte paramétrée dans la

vidéo détectait des « risques d'infection chez les bébés » à partir de deux ressources observables, la « température » et la « tension ». Les spectateurs de la vidéo voient comment définir les valeurs linguistiques des ressources observables et comprennent comment l'ambigüité est gérée. La vidéo montre aussi comment contextualiser la situation d'alerte avec l'âge. Les valeurs linguistiques de l'« âge » modifient celles de la « tension » : une « tension » de 150 mm Hg chez un nouveau-né est une tension normale, tandis que chez un nourrisson c'est une « hypertension ». Finalement, la vidéo montre les alertes détectées sur deux patients, un nouveau-né et un nourrisson, avant et après la contextualisation. La vidéo seule a suffi pour que les participants puissent par eux-mêmes définir quatre situations d'alerte qui répondent à leurs besoins immédiats.

7.3.2.3 Situations d'alerte

Les ressources observables ainsi que les situations d'alerte ont été présentées en détail dans le chapitre 5. Il s'agit de la « sous-dialyse », la « sur-dialyse », l'« émergence répétitive », et l'« émergence impérative ».

Il y a « sur-dialyse » si le pourcentage de changement de poids (PCP) est une « perte » et la tension systolique une « hypotension ». « sous-dialyse » si PCP est une « prise » et la tension systolique une « hypertension ». « émergence répétitive » si l'émergence douteuse se trouve dans l'*état* « peut attendre » et se comporte comme une *tendance* « stable ». Enfin, il y a l'« émergence impérative », si l'émergence douteuse « est impérative ».

7.3.2.4 Constats lors du premier rendez-vous

Lorsque nous avons parlé des systèmes d'alertes, les participants ont pensé tout de suite à des systèmes experts. Nous leur avons présenté le principe de TEMPAS où c'est l'utilisateur qui définit ce qui l'intéresse. Les situations d'alerte définies sont la « propriété intellectuelle » de l'utilisateur qui les a définies.

Après la vidéo, le groupe « définition » a montré un intérêt particulier pour les alertes multi-paramètres, la possibilité de réduire les faux-positifs, et les indices de qualité afin de prioriser les alertes. Ils ont aussi montré un intérêt sur les temps de validité et d'expiration des ressources observables ainsi que le temps de validité des alertes, dimensions, qui d'après eux, n'étaient pas prises en compte dans d'autres systèmes.

7.3.3 Création des scénarii de tests

Nous avons défini des scénarii de tests avec des situations d'alerte définies par les participants. Chaque test cherche à valider nos hypothèses concernant les qualités de TEMPAS.

7.3.3.1 Hypothèses à valider

Nous avons défini une série de tests qui valident une partie des hypothèses ci-dessous.

Expressivité des situations d'alerte

D'abord, nous nous sommes intéressés à la validation de l'expressivité des situations d'alerte. L'utilisateur comprend ce qu'est une ressource observable, la transformation de valeurs quantitatives en valeurs linguistiques pour atténuer l'ambigüité, la définition des situations d'alerte à partir des valeurs linguistiques, et la signification des indices de qualité associés aux alertes détectées.

Flexibilité du système

Ensuite, nous voulions valider la *flexibilité* de TEMPAS qui permet l'*adaptabilité* en temps réel et qui prend différentes formes :

- La *contextualisation* qui améliore les alertes détectées résultant de l'évaluation des situations d'alerte. L'amélioration s'exprime par des valeurs des indices de qualité plus en accord avec l'alerte ;
- L'*utilité* des valeurs linguistiques lors de la construction de conditions d'activation pour travailler avec des valeurs ambiguës et pour exprimer ce qui caractérise les situations d'alerte pour un utilisateur
- La prise en compte de l'aspect temporel

Utilité des indices de qualité

L'utilisateur comprend la signification des indices de qualité : ils ont deux utilités importantes à terme, améliorer la précision et l'exactitude, et contrôler la notification d'alertes. Prioriser les alertes a une utilité à l'instant t.

La précision et l'exactitude prennent en compte les vrais-positifs, les vrais-négatifs, les faux-positifs et les faux-négatifs. Si plus d'alertes que prévu sont notifiées ou moins que prévu, les premières actions sont d'augmenter ou réduire les seuils de filtrage associés à chaque situation d'alerte. D'autre part, il est nécessaire de modifier les plages de valeurs quantitatives représentées par les valeurs linguistiques ou les temps de validité et d'expiration des ressources observables. Enfin, il faut modifier la situation d'alerte elle-même en modifiant, supprimant, ou ajoutant des conditions d'activation.

La notification explicite est contrôlée par le cycle de vie de l'alerte défini à l'aide de quatre scénarii enrichis avec les indices de qualité (par exemple en notifiant explicitement seulement si l'indice d'applicabilité ou l'indice de confiance augmente).

7.3.3.2 Structure des tests

Nous avons défini onze tests listés dans le Tableau 51 avec des finalités variées. Tous les participants font tous les tests.

Tableau 51. Tests des utilisateurs

Test	Finalité	Situation d'alerte	Séquences d'activité Utilisateur	Données	Alertes détectées
1A	Expressivité des situations d'alerte – Réduction des faux-positifs	Sur-dialyse		« M. Albert » Tension : 150 Poids : 62.4	Alerte explicite (IAA = 0.5, ICA = 0.5)
			Modifier le calcul des valeurs linguistiques pour le PCP		Aucune alerte notifiée
1B			Modifier les seuils de filtrage		Alerte explicite (IAA = 0.23, ICA = 0.5)
2A	Ajuster la temporalité des variables	Sous-dialyse		« Mme. Brigitte » Tension : 157 Poids : 60.6	Alerte explicite (IAA = 1, ICA = 1)
2B			Modification du temps de validité et du temps d'expiration		Alerte explicite (IAA = 1, ICA = 0.5)
3A	Réduire la sur-notification			« Mme. Brigitte » Tension : 161 Poids : 58.9	Alerte explicite (IAA = 0.5, ICA = 0.5)
			Modifier le cycle de vie de l'alerte		
3B				« Mme. Brigitte » Tension : 132 Poids : 60.2	Alerte implicite (IAA = 0.5, ICA = 0.5)

Test	Finalité	Situation d'alerte	Séquences d'activité Utilisateur	Données	Alertes détectées
4A	Priorisation des alertes			« M. Albert » Tension : 153 Poids : 62 « M. Charles » Tension : 136 Poids : 106 « Mme. Brigitte » Tension : 153 Poids : 60.2 « M. David » Tension : 157 Poids : 72	« M. Albert » Alerte explicite (IAA = 0.4, ICA = 0.5) « M. Charles » Alerte explicite (IAA = 0.32, ICA = 0.5) « Mme. Brigitte » Alerte explicite (IAA = 0.9, ICA = 1) « M. David » Alerte explicite (IAA = 1, ICA = 1)
			Augmenter le filtrage dans la liste d'alertes		« M. David » Alerte priorisée (IAA = 1, ICA = 1)
5A	Expressivité des situations d'alerte – importance des conditions d'activation	Emergence impérative		« M. David » Ecoulement	« M. David » Alerte explicite (IAA = 1, ICA = 1)
		Emergence répétitive		« Mme. Brigitte » Rougeur	« Mme. Brigitte » Alerte explicite (IAA = 0.5, ICA = 0.5)
5B			Modifier l'importance de la condition d'activation	« Mme. Brigitte » J : Rougeur « Mme. Brigitte » J-1 : Rougeur	Aucune alerte notifiée
5C				« Mme. Brigitte » J : Rougeur « Mme. Brigitte » J-1 : Rougeur « Mme. Brigitte » J-2 : Rougeur	Aucune alerte notifiée

Test	Finalité	Situation d'alerte	Séquences d'activité Utilisateur	Données	Alertes détectées
5D				« Mme. Brigitte » J : Rougeur « Mme. Brigitte » J-1 : Rougeur « Mme. Brigitte » J-2 : Rougeur « Mme. Brigitte » J-3 : Rougeur	« Mme. Brigitte » Alerte explicite (IAA = 1, ICA = 1)

7.3.4 Expérimentation avec les utilisateurs experts

Nous avons défini un protocole d'expérimentation que nous avons appliqué individuellement à chacun des participants. Il est constitué d'une séquence d'activités présentées ci-dessous. L'application du protocole d'application a duré en moyenne 80 minutes par utilisateur.

7.3.4.1 Formation à l'application

Comme s'il s'agit de participants qui ne connaissent pas TEMPAS, il est nécessaire de les former à l'application. La formation commence avec la vidéo de présentation de TEMPAS suivie par une démonstration in vivo de l'application.

7.3.4.2 Appropriation de l'application cliente

Les participants font des tests d'adaptation de la situation d'alerte « infection chez les bébés ». Le but est que le participant s'approprie l'application à laquelle il n'est pas habitué. En parallèle, il a accès à un guide de l'application sous format papier avec des copies d'écran qui expliquent brièvement le fonctionnement des composants TEMPAS dans FSD.

7.3.4.3 Débriefing sur l'IHM

Nous identifions tous les problèmes possibles que le participant peut avoir avec l'application cliente. A aucun moment nous cherchons à valider l'interface graphique ni les interactions possibles avec l'application mais il est indispensable que le participant puisse se l'approprier pour que les réactions ne soient pas liées à l'interface mais à ses fonctionnalités qui valident nos hypothèses.

7.3.4.4 Exécution des tests

Une fois que le participant s'est approprié l'application, nous procédons à l'exécution de la séquence de tests. Un document contient la séquence de tests que

l'utilisateur doit faire. Un discours guidant le participant accompagne l'exécution des tests afin de conserver l'intérêt du participant pendant toute la durée de l'expérimentation.

7.3.4.5 Débriefing des tests

A la fin des tests, nous avons fait un entretien avec les participants et nous avons posé un ensemble de questions pour récupérer leurs réactions sur les fonctionnalités de TEMPAS.

7.3.4.6 Formulaire d'utilisabilité

Les participants ont rempli le formulaire SUS (de l'anglais « System Usability Scale ») contenant 10 questions qui permettent de calculer l'utilisabilité du système (cf. Tableau 54).

7.3.4.7 Mise en place de l'expérimentation

Nous avons utilisé deux machines pour faire l'expérimentation : une machine pour les participants avec l'application cliente FSD (Futura Smart Design) et une autre utilisée comme serveur centralisé dont les données sont simulées en temps réel. Les alertes détectées par TEMPAS ont un caractère *actuel* c'est-à-dire que lors de l'évaluation d'une situation d'alerte, le système utilise les dernières observations connues. De plus, le cycle de vie de l'alerte utilise l'instant de détection des alertes qui est le moment auquel le système commence l'évaluation d'une situation d'alerte. Ces deux raisons nous obligent à simuler les données de patients en temps réel pour les tests concernant le cycle de vie de l'alerte et l'indice de confiance qui prennent en compte l'aspect temporel.

7.3.5 Résultats

Nous avons fait tester TEMPAS par trois participants avec des profils différents : le responsable informatique, le responsable télémédecine et une infirmière qui suit à distance les patients. Le Tableau 52 récapitule les retours des participants après les tests de l'application. Le Tableau 53 liste les difficultés que les utilisateurs ont trouvées. Dans les tableaux, nous employons les phrases exprimées par les utilisateurs lors des tests entre guillemets. Lorsque nous utilisons plusieurs phrases nous les groupons par des sauts de ligne simples.

Tableau 52. Résultats des expérimentations

Hypothèse	Retours	Conclusion
Réduction des faux-positifs	«Certaines alertes qui apportent la même réponse pourraient être gérées ensemble» «Il n'y a pas d'alerte inutiles, il vaut mieux en avoir trop que de ne pas en avoir du tout» «Vu le volume de données que l'on gère, on ne peut pas croiser les valeurs manuellement» «Une tension de 150 déclenche une alerte, mais parfois une tension de 147 devrait la déclencher lorsqu'on la croise avec d'autres valeurs»	TEMPAS permet de créer des situations d'alerte complexes avec des règles simples qui prennent en compte tous les facteurs. Ainsi, si l'alerte est déclenchée par un seul, deux, ou trois facteurs, c'est la même alerte. Le graphe modificateur permet de prendre en compte le contexte du patient, ce qui se traduit par le croisement des informations utiles lors de l'évaluation d'une situation d'alerte. TEMPAS permet donc de réduire les faux-positifs.
Adaptation des situations d'alerte	«Si j'ai beaucoup d'alertes, c'est parce que j'ai mal paramétré l'alerte et il faut la vérifier. C'est ma faute. Le tout c'est de bien la paramétrer» «Je peux l'adapter aux besoins de chaque patient»	Les situations d'alerte doivent être adaptées si les résultats ne sont pas ceux attendus : soit des faux-positifs ou des faux-négatifs. TEMPAS est adaptable par les utilisateurs.

Hypothèse	Retours	Conclusion
Expressivité des situations d'alerte	«Il est possible de prendre en compte le contexte du patient» «Je peux modifier les événements du patient» «Je peux exprimer ce que je veux» «C'est bien de prendre le PCP au lieu du poids» «Ce (les alertes) qui va nous revenir c'est ce qu'on a mis (ce qu'on a paramétré)» «On va pouvoir créer des situations d'alerte complexes» «C'est potentiellement exploitable» «Le système fait le pont entre les systèmes simples et les systèmes experts» «Bien définir les situations d'alerte facilite la délégation de tâches» «Beaucoup de scénarios, beaucoup de cas d'utilisation»	TEMPAS permet de définir des situations d'alerte à partir des valeurs linguistiques qui expriment l'état ou la tendance des variables. Les données utilisées lors de l'évaluation des situations d'alerte sont basiques. Les utilisateurs peuvent exprimer une grande variété de situations d'alerte.
Contrôle de notification	«Si j'ai beaucoup d'alertes notifiées, je jette l'ordinateur par la fenêtre. C'est bien de pouvoir en réduire le nombre» «c'est important de réduire le nombre de notifications»	TEMPAS notifie seulement les alertes qui dépassent une qualité minimale requise. En plus, le cycle de vie permet de réduire la notification explicite. TEMPAS contrôle la notification pour ne pas sur-notifier

Hypothèse	Retours	Conclusion
Priorisation des alertes	«C'est très pratique lorsque j'ai beaucoup d'alertes en rouge (à traiter) et peu de temps» «cela évitera de réfléchir» «La graduation des alertes (indices de qualité) est très pratique» «C'est très bien ça» (x2)	Les indices de qualité servent à prioriser les alertes. Plus un indice est élevé, plus le patient est concerné ou plus récentes sont ses données. Le *filtrage par ratio* permet de prioriser des alertes indifféremment de la situation d'alerte qu'elles instancient. Les indices de qualité sont utiles pour prioriser les alertes
Utilisation des valeurs linguistiques	«Les médecins ne se mettent pas d'accord avec des valeurs quantitatives» «C'est malin d'utiliser des valeurs sémantiques qui prennent en compte l'ambigüité»	TEMPAS utilise de valeurs linguistiques qui prennent en compte l'ambiguïté et qui servent à définir des situations d'alerte. Il est possible de changer le calcul des valeurs linguistiques sans modifier la situation d'alerte afin que les patients soient plus, ou moins, concernés par l'alerte. L'utilisation des valeurs linguistiques facilite la définition des situations d'alerte
Prise en compte de l'aspect temporel	«Judicieux et complet» «C'est pertinent» «C'est important»	TEMPAS utilise l'aspect temporel représenté globalement par le temps de validité et d'expiration des ressources observables utilisées pour calculer l'indice de confiance de l'alerte, et le temps de validité de l'alerte utilisé pour le contrôle de la notification. Prendre en compte le temps améliore la confiance des alertes.

Hypothèse	Retours	Conclusion
Les informations associées à l'alerte facilitent le travail des utilisateurs	«Si systématiquement pour chaque patient, on avait sa dernière consultation où l'on a tous les éléments» «Cela nous évite une perte de temps» «Je n'ai pas besoin d'aller chercher ailleurs» «Les infirmières (libérales) changent et ne connaissent pas le passé du patient»	L'équation de l'alerte contient toutes les ressources observables qui servent à construire et à évaluer des conditions d'activation définissant une situation d'alerte. L'équation contient aussi toutes les ressources dont la dernière valeur connue et utilisée afin de construire le message de l'alerte. Ces ressources n'interférent pas dans le résultat des alertes détectées. TEMPAS permet de créer des messages d'alerte qui s'adaptent aux besoins des utilisateurs

Tableau 53. Difficultés de TEMPAS

Difficulté	Réflexion participants	Solution TEMPAS
Traiter les alertes	«après avoir détecté une alerte, que se passe-t'il?»	A toute situation d'alerte, lui est associé un message à afficher. Celui-ci peut être un protocole à suivre. C'est à l'utilisateur de le définir. Enfin, ce message a pour objectif d'expliquer à l'utilisateur les décisions à prendre. TEMPAS est indépendant du domaine d'application. C'est au système qui intègre TEMPAS de choisir comment les alertes doivent être gérées puisque la finalité des situations d'alerte est celle que l'utilisateur veut lui donner. Dans FSD, on peut récupérer et afficher les données du patient sur qui une alerte a été détectée : ses caractéristiques, ses constantes médicales, etc.

Difficulté	Réflexion participants	Solution TEMPAS
Appropriation des indices	«Comment s'approprier des indices, on voit bien à quoi ça sert, mais comment se les approprier ?»	TEMPAS utilise deux indices de qualité qui expriment dans quelle mesure un patient est concerné par l'alerte, et la confiance que l'utilisateur peut faire aux données utilisées par rapport à sa fraicheur. Les indices sont utilisés pour filtrer les alertes avec des mauvaises qualités, prioriser les alertes, contrôler la notification explicite des alertes. Etant donné les fonctionnalités des indices, pour se les approprier il faut les utiliser, créer de situations d'alerte est analyser les résultats dans le cas des situations d'alerte complexes.
Alertes et incidents	«Il est nécessaire de différencier les alertes et les incidents»	TEMPAS travaille avec la notion de situation d'alerte qui peut être simple ou complexe. Dans les autres systèmes, les incidents sont des alertes complexes déterminées avec des alertes simples.
Sémantique	«Il y a beaucoup de concepts nouveaux» «Il faut trouver un glossaire commun» «il faut s'habituer au vocabulaire» (x2)	TEMPAS utilise une sémantique que nous avons spécifiée afin de le rendre indépendant du domaine d'application
Situations d'alerte dans le domaine d'application	«le tout c'est de tout bien configurer» «à voir dans la vraie vie»	Le paramétrage des situations d'alerte dépend des utilisateurs. Ceux-ci pourront améliorer et adapter les alertes à fur et à mesure qu'ils maîtrisent mieux le système.

Difficulté	Réflexion participants	Solution TEMPAS
Besoin de formation	«A terme, je pourrai configurer des situations d'alerte simples tout seul» «J'aurai besoin d'une formation d'à peu près 3 jours pour pouvoir me débrouiller par moi-même ensuite, ça ira» «Je pense qu'un médecin aurait besoin d'un accompagnement pour la définition des situations d'alerte» «Au début il faut bien quelqu'un pour nous former» «Pour moi, le paramétrage est complexe pour un professionnel de santé lambda» «Le professionnel de santé n'a pas de temps à dédier pour la configuration des situations d'alerte. Ils sont habitués aux vieilles méthodes»	TEMPAS utilise des nombreux concepts concernant la préparation de connaissances et la définition des situations d'alerte, ainsi que des fonctionnalités des indices de qualité.
Couleurs pour priorisation des alertes	«Il serait pertinent d'ajouter un dégradé de jaune à rouge par rapport à l'IAA de l'alerte»	Visuellement, dans FSD quatre couleurs servent à exprimer l'état de l'alerte : « nouvelle », « vue », « traitée », et «périmée ». Les indices de qualité sont présents en valeur numérique.

Difficulté	Réflexion participants	Solution TEMPAS
Rôle de concepteur d'alertes	« Je ne vois pas un médecin qui n'est pas formé paramétrer une alerte »	TEMPAS a été intégré à FSD. Les utilisateurs de FSD ont des rôles et les rôles sont des vues de son espace de travail avec des composants spécifiques. Par exemple, le rôle « concepteur d'alerte » est l'unique rôle à avoir une vue avec accès aux composants permettant de configurer les alertes. Ceci a été expliqué aux utilisateurs lorsqu'ils étaient intéressés. Toutefois, nous avons développé TEMPAS de sorte à maximiser le nombre d'utilisateurs concepteurs d'alertes.
Adaptation sur un patient spécifique	« Si je veux adapter le seuil pour un patient qui ne récupère pas à la même vitesse que les autres, me faut-il créer une nouvelle situation d'alerte?»	TEMPAS prend en compte le contexte du patient. Toutefois celui-ci est exprimé dans l'équation de l'alerte, à l'aide d'une ressource observable sans changer la situation d'alerte. Par exemple, en ajoutant une caractéristique du patient « évolution lente » qui modifie le calcul des valeurs linguistiques des autres ressources observables. Ceci n'affecte pas la situation d'alerte sur les autres patients concernés puisque leur évolution n'est pas lente ou c'est une donnée inconnue, manquante, qui n'est donc pas prise en compte lors de l'évaluation des situations d'alerte parce qu'elle n'a pas changé.

Le Tableau 54 montre le formulaire utilisé pour calculer l'utilisabilité du système ainsi que les réponses des participants. *« Vous donnerez votre opinion sur chacune des propositions sur l'échelle d'accord : de pas du tout d'accord à tout à fait d'accord, les notes 2, 3 et 4 servent à moduler votre réponse »* (Brooke, 1996). Nous pouvons

remarquer que les utilisateurs ont trouvé que le système pouvait répondre à des besoins immédiats concernant les alertes, qu'il est nécessaire d'être formé afin de pouvoir s'en servir, ou d'être accompagné par quelqu'un formé lors de la définition des situations d'alerte.

Tableau 54. Formulaire SUS

	Pas du tout d'accord	2	3	4	Tout à fait d'accord
1 - Je souhaiterais utiliser ce système fréquemment					3
2 - Je pensais que ce système était inutilement complexe	2	1			
3 - Je pensais que le système était facile à utiliser		1		1	1
4 - Je pense avoir besoin de l'aide d'un technicien pour utiliser ce système			2		1
5 - J'estime que les différentes fonctions du système sont bien intégrées					3
6 - Je pense qu'il y a trop d'incohérences dans ce système	3				
7 - J'imagine que la plupart des gens peuvent apprendre très rapidement à se servir de ce système			1	1	1
8 - Le système est très "encombrant" à utiliser	3				
9 - J'ai énormément confiance en ce système				3	
10 - J'ai dû apprendre beaucoup de choses avant de pouvoir utiliser ce système	1	2			

7.3.6 Synthèse

Nous avons utilisé TEMPAS pour définir deux situations d'alerte dans le domaine de la dialyse à domicile. Nous avons fait des tests pour valider globalement l'expressivité des situations d'alerte, la flexibilité du système afin de l'adapter dans n'importe quel domaine d'application et pouvoir s'en servir, et finalement l'utilité des indices de qualité lors de l'exploitation des alertes détectées. Néanmoins, TEMPAS emploie un grand nombre de concepts et de fonctionnalités qui le rendent indépendant du domaine d'application mais qui rendent difficile son appropriation immédiate. Nous avons fait des tests de validation de quelques fonctionnalités du système. 90 minutes ne suffisent pas pour que l'utilisateur s'approprie l'outil et pour lui montrer les alertes détectées dans des scénarii spécifiques.

Les participants ont vu le potentiel de TEMPAS lorsque des alertes autres que celles connues du domaine de la dialyse à domicile ont été paramétrées, comme la

rougeur répétitive. TEMPAS est un système simple qui permet de définir des situations d'alerte complexes. Toutefois, le besoin d'une formation ou un accompagnement sont nécessaires lors de la définition des situations d'alerte plus élaborées. Il est possible d'exprimer une grande variété des situations d'alerte, mais il est indispensable de trouver le bon paramétrage (c'est le cas pour tous les systèmes d'alertes). En revanche, une fois les alertes paramétrées, les utilisateurs gagnent du temps dans la gestion d'informations, non seulement grâce aux informations que l'alerte leur fournit, mais parce que les alertes qui lui sont notifiées sont précisément celles qui nécessitent leur attention et qui doivent être traitées.

Dans le domaine médical, les médecins n'ont pas beaucoup du temps à consacrer à la configuration des alertes. Cependant, il est nécessaire que le professionnel de santé réfléchisse sur sa pratique. Les systèmes d'alertes fournissent des meilleures informations des patients, mais aussi peuvent aider le patient à avoir une meilleure compréhension de ses pathologies.

L'IHM de FSD a facilité l'acceptation de TEMPAS comme système d'alertes. FSD vise à réduire le nombre d'actions (clicks) que l'utilisateur doit faire pour trouver l'information dont il a besoin.

8 Conclusion et perspectives

8.1 Conclusion 211
8.2 Perspectives 212

8.1 Conclusion

Nous avons abordé dans cette thèse la problématique des systèmes d'alertes. Nous avons proposé un système d'alertes personnalisable indépendant du domaine d'application. Le système d'alertes est adaptable au contexte grâce à la participation des utilisateurs qui définissent des situations d'alerte en utilisant un ensemble de conditions d'activation construites avec des valeurs linguistiques. Deux indices de qualité sont associés aux alertes détectées. L'indice d'applicabilité exprime dans quelle mesure une entité est concernée par l'alerte. Il se calcule à partir de l'ambigüité avec laquelle la valeur linguistique représente la valeur observée et l'importance respective de la condition d'activation qui est satisfaite. Le deuxième indice exprime la confiance que l'utilisateur peut faire à la fraîcheur des données utilisées lors de l'évaluation de la situation d'alerte. Les deux indices sont utilisés dans la politique de notification chargée de réduire le nombre d'alertes notifiées de forme implicite et explicite aux utilisateurs. Les deux indices de qualité associés aux alertes détectées servent aux utilisateurs pour reconnaître les bons et les mauvais paramétrages des situations d'alerte, non seulement par la notification ou par la non-notification d'alertes, mais aussi pour la valeur numérique des indices associés à l'alerte. Le calcul des indices est fait par rapport aux connaissances préparées par les utilisateurs ce qui le rend adaptable au domaine d'application.

Nous proposons une solution intermédiaire entre les systèmes d'alertes très simples qui génèrent beaucoup d'alertes dont des faux-positifs et des alertes inutiles, et des systèmes experts très pointus dans des situations d'alerte identifiées à l'avance. Nous avons vu la pertinence d'une telle approche dans le domaine de la télémédecine. TEMPAS ne fournit pas d'alertes pré-paramétrées concernant un domaine spécifique, mais permet de définir des alertes compréhensibles par les utilisateurs. Ceux-ci peuvent adapter les situations d'alerte à leur contexte d'utilisation et définir de nouvelles situations d'alerte lorsque des situations inattendues ont été identifiées. Les situations d'alerte complexes sont définies à partir de règles simples.

Nous avons intégré TEMPAS au système de gestion d'informations en santé *Futura Smart Desing* (FSD). Des composants fonctionnels ont été développés afin que les utilisateurs puissent préparer des connaissances et définir des situations d'alerte. L'objectif est que l'utilisateur estime que la fonctionnalité des alertes fait partie du système d'information lui-même. Ainsi, on avantage l'appropriation de TEMPAS par l'utilisateur. La plupart des informations nécessaires à la définition des situations d'alerte qui intéressent les utilisateurs existent déjà dans le système d'information qu'ils utilisent. Le problème est souvent de les récupérer automatiquement. Nous

avons défini une intégration au niveau des données qui, grâce à l'utilisation des vues permet d'exploiter les informations automatiquement par TEMPAS. Ainsi, des informations contextuelles externes comme « la température du patient Dupont » ou internes comme « le rôle du médecin traitant du patient Dupont » peuvent être utilisées lors de la définition des situations d'alerte ou dans la construction du message de l'alerte.

Les résultats des expérimentations sont très encourageants. TEMPAS a permis de définir des situations d'alerte identifiées dans le domaine de la dialyse à domicile ainsi que des situations d'alerte concernant la gestion de l'information de l'établissement qui permettent de déléguer des responsabilités.

8.2 Perspectives

Etant donné l'état avancé de l'intégration de TEMPAS avec un système d'information de santé, les perspectives concernent entre autres, l'intégration avec d'autres systèmes d'information et domaines d'application.

8.2.1 Exploitation des observations

Lors de l'évaluation d'une situation d'alerte, le système utilise la dernière observation connue pour chaque ressource observable. Une évolution possible, serait d'exploiter toutes les observations de la ressource à condition que l'indice de confiance de l'observation soit supérieur à zéro. Une telle approche détectera plus d'alertes et pourra réduire des faux-négatifs causés par des conditions d'activation temporelles ou des valeurs observées satisfaisant des conditions d'activation intrinsèques.

TEMPAS pourrait donner le choix à l'utilisateur de privilégier les faux-positifs sur les faux-négatifs ou vice-versa puisque le poids de leur impact dépend du domaine d'application.

8.2.2 Qualité des données

Nous utilisons la fraicheur des données afin de calculer l'indice de confiance des alertes et nous les supposons fiables indépendamment de la source. Les travaux qui utilisent la confiance comme indicateur de qualité s'appuient sur la confiance fait au dispositif de capture, au dispositif de détection, ou à la personne qui fournit la donnée. C'est un indicateur qui ne se calcule pas, qui est dépendant du domaine d'application et que nous n'utilisons pas lors de l'évaluation des situations d'alerte.

Il serait intéressant de prendre en compte la fiabilité que l'utilisateur accorde à la source de données. Une première approche serait de l'exprimer dans le message de l'alerte. Une deuxième plus approfondie serait de créer un troisième indice de qualité concernant cette fiabilité à la source où toute source est fiable sauf lorsque l'utilisateur exprime le contraire. L'intégration au niveau de données serait affectée puisque une même base de données pourrait être peuplée par différentes sources dont on ne

garderait pas la trace. Si la confiance attribuée à la source est subjective, il est nécessaire de prendre ce facteur en compte avant notification à l'utilisateur.

Exemple

Dans le modèle de données la température de M. Dupont saisie par une infirmière ou par un capteur de température est représentée de la même façon.
Le « Dr. Jean » fait confiance aux données saisies par l'infirmière « Marion ». Ce n'est pas le cas pour le « Dr. Amélie »

8.2.3 Retour des utilisateurs

Les situations d'alerte de TEMPAS sont définies par les utilisateurs. Lorsque les conditions d'activation sont satisfaites et la qualité minimale est atteinte, des alertes sont notifiées. Toutefois, il peut s'agir de faux-positifs. Dans ce cas, l'utilisateur adapte le système d'alertes en modifiant les connaissances concernant les ressources observables ou en modifiant la situation d'alerte et les conditions d'activation qui la composent. En revanche, nous ne récupérons pas les retours de l'utilisateur concernant son avis sur l'alerte détectée, s'il s'agit d'un vrai-positif ou d'un faux-positif.

Nous optons pour une approche participative des utilisateurs lors de la définition des situations d'alerte et de son adaptation. Une situation d'alerte bien paramétrée fait ce que l'utilisateur attend d'elle. Dans Futura Smart Design, l'état « nouvelle » d'une alerte est changé manuellement à « traitée » et automatiquement à « périmée ». Nous pourrions profiter du changement manuel de l'état de l'alerte pour récupérer les retours des utilisateurs. Ainsi, une alerte marquée comme « traitée » pourrait s'interpréter comme un vrai-positif. Un nouvel état « fausse » en plus des états précédents permettrai aux utilisateurs d'exprimer qu'il s'agit d'un faux-positif.

Identifier explicitement les vrais-positifs et les faux-positifs permet de calculer la précision de la situation d'alerte dans son état actuelle. En revanche, si la situation d'alerte ou les connaissances utilisées pour son évaluation changent la précision change à « inconnue » étant donné que les entrées utilisées pour déterminer une alerte ont changé. La précision pourrait être présentée aux utilisateurs à côté du nombre d'alertes détectées avec la configuration actuelle de la situation d'alerte afin qu'il puisse l'exploiter comme un indicateur de la situation d'alerte. Si la précision est très basse, l'utilisateur pourrait décider d'adapter la situation d'alerte. TEMPAS calcule des indicateurs des alertes détectées (IAA, ICA). Calculer des indicateurs des situations d'alerte est une piste à explorer.

Récupérer le retour de chaque utilisateur permettrait aussi de faire des recommandations aux utilisateurs par profil et par situation d'alerte.

Exemple

L'infirmière « Anne » est la destinataire d'une situation d'alerte « SA1 » associée aux ressources « R1 », « R2 », « R3 », et « R4 » dont la précision des situations d'alerte est 98%.

L'infirmière « Béatrice » est la destinataire d'une situation d'alerte « SA 2 » associée aux ressources « R1 », « R2 », « R3 », et « R5 » dont la précision des situations d'alerte est 70%.
Le système pourrait recommander à l'infirmière « Béatrice » d'utiliser la ressource « R4 » dans la situation d'alerte « SA 2 ».

8.2.4 Exploration d'autres domaines d'application

Nous avons développé TEMPAS comme une solution indépendante du domaine d'application. Toutefois le cadre du développement de l'application cliente ainsi que des expérimentations nous ont orientés vers le domaine médical. Nous avons testé TEMPAS sur des données issues d'une unité de soins intensifs de nouveaux-nés et sur des données de patients médicalisés à domicile. Dans la deuxième expérimentation les utilisateurs ont associé une finalité aux situations d'alerte. Il serait très intéressant de continuer des expérimentations avec des utilisateurs de profils différents qui suivent les mêmes patients, par exemple dans l'établissement de santé et chez eux. Ceci nous permettrait de voir l'adaptabilité de TEMPAS dans différents environnements de santé avec des patients en commun.

Utiliser TEMPAS dans d'autres domaines d'application comme la domotique, l'intégrer avec des systèmes d'information où l'utilisateur participe à sa construction, ou des systèmes fonctionnant sur des plates-formes NoSQL nous permettrait d'évaluer la flexibilité et adaptabilité de TEMPAS.

8.2.5 Impact

Il serait intéressant d'établir la terminologie la plus adéquate pour les utilisateurs afin qu'ils puissent s'approprier plus facilement l'outil et éviter ainsi de possibles frustrations qui mèneraient à la réduction de l'utilisation du système. Une formation des utilisateurs est facultative pour bien se servir de TEMPAS qui introduit de nombreux concepts, fonctionnalités, et une flexibilité atteinte par un grand paramétrage. La formation à l'utilisation de l'application peut être faite aux utilisateurs concepteurs des situations d'alerte mais aussi au personnel du département informatique des établissements afin qu'ils puissent accompagner les nouveaux utilisateurs lors de la conception des situations d'alerte complexes.

Les expérimentations avec des utilisateurs du domaine de la télémédecine ont mis en évidence l'utilité d'un système d'alertes tel que TEMPAS qui permet de résoudre des besoins immédiats. Toutefois, il est nécessaire de connaître son impact au sein des organisations pour démontrer sa réelle utilité opérationnelle permanente. Pour cela fournir des moyens pour connaître la courbe d'apprentissage de TEMPAS, la valeur ajoutée, faciliterait l'utilisation du système. Par exemple, à l'aide des statistiques concernant le gain en productivité ou, dans le cas de la santé, en termes de consommation de services de soin, ou encore à partir du nombre d'alertes notifiées, déterminer combien ont servi à améliorer les processus à l'intérieur de l'établissement, ou combien de nouveaux services de soin ont été consommés grâce à une meilleure prise en charge du patient à la suite de l'utilisation de TEMPAS, seraient des appuis significatifs pour aider à l'amélioration du système d'information de santé.

Bibliographie

Agrawal, R., Faloutsos, C. & Swami, A., 1993. Efficient similarity search in sequence databases. *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms,* pp. 69-84.

Agrawal, R., Psaila, G., Wimmers, E. L. & Za, M., 1995. Querying Shapes of Histories. *Proceedings of the 21th International Conference on Very Large Data Bases,* pp. 502-514.

Allen, J. F., 1983. Maintaining Knowledge About Temporal Intervals. *Commun. ACM,* 26(11), pp. 832--843.

Alsubhi, K., Aib, I. & Boutaba, R., 2012. FuzMet: a fuzzy-logic based alert prioritization engine for intrusion detection systems. *International Journal of Network Management,* 22(4), pp. 263-284.

Association de Dialyse Varoise (A.DI.VA.), 2010. *Informations : reins, insuffisance rénale, hémodialyse.* [Online]
Available at: http://www.association-dialyse-varoise.fr/informations-reins-hemodialyse.php
[Accessed 10 08 2014].

ASTER, 2010. *ASTER : alerte épidémiologique en temps réel.* [Online]
Available at: http://www.defense.gouv.fr/sante/actualites/aster-alerte-epidemiologique-en-temps-reel
[Accessed 08 2014].

Atzema, C. et al., 2006. ALARMED: adverse events in low-risk patients with chest pain receiving continuous electrocardiographic monitoring in the emergency department. A pilot study. *The American Journal of Emergency Medicine,* 24(1), pp. 62-67.

Bai, Y. & Wang, D., 2006. Fundamentals of Fuzzy Logic Control — Fuzzy Sets, Fuzzy Rules and Defuzzifications. In: Y. Bai, H. Zhuang & D. Wang, eds. *Advanced Fuzzy Logic Technologies in Industrial Applications.* London: Springer , pp. 17-36.

Bai, Y., Zhuang, H. & Wang, D., 2007. *Advanced fuzzy logic technologies in industrial applications.* London: Springer.

Bakar, N. & Belaton, B., 2005. Towards implementing intrusion alert quality framework. *Distributed Frameworks for Multimedia Applications,* pp. 198-205.

Baldauf, M., Dustdar, S. & Rosenberg, F., 2007. A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.,* 2(4), pp. 263-277.

Banko, M. & Brill, E., 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics,* pp. 26-33.

Bay Alarm Medical, 2014. *Medical Alert Systems.* [Online]
Available at: http://www.bayalarmmedical.com/
[Accessed 08 2014].

Be Alert, 2014. *Be-Alert: soyez alerté en situation d'urgence | DG Centre de Crise.* [Online]
Available at: http://be-alert.be/
[Accessed 08 2014].

Benazzouz, Y., 2011. *Découverte de contexte pour une adaptation automatique de services en intelligence ambiante,* Saint-Etienne: Ecole Nationale Supérieure des Mines de Saint-Etienne.

Beuscart, R., McNair, P. & Brender, J., 2009. The PSIP Consortium, Patient safety through intelligent procedures in medication: the PSIP project. *Studies in health technology and informatics,* pp. 6-13.

Borowski, M., Siebig, S., Wrede, C. & Imhoff, M., 2011. Reducing False Alarms of Intensive Care Online-Monitoring Systems: An Evaluation of Two Signal Extraction Algorithms. *Computational and Mathematical Methods in Medicine.*

Brooke, J., 1996. SUS: A quick and dirty usability scale In Usability evaluation in industry. *Usability Evaluation in Industry.*

Brouhard, R., 2012. *Definition of Medical Alert System.* [Online]
Available at: http://firstaid.about.com/od/glossary/g/10_Medical-Alert-System.htm
[Accessed 02 06 2014].

Bucur, O., Beaune, P. & Boissier, O., 2005. What is context and how can an agent learn to find and use it when making decisions?. *Multi-Agent Systems and Applications IV,* pp. 112-121.

Cayuga, 2013. *SourceForge.net: Cayuga Complex Event Processing System - Project Web Hosting - Open Source Software.* [Online]
Available at: http://cayuga.sourceforge.net/
[Accessed 08 2013].

Chandrasekaran, M., Baig, M. & Upadhyaya, S., 2006. AVARE: aggregated vulnerability assessment and response against zero-day exploits. *Performance, Computing, and Communications Conference,* pp. 603-610.

Chan, K.-P. & Fu, A.-C., 1999. Efficient time series matching by wavelets. *Data Engineering, 1999. Proceedings., 15th International Conference on,* pp. 126-133.

Chan, P. K. & Mahoney, M. V., 2005. Modeling Multiple Time Series for Anomaly Detection. *Proceedings of the Fifth IEEE International Conference on Data Mining,* pp. 90-97.

Charbonnier, S. & Gentil, S., 2007. A trend-based alarm system to improve patient monitoring in intensive care units. *Control Engineering Practice,* 15(9), pp. 1039-1050.

Charbonnier, S. & Portet, F., 2012. A self-tuning adaptive trend extraction method for process monitoring and diagnosis. *Journal of Process Control,* 22(6), pp. 1127-1138.

CHU de Liège, 2010. *Site CHU - Dialyse Péritonéale.* [Online]
Available at: http://www.chu.ulg.ac.be/jcms/c_152191/dp
[Accessed 10 08 2014].

Cisco, 2013. *Cisco Prime Infrastructure 2.0 User Guide.* San Jose, CA: Cisco Systems, Inc..

City of Decatur, 2008. *City Code of Decatur.* [Online]
Available at: http://www.ci.decatur.il.us/citycode/code45.pdf
[Accessed 02 06 2014].

Combi, C. & Montanari, A., 2001. Data Models with Multiple Temporal Dimensions: Completing the Picture. *Advanced Information Systems Engineering,* Volume 2068, pp. 187-202.

Console, L. & Torasso, P., 1988. A Logical Approach To Deal With Incomplete Causal Models In Diagnostic Problem Solving. *Uncertainty and Intelligent Systems},* Volume 313, pp. 255-264.

Cvach, M., 2012. Monitor Alarm Fatigue: An Integrative Review. *Biomedical instrumentation & technology / Association for the Advancement of Medical Instrumentation,* 46(4), pp. 268-277.

Dey, A. K., 2001. Understanding and Using Context. *Personal Ubiquitous Computing,* 5(1), pp. 4-7.

Dey, A. K., Abowd, G. D. & Salber, D., 2001. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction,* Volume 16, pp. 97-166.

Diatelic HD, 2014. *http://www.diatelic.fr/.* [Online]
Available at: http://www.diatelic.fr/docs/Diatelic_HD.pdf
[Accessed 02 08 2014].

Diatelic, 2011. *Diatelic.* [Online]
Available at: http://www.diatelic.fr/diatelic_DP.html
[Accessed 10 08 2014].

Drools, 2013. *Drools.* [Online]
Available at: http://drools.jboss.org/drools-fusion.html
[Accessed 08 2013].

Esper, 2013. *Esper - Complex Event Processing.* [Online]
Available at: http://esper.codehaus.org/
[Accessed 08 2013].

Freksa, C., 1992. Temporal Reasoning Based on Semi-intervals. *Artif. Intell.,* 54(1-2), pp. 199-227.

Gaudin, C., Bonnardel, N., Pellegrin, L. & Chaudet, H., 2012. Les activités de gestion d'alerte épidémiologique : les transformations induites par l'utilisation d'un système de surveillance en temps réel. *Perspectives interdisciplinaires sur le travail et la santé,* 1(14).

Gee, T. & Moorman, B. A., 2011. Reducing Alarm Hazards: Selection and Implementation of Alarm Notification Systems. *Patient Safety & Quality Healthcare,* 8(2), pp. 14-17.

Gürgen, L., 2007. *Gestion à grande échelle de données de capteurs hétérogènes,* Grenoble: Institut National Polytechnique de Grenoble.

Hasle, P. & Øhrstrøm, P., 1995. *Temporal Logic : From Ancient Ideas to Artificial Intelligence.* Dordrecht: Springer Netherlands.

Heierman, I. E. O. & Cook, D. J., 2003. Improving Home Automation by Discovering Regularly Occurring Device Usage Patterns. *Proceedings of the Third IEEE International Conference on Data Mining,* pp. 537-540.

Henricksen, K., 2003. *A Framework for Context-Aware Pervasive Computing Applications,* Brisbane: The university of queensland.

Hripcsak, G., 1994. Writing Arden Syntax medical logic modules. *Computers in Biology and Medicine,* 24(5), pp. 331 - 363.

Hunter, J. & McIntosh, N., 1999. Knowledge-Based Event Detection in Complex Time Series Data. *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making,* pp. 271-280.

Hyperic, 2013. *vFabric Documentation Center.* [Online]
Available at: http://pubs.vmware.com/vfabric5/index.jsp
[Accessed 08 2014].

Iskio, J. et al., 2006. Improving Acceptance of Computerized Prescribing Alerts in Ambulatory Care. *Journal of the American Medical Informatics Association:JAMIA,* 13(1), pp. 5-11.

Jan, N.-Y., Lin, S.-C., Tseng, S.-S. & Lin, N. P., 2009. A decision support system for constructing an alert classification model. *Expert Systems with Applications,* 36(8), pp. 11145-11155.

Jenders, R. & Hripcsak, G., 1995. Medical decision support: experience with implementing the Arden Syntax at the Columbia-Presbyterian Medical Center.. *Application in Medical,* pp. 169-173.

Jensen, C. S. et al., 1997. The Consensus Glossary of Temporal Database Concepts - February. *Temporal Databases, Dagstuhl,* pp. 367-405.

Jensen, C. S. & Snodgrass, R. T., 2008. *Temporal Database Entries for the Springer Encyclopedia of Database Systems.* Other: Timecenter.

Kahn, K. & Gorry, G., 1977. Mechanizing temporal knowledge. *Artificial Intelligence,* 9(1), pp. 87 - 108.

Kam, P.-s. & Fu, A. W.-C., 2000. Discovering Temporal Patterns for Interval-Based Events. *Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery,* pp. 317-326.

Keogh , E. . J., Chu, S., Hart, D. & Pazzani, M. J., 2001. An Online Algorithm for Segmenting Time Series. *Proceedings IEEE International Conference on Data Mining,* pp. 289-296.

Keogh, E., Chakrabarti, K., Pazzani, M. & Mehrotra, S., 2001. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems,* 3(3), pp. 263-286.

Keogh, E., Chu, S., Hart, D. & Pazzani, M., 1993. Segmenting Time Series: A Survey and Novel Approach. *Data mining in Time Series Databases,* pp. 1-22.

Keravnou, E., 1996. Temporal diagnostic reasoning based on time-objects. *Artificial Intelligence in Medicine,* 8(3), pp. 235-265.

Kesselheim, A. S. et al., 2011. Clinical decision support systems could be modified to reduce 'alert fatigue'while still minimizing the risk of litigation. *Health Affairs,* 30(12), pp. 2310-2317.

Kim, S.-K. & Chakravarthy, S., 1994. Modeling time: Adequacy of three distinct time concepts for temporal databases. *Entity-Relationship Approach,* Volume 823, pp. 475-491.

King, A. et al., 2012. Evaluation of a smart alarm for intensive care using clinical data. *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE,* pp. 166-169.

Kowalski, R. & Sergot, M., 1986. A Logic-based Calculus of Events. *New Generation Computing,* 4(1), pp. 67-95.

Krall, M. A. & Sittig, D. F., 2002. Clinician's assessments of outpatient electronic medical record alert and reminder usability and usefulness requirements.. *American Medical Informatics Association (AMIA) Annual Symposium,* pp. 400-404.

Laerhoven, K., 2001. Combining the Self-Organizing Map and K-Means Clustering for On-Line Classification of Sensor Data. *Artificial Neural Networks — ICANN 2001,* Volume 2130, pp. 464-469.

Larizza, C., Bellazzi, R. & Riva, A., 1997. Temporal Abstractions for diabetic patients management. *Artificial Intelligence in Medicine,* Volume 1211, pp. 319-330.

Larizza, C., Bernuzzi, G. & Stefanelli, M., 1995. A general framework for building patient monitoring systems. *Artificial Intelligence in Medicine,* Volume 934, pp. 91-102.

LifeFone, 2014. *Medical Alarm & Medical Alert Systems.* [Online]
Available at: http://www.lifefone.com/
[Accessed 08 2014].

Lowe, A., Harrison, M. J. & Jones, R. W., 1999. Diagnostic monitoring in anaesthesia using fuzzy trend templates for matching temporal patterns. *Artificial Intelligence in Medicine,* 16(2), pp. 183 - 199.

Manzi de Arantes Junior, W., 2006. *Pluggable Alert System,* Lyon: Ecole Doctorale Informatique et Information pour la Société.

Mayrhofer, R. M., 2004. *An Architecture for Context Prediction,* Linz: Johannes Kepler University.

Mccarthy, J. & Hayes, P. J., 1969. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: *Machine Intelligence.* Edinburg: Edinburgh University Press, pp. 463-502.

Mcdermott, D., 1982. A temporal logic for reasoning about processes and plans. *Cognitive Science,* 6(2), pp. 10--155.

Mörchen, F., 2006. Algorithms for Time Series Knowledge Mining. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 668-673.

Musen, M., Shahar, Y. & Shortliffe, E., 2006. Clinical Decision-Support Systems. In: E. Shortliffe & J. Cimino, eds. *Biomedical Informatics.* New York: Springer New York, pp. 698-736.

Ngo, L., Haddawy, P., Krieger, R. A. & Helwig, J., 1997. Efficient temporal probabilistic reasoning via context-sensitive model construction. *Computers in Biology and Medicine,* 27(5), pp. 453-476.

Nihon Kohden Corporation, 2014. *Nihon Kohden - BSM-1700.* [Online] Available at: http://www.nihonkohden.com/products_en/type/mon/bsm1700.html [Accessed 08 2014].

Nihon Kohden Corporation, 2014. *Nihon Kohden - BSM-1700.* [En ligne] Available at: http://www.nihonkohden.com/products_en/type/mon/bsm1700.html [Accès le 08 2014].

Nihon Kohden Corporation, 2014. *Nihon Kohden - CNS-9601.* [Online] Available at: http://www.nihonkohden.com/products_en/type/mon/cns9601.html [Accessed 08 2014].

Njogu, H. W., Jiawei, L., Kiere, J. N. & Hanyurwimfura, D., 2013. A comprehensive vulnerability based alert management approach for large networks. *Future Generation Computer Systems,* 29(1), pp. 27-45.

Ozsoyoilu, G. & Snodgrass, R. T., 1995. Temporal and Real-Time Databases: A Survey. *IEEE Trans. Knowl. Data Eng.,* 7(4), pp. 513-532.

Petit, L., 2012. *Gestion de flux de données pour l'observation de systèmes,* Grenoble: Université de Grenoble.

Petit, L., Labbe, C. & Roncancio, C., 2010. An algebric window model for data stream management. *Proceedings of the Ninth ACM International Workshop on Data Engineering for Wireless and Mobile Access,* pp. 17-24.

Phansalkar, S. et al., 2010. A review of human factors principles for the design and implementation of medication safety alerts in clinical information systems. *Journal of the American Medical Informatics Association:JAMIA,* 17(5), pp. 493-501.

Philips, 2009. *Philips Healthcare Informatics Strategy.* [Online] Available at: http://www.healthleadersmedia.com/content/240723.pdf [Accessed 03 03 2014].

Philips, 2011. *Smart systems, saving lives : eICU Program.* [En ligne] Available at: http://telehealthtechnology.org/sites/default/files/documents/user_reviewed_products/eICUProgram_Datasheet.pdf [Accès le 03 03 2014].

Philips, 2014. *Philips eICU Program.* [Online]
Available at:
http://www.healthcare.philips.com/main/products/patient_monitoring/products/eicu
[Accessed 08 2014].

Portet, F. et al., 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence,* 173(7), pp. 789-816.

Prekop, P. & Burnett, M., 2003. Activities, Context and Ubiquitous Computing. *Comput. Commun.,* 26(11), pp. 1168-1176.

Prior, A. N., 1957. *Time and Modality.* Oxford: Clarendon Press.

Pryor, T. & Hripcsak, G., 1993. The arden syntax for medical logic modules. *International journal of clinical monitoring and computing,* 10(4), pp. 215-224.

Reach Alert, 2014. *Reach Alert | Emergency and Group Messaging for groups of all sizes.* [Online]
Available at: http://reachalert.com/
[Accessed 08 2014].

ReachPlus , A. S., 2014. *ReachPlus Alerts | Mass Notification Made Easy.* [Online]
Available at: http://www.reachplus.com/
[Accessed 08 2014].

Roddick, J. F. & Mooney, C. H., 2005. Linear Temporal Sequences and Their Interpretation Using Midpoint Relationships. *IEEE Trans. on Knowl. and Data Eng.,* 17(1), pp. 133-135.

Salatian, A. & Hunter, J., 1999. Deriving Trends in Historical and Real-Time Continuously Sampled Medical Data. *Journal of Intelligent Information Systems,* 13(1-2), pp. 45-71.

Schockaert, S., De Cock, M. & Kerre, E. E., 2008. Fuzzifying Allen's Temporal Interval Relations. *Trans. Fuz Sys.,* 16(2), pp. 517-533.

Shahar, Y., 1997. A framework for knowledge-based temporal abstraction. *Artificial Intelligence,* 90(1-2), pp. 79-133.

Shahar, Y., Miksch, S. & Johnson, P., 1998. The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine,* 14(1-2), pp. 29-51.

Shahar, Y. & Musen, M. A., 1993. RESUME: a temporal-abstraction system for patient. *Comput Biomed Res,* pp. 255-273.

Shanahan, M., 1999. The Event Calculus Explained. In: M. J. Wooldridge & M. Veloso, eds. *Artificial Intelligence Today.* Berlin: Springer-Verlag, pp. 409-430.

Shoemaker , A. L., 1996. What's Normal? -- Body Temperature, Gender, and Heart Rate. *Journal of Statistics Education,* 4(2), p. 1.

Simmers, L. M., Simmers-Nartker, K. & Simmers-Kobelak, S., 2008. *Diversified Health Occupations.* 7 ed. New York: Delmar Cengage Learning.

Sittig , D. F., Cheung, K.-H. & Berman, L., 1992. Fuzzy classification of heart rate trends and artifacts. *International journal of clinical monitoring and computing,* 9(4), pp. 251-257.

Snodgrass, R. T., 1995. *The TSQL2 Temporal Query Language.* -: Kluwer.

Soo, M. D., Jensen, C. S. & Snodgrass, R. T., 1995. An Algebra for TSQL2. *The TSQL2 Temporal Query Language, chapter 27,* pp. 505-546.

Storm, 2013. *Storm, distributed and fault-tolerant realtime computation.* [Online]
Available at: http://storm.incubator.apache.org/
[Accessed 08 2013].

Strong, D. M., Lee, Y. W. & Wang, R. Y., 1997. Data Quality in Context. *Commuications of the ACM,* 40(5), pp. 103-110.

Sybase Event Stream Processor, 2013. *Complex Event Processing (CEP) Technology & Real-Time Business Process Management - SAP Sybase Event Stream Processor - Sybase Inc.* [Online]
Available at: http://www.sybase.com/products/financialservicessolutions/complex-event-processing
[Accessed 08 2013].

TMM Software, 2013. *ap'Telecare – TMM Software.* [En ligne]
Available at: http://www.tmm-software.com/?page_id=2923
[Accès le 10 08 2014].

Tu, S. W. et al., 1995. Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTEGE-II to protocol-based decision support. *Artificial intelligence in medicine,* 7(3), pp. 257-289.

Udechukwu, A., Barker, K. & Alhajj, R., 2004. Discovering all frequent trends in time series. *Proceedings of the winter international synposium on Information and communication technologies,* pp. 1-6.

van der Sijs, H., Aarts , J., Vulto , A. & Berg, M., 2006. Review Paper: Overriding of Drug Safety Alerts in Computerized Physician Order Entry. *Journal of the American Medical Informatics Association:JAMIA,* 13(2), pp. 138-147.

Villafane, R., Hua, K. A. & Tran, D., 1999. Mining Interval Time Series. *Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery,* pp. 318-330.

Wang, R. Y. & Strong, D. M., 1996. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems,* 12(4), pp. 5-33.

WordNet, 2014. *WordNet.* [Online]
Available at: http://wordnetweb.princeton.edu/perl/webwn
[Accessed 08 2014].

Wyckoff, M., 2009. Improving how we use and respond to clinical alarms. *American Nurse Today,* 4(9), pp. 37-39.

Zadeh, L., 1965. Fuzzy sets. *Information and Control,* 8(3), pp. 338-353.

Annexes

Annexe A – Guide d'utilisation de TEMPAS

TEMPAS est un système d'alertes qui permet d'exploiter les données d'autres systèmes d'information afin de définir des situations d'alerte.

Une situation d'alerte est définie à partir d'une ou plusieurs conditions qui lorsque satisfaites génèrent une alerte. Chaque condition a une importance. Ainsi le plus de conditions sont satisfaites le plus élevée est la qualité d'une alerte détectée.

Les conditions d'activation sont construites à partir des valeurs linguistiques qui expriment l'état ou la tendance d'une variable. Par exemple « température élevée » ou « tendance de la température à la hausse ». Les valeurs linguistiques partagent des zones d'incertitude. Par exemple, une température entre 37.1 °C et 37. 5 °C est considéré comme « normale » et au même temps « élevée ». S'il s'agit 37.4°C, c'est plus « élevée que normale ».

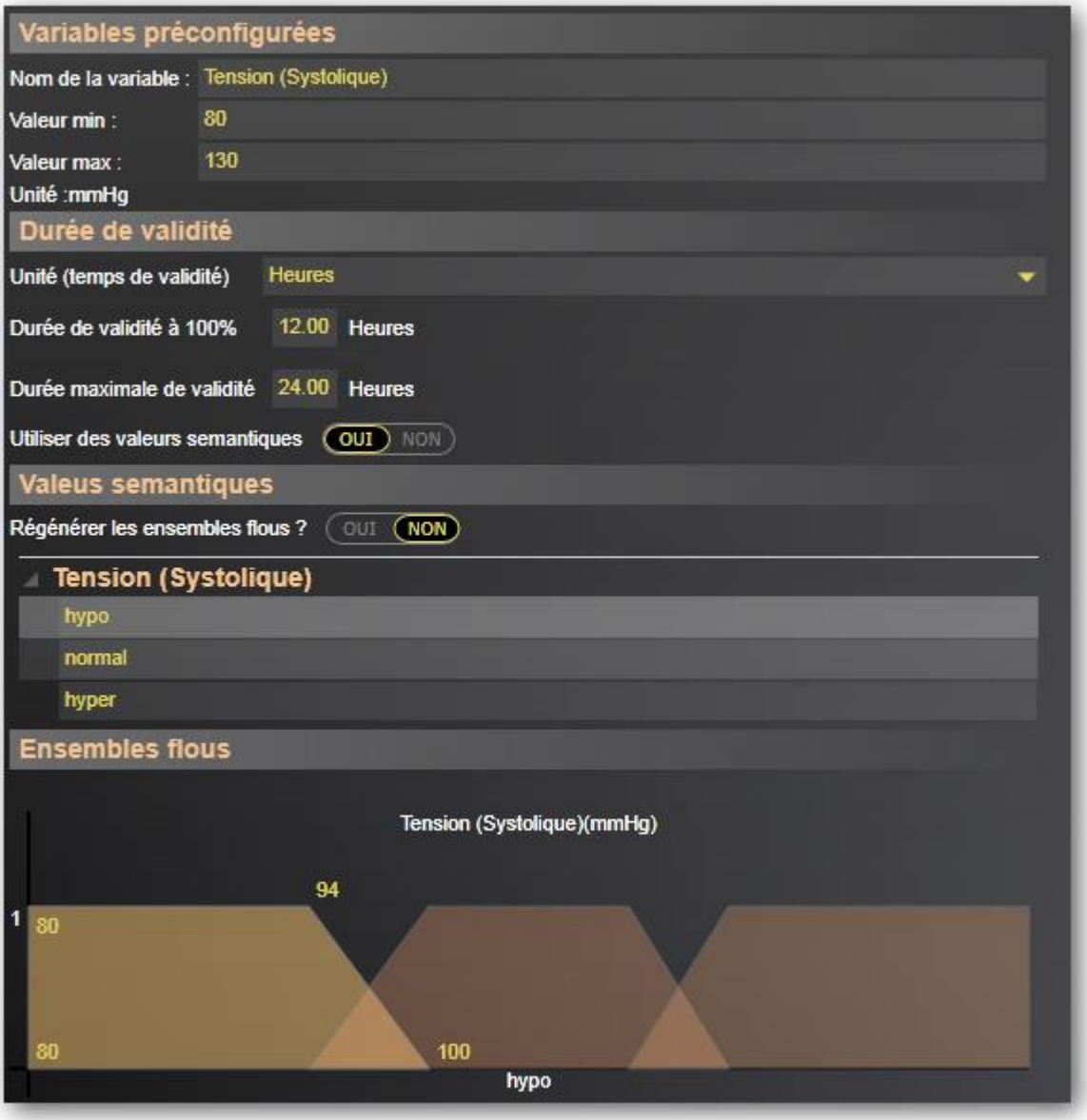
Le processus à suivre pour définir une situation d'alerte dans FSD est : Préconfigurer les variables (si jamais elles ne le sont pas déjà) ; Instancier les variables ; et définir les situations d'alerte.

Configuration des alertes

1. Configuration des variables

Composant[Gestion des Variables]

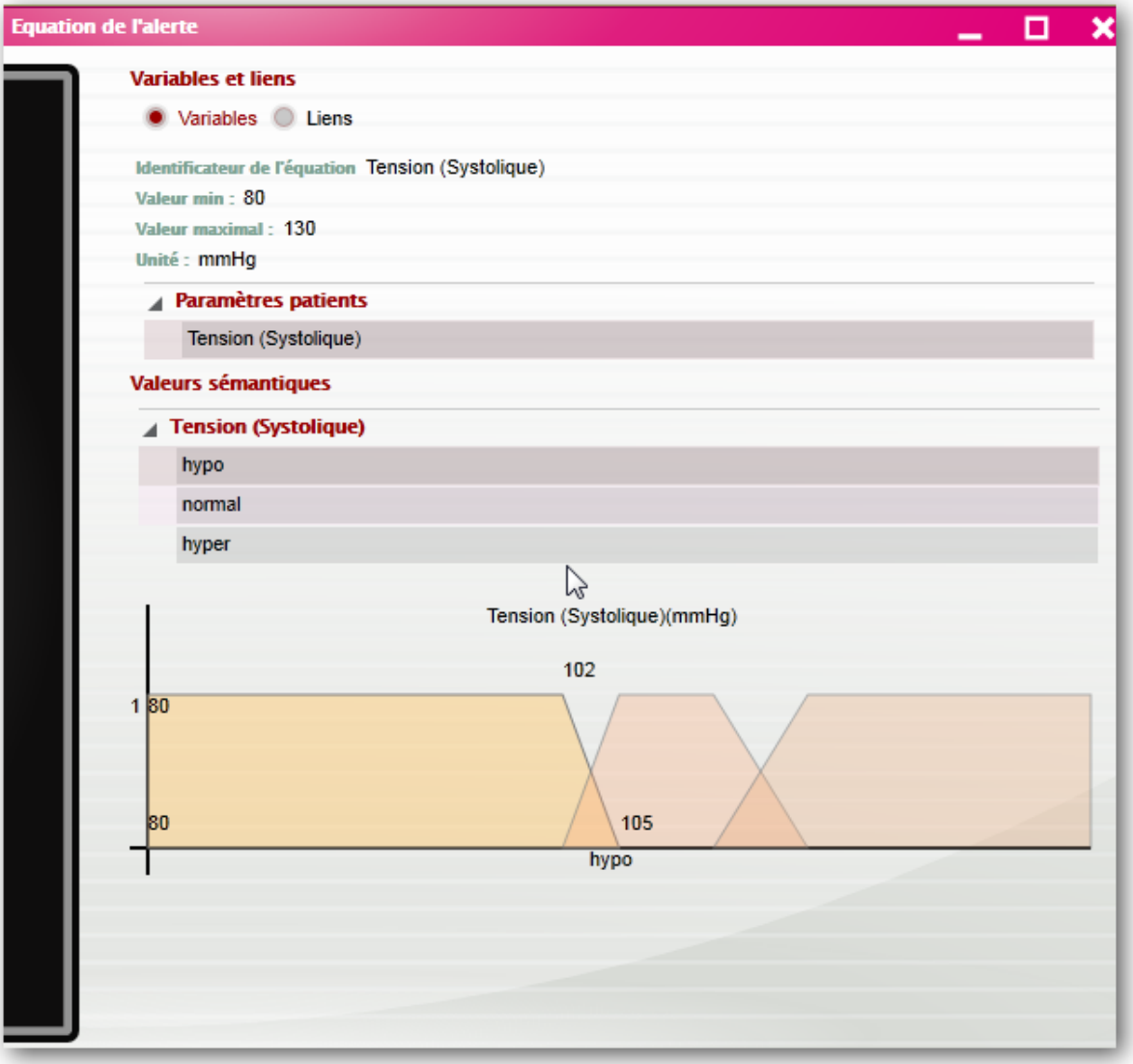
Avant de pouvoir définir des situations d'alerte il est nécessaire de préconfigurer les variables qui vont être utilisées pour la construction des alertes. La pré-configuration d'une variable a lieu une seule fois, ensuite elle peut être utilisée et redéfinie plusieurs fois lors de la construction des différentes situations d'alerte.



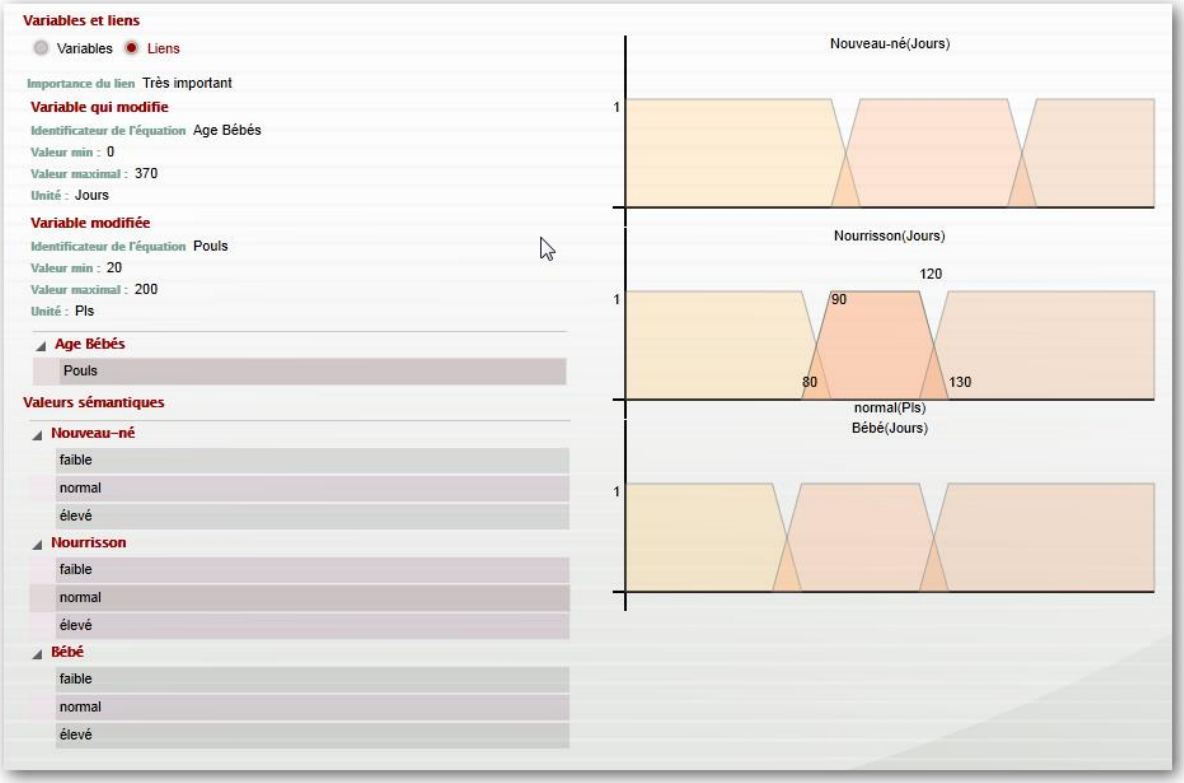
2. *Instanciation des variables préconfigurées*

Composant[Equation de l'alerte] et Composant[Tendances]

L'instanciation des variables préconfigurées permet de redéfinir quelques propriétés comme le calcul des valeurs linguistiques. Cette étape est indispensable pour pouvoir définir des situations d'alerte.



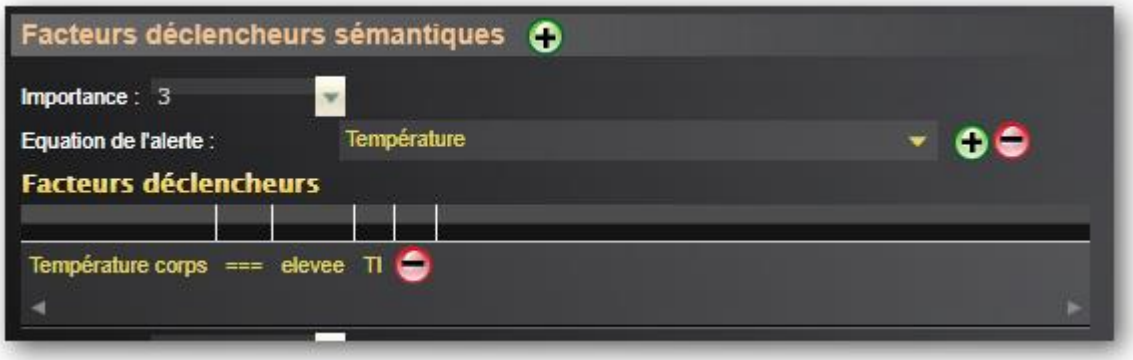
Grâce au graphe modificateur, il est possible de définir comment les valeurs linguistiques d'une variable modifient les valeurs linguistiques d'une autre. Par exemple, dans le cas des bébés (moins d'un an), les valeurs linguistiques modifient ceux du pouls



3. *Définition des situations d'alerte*

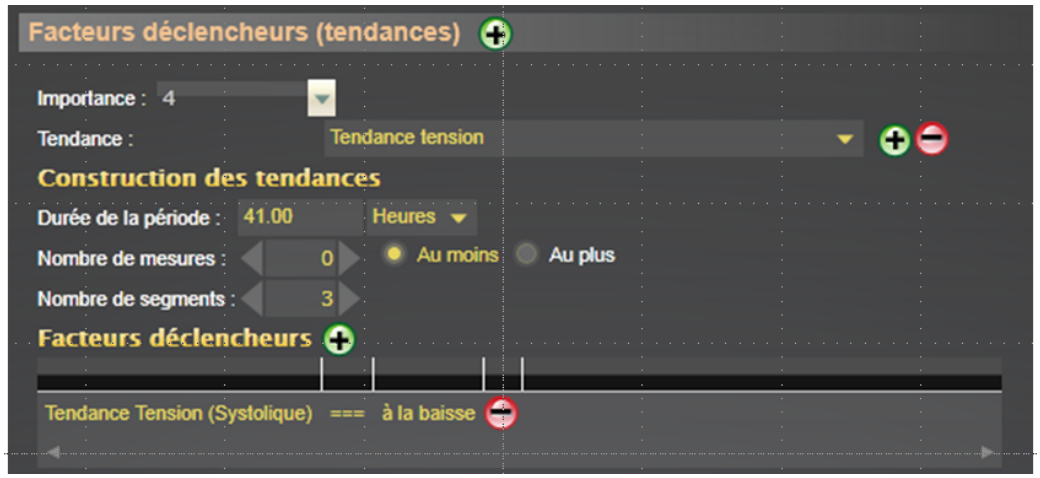
Composant[Alerte]

Les situations d'alerte sont définies à partir des conditions d'activation. Celles-ci sont construites avec les variables instanciées précédemment et des valeurs linguistiques de référence. Par exemple, si « la température est élevée » et « la tendance de la tension systolique est à la baisse ».

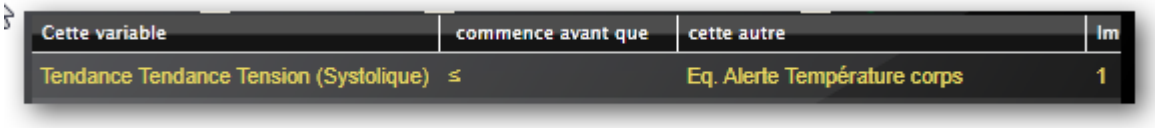


Dans le cas des tendances, il est nécessaire de définir la période dans laquelle les tendances doivent être détectées. Par exemple, avec toutes les observations des dernières 41 heures. Si jamais il le faut, il est possible de limiter le nombre d'observations à utiliser inclues dans la période définie. Finalement, le nombre de

segments à détecter. Chaque segment détecté est utilisé pour la vérification de la condition d'activation. C'est-à-dire, si un des segments représente une tendance à la baisse (regarder la figure ci-dessous), il est utilisé pour l'instanciation de l'alerte détectée.



Un deuxième type de condition permet de mieux définir la situation d'alerte. Ces sont des conditions temporelles. En principe, les observations utilisées pour évaluer une situation d'alerte se passent les unes proches aux autres. Cependant, il est possible de spécifier qu'un événement commence avant ou au même temps qu'un autre.



Une situation d'alerte est définie comme l'addition des conditions. C'est-à-dire que plus de conditions sont satisfaites lors de l'évaluation de la situation d'alerte, plus de qualité auront les alertes détectées. Un filtre sur la qualité permet de notifier, ou non, aux utilisateurs les alertes détectées.

L'évaluation d'une situation d'alerte utilise les informations les plus récentes. Pour les valeurs linguistiques exprimant l'état d'une variable, la dernière valeur connue est utilisée. Lorsqu'il s'agit d'une tendance il est nécessaire de définir la période sur laquelle doivent se détecter les tendances, ainsi que le nombre de tendances à trouver.

Annexe B – Alertes d'infection chez les bébés

Infection chez les bébés

DEMO 1

Pré-paramétrer une variable

- Ouvrir le composant [Gestion de variables]
    - Pré-paramétrer la variable pouls :
        - Ajouter une nouvelle variable
        - Aller à la section Durée de validité
            - Temps de validité : 30 minutes
                - Une donnée est 100% fiable si elle date de moins de deux minutes
            - Temps d'expiration : 50 minutes
                - Une donnée n'est plus fiable si elle date de plus de cinq minutes
        - Allez dans la section Valeurs sémantiques :
            - Trois valeurs linguistiques :
                - Faible
                - Normal
                - Elevé
        - Allez dans la section Ensembles flous
            - Faible : min 20 – max 100 => (20, 20, **90, 100**)
            - Normal : min 90 - max 130 => (90, 100, **120, 130**)
            - Elevé : min 120 – max 200 => (120, 130, 200, 200)
    - Pré-paramétrer la variable Température :
        - Ajouter une nouvelle variable
        - Aller à la section Durée de validité

- Temps de validité : 30 minutes
  - Une donnée est 100% fiable si elle date de moins de deux minutes
- Temps d'expiration : 50 minutes
  - Une donnée n'est plus fiable si elle date de plus de cinq minutes

- Allez dans la section Valeurs sémantiques :
  - Trois valeurs linguistiques :
    - Basse
    - Normale
    - Elevée
- Allez dans la section Ensembles flous
  - Basse : min 35 °C – max 36,9 °C => (35, 35, **36.7, 36.9**)
  - Normale : min 36,7 °C – max 37,4 °C => (36.7, 36.9, **37.2, 37.4**)
  - Elevée : min 37,2 °C - max 42 °C => (37.2, 37.4, 42, 42)

Instancier les variables

- Ouvrir le composant [Equation de l'alerte]
  - Nommer l'équation comme « *infection chez les bébés* »
  - Ajouter le pouls à la zone de variables (glisser-déposer)
  - Ajouter la température à la zone de variables

Définir la situation d'alerte

- Ouvrir le composant [Alerte]
  - Aller à la section ***Informations de l'alerte***
    - La nommer l'alerte « infection chez les bébés »

- Le message à afficher est « Risque d'infection chez le bébé »

- Aller à la section ***Facteurs déclencheurs sémantiques***
  - Nous allons définir les conditions d'activation de l'alerte
  - **POUR LE POULS**
    - Faire click dans le bouton (+) à côté du titre
    - Donner l'importance de 3
    - Sélectionner l'équation de l'alerte « *infection chez les bébés* »
      - Ajouter les facteurs déclencheurs
        - Faire click dans le bouton (+)
          - Sélectionner le Pouls
          - =
          - Elevé
          - Très important
          - Faire click dans le bouton (+) au-dessous
  - **POUR LA TEMPERATURE**
    - Faire click dans le bouton (+) à côté du titre
    - Donner l'importance de 2
    - Sélectionner l'équation de l'alerte « *infection chez les bébés* »
      - Ajouter les facteurs déclencheurs
        - Faire click dans le bouton (+)
          - Sélectionner la Température
          - =
          - Elevée

- Très important
- Faire click dans le bouton (+) au-dessous

  - Aller à la section ***Cycle de vie de l'alerte***
    - Nous allons définir pendant combien de temps il est possible d'agir sur une alerte détectée. Ajuster le
      - Temps de validité : 20 minutes

Rentrer des informations sur les patients pour évaluer les alertes

- Charger le patient 1
  - Taper « bébé » dans la barre de recherche en bas et à gauche de l'application
  - Sélectionner le patient « Premier bébé »
- Ouvrir le composant « Suivi paramètres »
  - Ajouter un pouls de :
    - Un pouls de 130
    - Une température de 38
    - A l'heure actuelle

- Charger le patient 2
  - Taper « bébé » dans la barre de recherche en bas et à gauche de l'application
  - Sélectionner le patient « Deuxième bébé »
- Ouvrir le composant « Suivi paramètres »
  - Ajouter un pouls de :
    - Un pouls de 130
    - Une température de 38
    - A l'heure actuelle

****

DEMO 2 (A) Contextualisation

Instancier les variables

- Ouvrir le composant [Equation de l'alerte]
    - Modifier l'équation « *infection chez les bébés* »
    - Ajouter l'âge à la zone de variables (glisser-déposer)
    - Faire un lien de l'âge au pouls
    - Aller dans la section « *Variables et liens* »
    - Choisir « *Liens* »
    - Modifier le calcul des valeurs linguistiques
        - Nouveau-né
            - Pouls
                - Faible : min 20 – max 100 => (20, 20, **90, 100**)
                - Normal : min 90 – max 160 => (90, 100, **150, 160**)
                - Elevé : min 150 – max 200 => (150, 160, 200, 200)
        - Nourrisson
            - Pouls
                - Faible : min 20 – max 90 => (20, 20, **80, 90**)
                - Normal : min 80 – max 130 => (80, 90, **120, 130**)
                - Elevé : min 120 – max 200 => (120, 130, 200, 200)
        - Bébé
            - Pouls
                - Faible : min 20 – max 80 => (20, 20, **70, 80**)
                - Normal : min 70 – max 130 => (70, 80, **120, 130**)

- Elevé : min 120 – max 200 => (120, 130, 200, 200)

Rentrer des informations sur les patients pour évaluer les alertes

- Charger le « Premier bébé »
  - Taper « bébé » dans la barre de recherche en bas et à gauche de l'application
  - Sélectionner le patient « Premier bébé »
- Ouvrir le composant « Suivi paramètres »
  - Ajouter un pouls de :
    - Un pouls de 130
    - Une température de 38
    - A l'heure actuelle

- Charger le « Deuxième bébé »
  - Taper « bébé » dans la barre de recherche en bas et à gauche de l'application
  - Sélectionner le patient « Deuxième bébé »
- Ouvrir le composant « Suivi paramètres »
  - Ajouter un pouls de :
    - Un pouls de 130
    - Une température de 38
    - A l'heure actuelle

****

Nicolas PETIT : 70 jours

Agnes MARTIN : 102 jours

Test 2 (B)

Instancier les variables

- Ouvrir le composant [Equation de l'alerte]
    - Modifier l'équation « *infection chez les bébés* »
    - Aller dans la section « *Variables et liens »*
    - Choisir « *Variables* »
    - Faire click sur la variable « âge-jours »
    - Modifier la valeur sémantique « Nourrisson »
        - Nourrisson : min 100 – max 190 => (**100**, **120**, 180, 190)

Rentrer des informations sur les patients pour les suivre

- Charger le « PETIT Nicolas»
    - Taper « Nicolas» dans la barre de recherche en bas et à gauche de l'application
    - Sélectionner le patient « PETIT Nicolas»
- Ouvrir le composant « Suivi paramètres »
    - Ajouter un pouls de :
        - Un pouls de 130
        - Une température de 38
        - A l'heure actuelle

- Charger le « MARTIN Agnes»
    - Taper « Agnes» dans la barre de recherche en bas et à gauche de l'application
    - Sélectionner le patient « MARTIN Agnes»

- Ouvrir le composant « Suivi paramètres »
    - Ajouter un pouls de :
        - Un pouls de 130
        - Une température de 38
        - A l'heure actuelle

Test 3

Modifier la situation d'alerte

- Ouvrir le composant [Alerte]
    - Aller à la section ***Options de filtrage des alertes***
        - Baisser le filtrage par applicabilité à zéro
        - Baisser le filtrage par confiance à zéro

Interpréter les alertes détectées

- Placer la souris sur les alertes détectées dans la liste d'alertes (en bas, à gauche) et lire l'indice d'applicabilité Initiales **IA**
    - d'abord chez Agnes MARTIN,
    - ensuite chez Nicolas PETIT

****

Test 4 (A)

Rentrer des informations sur les patients pour évaluer les alertes

- Charger le « Premier Bébé »
    - Taper « bébé » dans la barre de recherche en bas et à gauche de l'application
    - Sélectionner le patient « Premier bébé »
- Ouvrir le composant « Suivi paramètres »
    - Ajouter un pouls de :

- Un pouls de 155
- A l'heure actuelle

Test 4 (B)

Modifier la situation d'alerte

- Ouvrir le composant [Alerte]
  - Aller à la section ***Cycle de vie de l'alerte***
    - Aller à l'scénario avec les deux flèches rouges
      - La première option « alerter si l'indice d'applicabilité augmente », le changer par « non »

Rentrer des informations sur les patients pour évaluer les alertes

- Charger le « Premier bébé »
  - Taper « bébé » dans la barre de recherche en bas et à gauche de l'application
  - Sélectionner le patient « Premier bébé »
- Ouvrir le composant « Suivi paramètres »
  - Ajouter un pouls de :
    - Un pouls de 170
    - A l'heure actuelle

****

Test 5 (A)

Rentrer des informations sur les patients pour évaluer les alertes

- Charger le patient « Nicolas Petit »
    - Taper « Nicolas» dans la barre de recherche en bas et à gauche de l'application
    - Sélectionner le patient « PETIT Nicolas »
- Ouvrir le composant « Suivi paramètres »
    - Ajouter un pouls de :
        - Un pouls de 170
        - Une température de 38
        - **DATANT D'UNE HEURE**

Test 5 (B)

Modifier le filtrage dans la liste des alertes

- Aller dans la liste d'alertes à gauche en bas
    - Bouger le curseur de filtrage IC vers la droite.

****

Test 6

Instanciation de la tendance

- Ouvrir le composant [tendance]
- Nommez-la « tendance température »
    - Instancier la tendance «température »
    - Modifier le calcul des valeurs linguistiques :
        - A la baisse : [-90, -10]
        - Stable : [-20, 10]
        - A la hausse : [5, 45]
        - Hausse rapide: [40, 90]

Modifier la situation d'alerte

- Ouvrir le composant [Alerte]
- Modifier la situation d'alerte
- Effacer la condition d'activation « température = hyperthermie»
- Aller à la section « Facteurs déclencheurs (tendances) »
    - Donner une importance de 1
    - Choisir la tendance « tendance température » que l'on vient de créer
    - Définir une *durée de la période* à 30 minutes
    - Définir *nombre de mesures* à 0 « au moins »
    - Définir *nombre de segments* à 2
    - Ajouter la condition d'activation => bouton (+) à côté du titre « facteurs déclencheurs »
        - Sélectionner la température corporelle
        - ≥
        - à la hausse

- Click sur le bouton (+) au-dessous

Rentrer des informations sur les patients pour évaluer les alertes

- Charger le « « PETIT Nicolas »
  - Taper « Nicolas » dans la barre de recherche en bas et à gauche de l'application
  - Sélectionner le patient « PETIT Nicolas »
- Ouvrir le composant « Suivi paramètres »
  - Ajouter un pouls de :
    - Un pouls de 170
    - Une température de 37.7
    - A l'heure actuelle

Annexe C – Alertes concernant la dialyse

Exemple Calydial

- **Test 1 – Perte de poids**
- ***Observations simulées***
- Albert
  - Tension : 150 mm Hg
  - Poids : 62.4 Kg (poids de référence : 63.5 Kg)
- Charles
  - Tension : 135 mm Hg
  - Poids : 97.2 Kg (poids de référence : 99 Kg)
- ***Modifier le calcul des valeurs linguistiques***
- Ouvrir le composant [Equation d'alerte]
  - Modifier l'équation appelée « Poids et Tension »
  - Définir la zone d'incertitude entre le poids de référence « perte » et « stable » à [-2,-1.5]

****

F5 dans l'application

****

- ***Modifier les seuils de filtrage***
- Ouvrir le composant [Alerte]
  - Modifier la situation d'alerte « Perte de poids»
  - Aller dans la section « Options de filtrage des alertes »
  - Glisser les deux curseurs vers la gauche jusqu'au bout

- **Test 2 – prise de poids**
- ***Observations simulées***
- Brigitte
    - Tension : 157 mm Hg
    - Poids : 60.6 Kg (poids de référence : 59 Kg)
- ***Modifier le temps de validité à 100% est seuils de filtrage***
- Ouvrir le composant [Gestion de Variables]
    - Modifier la variable « Poids Référence »
        - Aller dans la section « Durée de validité »
            - Définir une durée de validité à 100% : 3 jours
            - Définir une durée maximale de validité : 7 jours
    - Modifier la variable « Tension (systolique) »
        - Aller dans la section « Durée de validité »
            - Définir une durée de validité à 100% : 3 jours
            - Définir une durée maximale de validité : 7 jours

****

F5 dans l'application

****

- **Test 3 – prise de poids**
- ***Observations simulées***
- Brigitte
    - Une première fois dans la journée
        - Tension : 157 mm Hg
        - Poids : 60.6 Kg (poids de référence : 59 Kg)
    - Une deuxième fois dans l'heure d'avoir été notifié

        - Tension : 161 mm Hg
        - Poids : 58.9 Kg (poids de référence : 59 Kg)
    - Une troisième fois dans l'heure d'avoir été notifié
        - Tension : 132 mm Hg
        - Poids : 60.2 Kg (poids de référence : 59 Kg)

- ***Modifier le cycle de vie de l'alerte***
- Ouvrir le composant [Alerte]
    - Modifier la situation d'alerte « Gain de poids»
    - Aller dans la section « Cycle de vie»
    - Aller au scénario avec deux flèches rouges
    - Changer la première option « Alerte si l'indice d'applicabilité augmente » et la deuxième option « Alerte si l'indice d'applicabilité diminue » par « non »

- **Test 4 – prise de poids**
- ***Observations simulées***
- Albert
    - Tension : 153 mm Hg
    - Poids : 62 Kg (poids de référence : 63.5 Kg)
- Charles
    - Tension : 136 mm Hg
    - Poids : 106 Kg (poids de référence : 99 Kg)
- Brigitte
    - Tension : 153 mm Hg
    - Poids : 60.2 Kg (poids de référence : 59 Kg)

- David
    - Tension : 157 mm Hg
    - Poids : 72 Kg (poids de référence : 69 Kg)
- ***Modifier le filtrage dans la liste d'alertes***
- Aller dans la liste d'alertes
    - Glissez doucement le curseur IA vers la droite ou appuyez sur le bouton en forme de flèche qui se trouve à droite du curseur

---

- **Test 5 – Emergence douteuse**

****

F5 dans l'application

****

- ***Observations simulées***
- Brigitte
    - Rougeur
- David
    - Ecoulement
- ***Visualisation des émergences***
- Charger « Brigitte»
    - Taper Brigitte dans la barre de recherche en bas et à gauche de l'application
    - Sélectionner Brigitte
- Ouvrir le composant [Scores]
- ***Modification de la situation d'alerte***
- Ouvrir le composant [Alerte]
    - Modifier la situation d'alerte « Emergence répétitive »

- Augmenter à ***2*** l'importance du facteur déclencheur « tendance Emergence »

- ***Visualisation des émergences***
- Charger « Brigitte»
  - Taper Brigitte dans la barre de recherche en bas et à gauche de l'application
  - Sélectionner Brigitte
- Ouvrir le composant [Scores]
- ***Visualisation des émergences***
- Fermez et ouvrez le composant [Scores]
- ***Visualisation des émergences***
- Fermez et ouvrez le composant [Scores]

Annexe D – Dictionnaire des concepts TEMPAS

Agrégation par détection (Définition 30) : Page 121

L'agrégation par détection construit une seule alerte à partir de toutes les alertes détectées ayant passé les filtrages par qualité et par expressivité.

Alerte accessible (Définition 26) : Page 116

Une alerte accessible est une alerte construite à partir d'une ou plusieurs alertes détectées.

Alerte détectée (Définition 5) : Page 84

Une alerte détectée est l'instanciation ou concrétisation après évaluation d'une situation d'alerte sur une entité observée à un instant τ.

Chemin le plus important (Définition 37) : Page 142

Le chemin le plus important est celui dont la somme des poids des arcs formant le chemin dont les observations sont connues est la plus grande.

Condition d'activation (Définition 3) : Page 83

Une condition d'activation sert à exprimer ce qui doit être satisfait afin de détecter une alerte. Elle est formée par une ressource observable, un opérateur, et une valeur.

Condition d'activation intrinsèque (Définition 11) : Page 91

Une condition intrinsèque exprime l'état dans lequel une ressource observable doit se trouver ou comment la tendance doit se comporter afin de pouvoir être considérée en état d'alerte.

Condition d'activation temporelle (Définition 12) : Page 92

Une condition temporelle exprime l'ordre d'occurrence des horodatages des observations ou des segments des ressources observables utilisés dans la construction de conditions d'activation intrinsèques.

Cycle de vie de l'alerte (Définition 31) : Page 122

Le cycle de vie de l'alerte est un mécanisme qui permet de contrôler la notification explicite d'alertes accessibles.

Degré d'appartenance (Définition 9) : Page 87

Le degré d'appartenance est une valeur entre zéro et un qui exprime le degré de satisfaction de la transformation d'une valeur quantitative en une valeur linguistique donnée.

Enrichissement par « écrasement » (Définition 35) : Page 125

L'enrichissement par « écrasement » remplace les propriétés des alertes accessibles plus récentes avec celles de l'alerte accessible la plus ancienne ou les propriétés des alertes accessibles plus anciennes avec celles de l'alerte accessible la plus récente.

Enrichissement par état (Définition 36) : Page 126

L'enrichissement par état hérite de l'état de l'alerte accessible notifiée précédemment

Enrichissement par qualité (Définition 34) : Page 124

L'enrichissement par qualité exprime s'il faut notifier explicitement une alerte si l'indice d'applicabilité ou de confiance de l'alerte augmente ou diminue.

Entité observable (Définition 1) : Page 83

Une entité observable correspond à l'objet ciblé par les situations d'alerte. Elle répond à la question, qui est observé ?

Etat de l'alerte (Définition 33) : Page 122

L'état de l'alerte exprime la manière dont l'utilisateur peut interpréter une alerte accessible.

Filtrage par expressivité linguistique (Définition 28) : Page 119

Le filtrage par expressivité linguistique choisit les alertes détectées dont les valeurs linguistiques expriment le mieux la situation d'alerte :

- celle dont la valeur linguistique est la plus inférieure et la condition d'activation intrinsèque est formée par l'opérateur « ≤ »,

- celle dont la valeur linguistique est la plus supérieure et la condition est formée par l'opérateur « ≥ »,

- ou celle dont la valeur linguistique est la même que celle utilisée dans la condition formée par l'opérateur « = ».

Filtrage par longévité de la tendance (Définition 29) : Page 120

Le filtrage par longévité de la tendance choisit les alertes détectées dont il n'existe pas un segment, autre que celui associé à la tendance, qui approxime la totalité des observations approximées par le segment associé à la tendance.

Filtrage par qualité (Définition 27) : Page 119

Le filtrage par qualité fixe deux seuils, un pour l'indice d'applicabilité de l'alerte (IAA) et un autre pour l'indice de confiance de l'alerte (ICA). Si les valeurs des indices dépassent les seuils, l'alerte détectée est considérée comme telle.

Graphe modificateur (Définition 16) : Page 97

Le graphe modificateur exprime, à travers des arcs, comment les valeurs linguistiques d'une ressource affectent les valeurs linguistiques d'une autre dans un contexte

Indice d'applicabilité d'une alerte (IAA) (Définition 24) : Page 112

L'indice d'applicabilité d'une alerte (IAA) exprime dans quelle mesure une entité (qui est observée) est concernée par une situation d'alerte

Indice de confiance d'une alerte (ICA) (Définition 25) : Page 113

L'indice de confiance d'une alerte exprime la confiance dans les observations et les segments utilisés lors de l'évaluation d'une situation d'alerte

Indice de confiance de l'observation (ICO) (Définition 17) : Page 99

L'indice de confiance de l'observation (ICO) exprime la confiance que le système peut avoir dans une observation d'une ressource r sous un contexte c.

Indice de confiance de la condition (ICC) (Définition 23) : Page 111

L'indice de confiance de la condition (ICC) exprime le degré de confiance que le système a dans les observations ou les segments utilisés pour vérifier la condition intrinsèque.

Indice de confiance de la tendance (ICT) (Définition 18) : Page 101

L'indice de confiance de la tendance (ICT) reflète la confiance que le système peut avoir dans le segment associé à la tendance d'une ressource r par rapport à la distance temporelle entre les observations approximées par le segment dans un contexte c.

Indice de satisfaction d'une condition intrinsèque (ISCI) (Définition 21) : Page 108

L'indice de satisfaction d'une condition intrinsèque (ISCI) exprime le degré avec lequel une valeur numérique satisfait une condition intrinsèque formée par une valeur linguistique

Indice de satisfaction d'une condition temporelle (ISCT) (Définition 22) : Page 109

L'indice de satisfaction d'une condition temporelle exprime si les horodatages des observations ou des segments respectent la contrainte temporelle « commence avant ou en même temps que »

Indice Linguistique (IL) (Définition 19) : Page 102

L'indice linguistique (IL) exprime le degré de certitude de l'utilisation de la valeur linguistique l pour représenter la valeur numérique v sous le contexte c.

Indice Linguistique Révisé (ILR) (Définition 20) : Page 103

L'indice linguistique révisé (ILR) exprime le degré de certitude d'utilisation de la valeur linguistique l_0 pour représenter la valeur numérique v_0 sous le contexte c étant donné l'existence d'autres valeurs linguistiques $l_1, \dots, l_{n+1}$ représentant des valeurs numériques $v_1, \dots, v_{n+1}$ des ressources $r_1, \dots, r_{n+1}$

Informations contextuelles (Raffinement de la Définition 7) Page 92 Page 93

Les informations contextuelles permettent de mieux définir une situation (d'alerte) pour une entité.

Interaction entre ressources observables (Définition 15) : Page 96

L'interaction entre ressources observables exprime comment les valeurs linguistiques d'une ressource affectent les valeurs linguistiques d'une autre.

Observation (Définition 6) : Page 85

Une observation est définie par une entité observée, une ressource observable (ce qui est observé), la date et l'heure à laquelle l'observation a eu lieu, et la valeur observée qui peut être quantitative ou linguistique.

Ordre des valeurs linguistiques (Définition 8) : Page 87

Les valeurs linguistiques sont naturellement ordonnées.

Ressource observable (Définition 2) : Page 83

Une ressource observable est tout ce qui peut être observé sur une entité observable. Elle répond à la question, qu'est-ce qui est observé ?

Segment (Définition 10) : Page 89

Nous interprétons un segment comme l'approximation de toutes les observations dont

Situation d'alerte (Définition 4) : Page 84

Une situation d'alerte exprime une situation d'intérêt. Dans le cas spécifique de TEMPAS, une situation d'alerte est définie comme l'addition de conditions d'activation.

Temps d'expiration (Définition 14) : Page 94

Le temps d'expiration (ψ) exprime après combien de temps une observation devient obsolète. Outre le fait qu'elle n'est plus actuelle, s'en servir pour évaluer une alerte peut introduire une erreur.

Temps de validité (Définition 13) : Page 94

Le temps de validité (φ) de la ressource observable exprime pendant combien de temps une observation est considérée comme étant actuelle et toujours fiable lors de son utilisation dans l'évaluation d'une alerte

Temps de validité de l'alerte (Définition 32) : Page 122

Le temps de validité de l'alerte exprime combien de temps après qu'une situation d'alerte a été identifiée, elle n'est plus d'actualité.

Valeurs linguistiques d'une ressource observable (Définition 7) : Page 86

L'état ou la tendance d'une ressource observable peut s'exprimer par des valeurs linguistiques. Une valeur linguistique est représentée par un ensemble flou de forme