



HAL
open science

Proximités sémantiques et contextuelles pour l'apprentissage en mobilité: application à la visite de musée

Pierre-Yves Gicquel

► **To cite this version:**

Pierre-Yves Gicquel. Proximités sémantiques et contextuelles pour l'apprentissage en mobilité: application à la visite de musée. Sciences de l'ingénieur [physics]. Université de Technologie de Compiègne, 2013. Français. NNT: 2013COMP2077 . tel-01069547

HAL Id: tel-01069547

<https://theses.hal.science/tel-01069547>

Submitted on 29 Sep 2014

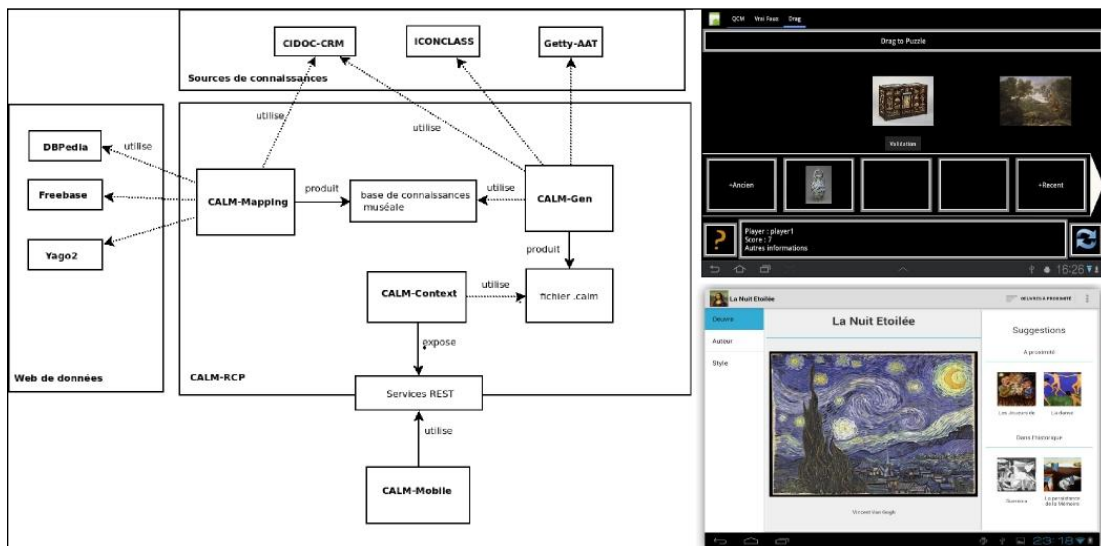
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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.

Par **Pierre-Yves GICQUEL**

Proximités sémantiques et contextuelles pour l'apprentissage en mobilité : application à la visite de musée

Thèse présentée
 pour l'obtention du grade
 de Docteur de l'UTC



Soutenue le 17 mai 2013
Spécialité : Technologies de l'information et des systèmes

D2077

THÈSE DE DOCTORAT

Thèse présentée pour obtenir le grade de Docteur de l'UTC
Champ disciplinaire : Technologies de l'Information et des Systèmes

PROXIMITÉS SÉMANTIQUES ET CONTEXTUELLES POUR L'APPRENTISSAGE EN MOBILITÉ : APPLICATION À LA VISITE DE MUSÉE

PIERRE-YVES GICQUEL

Soutenue publiquement le 17 mai 2013 devant un jury composé de :

<i>Rapporteurs</i>	Monique Grandbastien	Université de Lorraine
	Serge Garlatti	Télécom Bretagne
<i>Examineurs</i>	Sebastien George	INSA de Lyon
	Claude Moulin	Université de Technologie de Compiègne
	Philippe Trigano	Université de Technologie de Compiègne
	Laurence Tardy	Ecole du Louvre
<i>Directeur de thèse</i>	Dominique Lenne	Université de Technologie de Compiègne

Thèse réalisée au Laboratoire Heudiasyc
Université de Technologie de Compiègne
Centre de Recherches de Royallieu
60205 COMPIEGNE

Tél : +33 3 44 23 46 45
Fax : +33 3 44 23 44 77

Web : <http://www.hds.utc.fr/>

Sous la direction de Dominique Lenne dominique.lenne@utc.fr

Financement Allocation MESR

À Mamig et Tad-Coz

Remerciements

Je souhaite remercier en premier lieu mon directeur de thèse, Dominique Lenne, pour m'avoir donné une liberté de recherche et d'exploration importante tout en m'aidant, par nos entretiens quasiment quotidiens, à problématiser les idées les plus intéressantes et à garder des axes de recherches porteurs au long de ce travail. Sa rigueur et son exigence, tant dans la démarche scientifique que dans la façon d'exprimer précisément les concepts, m'ont beaucoup apporté au cours de ce travail. Je remercie également chaleureusement les rapporteurs de ce travail, Monique Grandbastien et Serge Garlatti, pour la relecture particulièrement attentive qu'ils ont faite de ce mémoire et leurs commentaires précis et constructifs. Je souhaite également remercier Claude Moulin, membre de mon jury et qui m'a aidé à plusieurs reprises au cours de discussions informelles par sa connaissance impressionnante des fondements mathématiques associés aux logiques de descriptions utilisées dans ce travail. Je remercie également Laurence Tardy, enseignante à l'école du Louvre qui a accepté de participer à ce jury et avec qui a été impliquée à plusieurs niveaux dans ce travail, tant au niveau que spécialiste en muséologie que comme médiatrice avec d'autres acteurs du monde muséal. Je remercie aussi chaleureusement Sebastien Georges pour avoir accepté de participer à ce jury et Philippe Trigano pour avoir accepté de présider ce jury.

Le travail réalisé pendant une thèse est en grande partie solitaire. J'ai eu la chance durant ce travail d'être entouré et accompagné par des personnes très proches que je souhaite également remercier. Tout d'abord mes parents Marie-Cécile Tilly et Yannig Gicquel qui m'ont accompagné et aidé tout au long de mes études et du chemin de la thèse. Ma soeur Laureline Gicquel qui a souvent été d'un grand réconfort. Mon grand-père Robert Gicquel qui m'a toujours poussé vers la curiosité scientifique et ma grand-mère Huguette Gicquel qui m'a également soutenu. Enfin ma tante Véronique Tilly qui a accepté de relire ce manuscrit pour m'aider dans certaines formulations et questions grammaticales. Enfin, je veux bien évidemment remercier ma compagne, Julie Macoin, qui m'a supporté dans tout les sens du terme durant les périodes difficiles comme les périodes agréables de ce travail. Son énergie, sa patience et sa gentillesse m'ont été d'une grande aide durant ces trois dernières années, et j'espère lui rendre la pareille dans les années à venir.

Enfin, je souhaite également remercier tous les collègues qui sont devenus des amis durant ces années et qui ont fait rimer ontologie avec journées déguisées du mardi.

Remerciements

Merci à Lieutenant Colonel pour la foule de sentiments qu'elle m'a inspirés, à Tangible Boy pour avoir enrichi ma ZPD, à Planif Girl pour la scénarisation de nos soirées, à Dodo pour Gradlu, à Ezechiel et Emilia pour m'avoir fait découvrir de nouveaux horizons, au stagiaire éternel pour sa gentillesse, à Pirate pour ce qu'il sait, à Semantic-Music Boy et sa compagne pour leur punch, à A2 pour le Shamrock, à Lydie m'avoir introduit auprès de ses connaissances, à FabC pour Mélissa, à Sebastien pour l'agrégation, aux Vincent's pour leur aide précieuse, et à tous les autres.

Résumé

L'essor des dispositifs mobiles a conduit à un nouveau paradigme d'Environnements Informatiques pour l'Apprentissage Humain (EIAH) : les environnements d'apprentissage ubiquitaires. Tout comme beaucoup d'EIAH classiques, les environnements d'apprentissage ubiquitaires permettent une médiation informatique entre l'apprenant et des connaissances propres au domaine d'apprentissage. Cependant, dans les EIAH ubiquitaires, cette médiation a lieu dans un contexte physique spécifique qui détermine en partie le déroulement des sessions d'apprentissage. Les objets de l'environnement physique, ainsi que les événements ou processus qui s'y déroulent, sont pris en compte pour fournir des informations adaptées ou proposer des interactions pertinentes.

Nous nous plaçons dans ce travail dans le cadre de l'apprentissage en musée, pour des visites libres ou en groupes scolaires organisés. Nous analysons en première partie les pratiques d'apprentissage en musée et en particulier l'apprentissage instrumenté par des dispositifs mobiles. Cette analyse nous permet d'introduire, d'une part, les notions de représentation et de manipulation automatique de connaissances utiles à notre travail et, d'autre part, les concepts propres aux systèmes informatiques sensibles au contexte qui nous intéresseront en particulier.

Nous proposons par la suite un modèle sémantique du patrimoine culturel permettant une description riche et complète des artefacts muséaux. L'apprentissage ubiquitaire nécessite une représentation adéquate du contexte. En nous inspirant des travaux de Zimmerman, nous proposons de représenter le contexte suivant plusieurs espaces contextuels, chaque espace contextuel étant associé à une ontologie. Ces espaces permettent de faire le lien entre les connaissances purement contextuelles et les connaissances issues de la modélisation du patrimoine culturel ; ils sont peuplés automatiquement en fonction des déplacements de l'utilisateur et de ses interactions avec le dispositif.

Nous définissons un processus d'opérationnalisation des modèles du patrimoine culturel et du contexte à l'aide de calculs de proximités sémantiques pour la génération automatique d'activités d'apprentissage. Afin de prendre en compte le contexte, nous définissons également des proximités contextuelles mesurant l'adéquation des activités générées au contexte de l'utilisateur. Pour permettre aux enseignants de spécifier des objectifs pédagogiques, nous proposons un formalisme de règles pédagogiques offrant deux modes de

Résumé

contrôle : un contrôle thématique et un contrôle contextuel. Ces derniers permettent de concilier la liberté d'exploration de l'environnement, indispensable dans le cadre d'un apprentissage ubiquitaire, avec un guidage pédagogique défini par l'enseignant.

Mots clés : Ontologies, apprentissage ubiquitaire, proximités sémantiques, musée, contexte.

Abstract

Semantic and Contextual Proximities for Mobile Learning: The Case Study of Museum Visits

Abstract :

The rise of mobile devices has led to a new paradigm of technology enhanced learning (TEL) environment: ubiquitous learning environment. Like many classic TEL environments, ubiquitous learning environments offer a computer based mediation between the learner and the learning domain. However, in ubiquitous environments, this mediation takes place in a specific physical context which partly determines the course of learning. The learner acts both in the physical and in the digital spaces. Objects in the physical environment, as well as events or processes that take place are taken into account to provide appropriate information or propose relevant interactions.

The field of application of this work is ubiquitous learning in museum, either in the case of informal learning (e.g. classic museum visit) or in the case of more formal school learning in museum. We first analyse learning practices in museum, especially using mobile devices. This analysis allows us to introduce, on the one hand, the concepts of knowledge representation used in our work and, on the other hand, the concepts specific to context sensitive systems.

We propose in the following a semantic model of cultural heritage allowing a rich and full description of museum artifacts. Ubiquitous learning requires an adequate representation of context. Drawing our inspiration from the work of Zimmerman, we propose to represent context according to several contextual spaces, each space being associated with an ontology. These spaces allow to relate the purely contextual knowledge and knowledge from the model of cultural heritage, they are automatically populated according to movements of the user and his interaction with the device.

We define a process for operationalizing the models of cultural heritage and context using semantic proximity computations for the automatic generation of learning activities. To take into account the context, we also define contextual proximity, measuring the adequacy of generated activities to the user context. To enable teachers to specify learning objectives, we propose a formalism of rules offering two modes of control: a thematic control and contextual control.

Abstract

Keywords : ontology, ubiquitous learning, semantic proximity, context.

Table des matières

Remerciements	7
Résumé	9
Abstract	11
Table des matières	i
Introduction	1
Problématique	2
Cadre théorique	3
Approche proposée	4
Organisation du mémoire	4
1 L'apprentissage en musée	7
1.1 Naissance et développement des musées	7
1.1.1 Des trésors aux musées modernes	8
1.1.2 Mécanismes d'expression du discours muséal	9
1.1.2.1 Organisation physique de l'espace	10
1.1.2.2 Matériaux écrits	10
1.1.2.3 Visite guidée et audioguide	11
1.2 Musées et publics : la didactique muséale	12
1.2.1 Approches pour l'étude des publics muséaux	12
1.2.1.1 Etudes sociologiques : publics des musées et leurs motivations	12

TABLE DES MATIÈRES

1.2.1.2	Etude des comportements : comment fréquente-t-on le musée?	14
1.2.1.3	Etudes cognitivistes : comment comprend-on le musée?	16
1.2.2	Apprendre en musée?	16
1.3	Conclusion : éléments d'un modèle du contexte en musée	17
Etat de l'art		19
2 Représentation des connaissances et métriques sémantiques		21
2.1	La représentation des connaissances	22
2.1.1	Représentation des connaissances et langages formels	22
2.1.2	Ontologies et web sémantique	25
2.1.2.1	Définitions	25
2.1.2.2	Les langages ontologiques du web sémantique	26
2.1.3	Interaction avec des connaissances formalisées : les systèmes à base de connaissances	28
2.1.4	Conclusion sur la représentation des connaissances	29
2.2	Mesures de similarités et proximités sémantiques	30
2.2.1	Définition	30
2.2.1.1	Similarité et proximité	31
2.2.2	Mesures de similarités et de proximités	31
2.2.2.1	Approches basées sur les hiérarchies de concepts	32
2.2.2.2	Approches basées sur les propriétés	34
2.2.2.3	Approches basées sur des corpus	37
2.2.3	Conclusion sur les mesures de similarités et de proximités	38
3 Représentation informatique du contexte		41
3.1	Définitions du contexte	42
3.2	Représentation et traitement des informations contextuelles	43
3.2.1	Stratégies de modélisation du contexte	44
3.2.2	Opérationnalisation des modèles contextuels	47

3.3	Informatique ubiquitaire et systèmes sensibles au contexte	51
3.3.1	Architecture des systèmes sensibles au contexte	51
3.3.2	Interaction avec les systèmes sensibles au contexte	52
3.3.3	Systèmes sensibles au contexte pour l'apprentissage humain	55
3.4	Conclusion	58
4	Systèmes sensibles au contexte pour l'assistance à la visite de musées	59
4.1	Approches pour l'assistance à la visite de musées	60
4.1.1	Approches orientées tâche	60
4.1.2	Approches orientées navigation	63
4.2	Systèmes basés sur les connaissances pour l'assistance à la visite de musée	66
4.2.1	Systèmes basés sur les connaissances	66
4.2.2	Limites des systèmes basés sur les connaissances	68
4.2.2.1	Limite en termes d'interaction	68
4.2.2.2	Limites pédagogiques	69
4.3	Conclusion	69
	Notre approche	71
5	Représentation sémantique du domaine et du contexte	73
5.1	Modélisation sémantique du domaine	74
5.1.1	Sources de connaissances utilisées	74
5.1.2	Modifications du modèle de CIDOC-CRM	78
5.1.3	Modèle sémantique des œuvres : intégration des taxonomies dans CIDOC-CRM	81
5.1.4	Conclusion sur la modélisation sémantique du domaine	83
5.2	Modélisation sémantique du contexte	83
5.2.1	Structure du modèle sémantique du contexte	84
5.2.2	Les espaces contextuels	85
5.2.3	Illustration	88
5.3	Conclusion	89

Notation utilisées	91
6 Proximités sémantiques et contextuelles pour l'interaction	93
6.1 Construction d'une mesure de proximité entre instances	94
6.1.1 Propriétés et concepts d'intérêt	96
6.1.2 Vecteur de proximités entre instances	97
6.1.2.1 Cas où les valeurs de la propriété sont des instances or- ganisées hiérarchiquement	98
6.1.2.2 Cas où les valeurs de la propriété sont des instances non hiérarchisées	99
6.1.2.3 Cas où les valeurs de la propriété sont de type littéral . .	100
6.1.2.4 Cas des liens directs entre instances	101
6.1.2.5 Agrégation du vecteur de proximités	101
6.1.2.6 Utilisation de la récursion pour la forme définitive du vecteur numérique de proximités	102
6.1.3 Explicitation des proximités	104
6.1.3.1 Cas où les valeurs de la propriété sont de type instance non hiérarchisée	104
6.1.3.2 Cas où les valeurs de la propriété sont de type instance hiérarchisée	105
6.1.3.3 Cas où les valeurs de la propriété sont de type littéral . .	107
6.1.3.4 Cas des liens directs entre les instances à comparer . . .	107
6.1.3.5 Utilisation de la récursion pour la forme définitive de l'explicitation des proximités	107
6.2 Construction de mesures de proximités contextuelles	109
6.2.1 Cas du contexte de localisation et du contexte historique	110
6.2.2 Cas du contexte d'intérêt	111
6.3 Interactions contextualisées	112
6.3.1 Génération de suggestions	112
6.3.2 Comparaison des instances	112
6.3.3 Calcul des taux d'adéquations au contexte, choix des instances à présenter	114
6.3.4 Génération des explicitations	114

6.3.5	Génération d'activités d'auto-évaluation	117
6.3.5.1	Activités de type QCM	117
6.3.5.2	Activités de classification	120
6.3.6	Annotations contextualisées	121
6.3.6.1	Modèle de l'annotation contextualisée	122
6.3.6.2	Proximités sémantiques et contextuelles entre annotations	122
6.3.6.3	Navigation et visualisation des annotations	126
6.4	Conclusion	128
7	Modes de contrôle du système	129
7.1	Contrôle du système par l'enseignant : intégration de connaissances pédagogiques	130
7.1.1	Contrôle thématique	130
7.1.2	Contrôle contextuel	132
7.1.2.1	Règles de contrôle contextuel	132
7.1.2.2	Règles de génération contextuelle	133
7.1.3	Conclusion sur l'intégration de connaissances pédagogiques	136
7.2	Contrôle du système par l'expert : expression des proximités	137
7.2.1	Présentation informelle du problème : insuffisance de la moyenne pondérée	137
7.2.2	Formalisation : utilisation de l'intégrale de Choquet	138
7.2.2.1	Importance et interaction des critères	140
7.2.3	Ajouts de contraintes liées à l'ontologie pour l'identification des capacités	141
7.2.3.1	Type de contraintes envisageables	141
7.2.3.2	Classement des critères suivant leur importance	141
7.2.3.3	Signes des indices d'interaction	142
7.2.4	Intégration des préférences de l'expert et construction de la capacité	144
7.2.5	Illustration	144
7.2.6	Conclusion sur l'intégration des connaissances de l'expert en muséologie	146
7.3	Conclusion	146

Implémentation et évaluation	147
8 La plateforme CALM	149
8.1 Scénarios d'utilisation	149
8.1.1 Utilisation par un visiteur seul	150
8.1.1.1 Navigation parmi les connaissances associées aux œuvres	150
8.1.1.2 Activités d'auto-évaluation et d'annotation	153
8.1.2 Utilisation par un groupe scolaire accompagné d'un enseignant . .	156
8.1.2.1 Préparation de la visite	157
8.1.2.2 Visite du musée	158
8.1.2.3 Après la visite	158
8.2 Réalisation de l'application <i>CALM-Mobile</i>	159
8.2.1 L'application <i>CALM-Mobile</i>	159
8.2.2 Communication avec le serveur via l'API RESTLet	160
8.2.3 Problèmes rencontrés	161
8.3 Présentation des applications serveur	162
8.3.1 Construction de la base de connaissances muséales : l'application <i>CALM-Mapping</i>	163
8.3.1.1 Objectif de l'application	163
8.3.1.2 Description de l'application	164
8.3.1.3 Réalisation	166
8.3.2 Génération des suggestions et des jeux d'auto-évaluation : <i>CALM-Gen</i>	167
8.3.2.1 Objectif de l'application	167
8.3.2.2 Description de l'application	167
8.3.2.3 Réalisation	169
8.3.3 Communication avec l'application Android : <i>CALM-Context</i> . . .	169
8.3.3.1 Réalisation	169
8.4 Conclusion	170

9	Expérimentations et validation	171
9.1	Sources de connaissances utilisées	172
9.2	Evaluation des suggestions et des jeux	174
9.2.1	Pertinence des suggestions	174
9.2.2	Evaluation des jeux d'auto-évaluation	175
9.3	Evaluation de l'utilisabilité de l'application mobile	176
9.4	Evaluation des performances	177
9.4.1	Méthodes et résultats de l'évaluation des performances	178
9.4.2	Prise en compte des résultats dans la conception du système	178
9.5	Conclusion	179
10	Conclusion et perspectives	181
10.1	Rappel des objectifs	182
10.2	Apports de la thèse	182
10.2.1	Modélisation du patrimoine et du contexte	182
10.2.2	Proximités sémantiques et contextuelles pour l'interaction contextualisée	183
10.2.3	Modes de contrôle du système	184
10.3	Perspectives	185
10.3.1	Perspectives à court terme	185
10.3.2	Perspectives à moyen et long terme	186
Annexes		189
A	L'intégrale de Choquet comme opérateur d'agrégation	191
A.1	Définition (rappel)	191
A.2	Caractérisation : indices d'importance et d'interaction, k-additivité	192
A.3	Utilisation de R pour déterminer des capacités	193
B	Architecture d'une application Android	195

TABLE DES MATIÈRES

C	Utilisation d'Eclipse-RCP pour les interfaces utilisateurs	197
C.1	Eclipse RCP	197
C.2	Illustration	198
D	L'application CALM-Gen	201
E	Questionnaire de satisfaction : chateau de Compiègne	203
	Bibliographie	205

Introduction

L'essor des périphériques mobiles a conduit à un nouveau paradigme d'Environnements Informatiques pour l'Apprentissage Humain : les environnements d'apprentissage ubiquitaires. Tout comme beaucoup d'EIAH classiques, les environnements d'apprentissage ubiquitaires permettent une médiation informatique entre l'apprenant et des connaissances propres au domaine d'apprentissage. Cependant, dans les EIAH ubiquitaires, cette médiation a lieu dans un contexte physique spécifique qui détermine en partie le déroulement des sessions d'apprentissage. L'apprenant agit parallèlement dans l'espace physique et dans l'espace numérique. Les objets de l'environnement physique, ainsi que les événements ou processus qui s'y déroulent, sont pris en compte pour fournir des informations adaptées ou proposer des interactions pertinentes.

La spécificité de l'apprentissage ubiquitaire, du point de vue informatique, tient au modèle dynamique du contexte qu'il nécessite. Dans le cadre des EIAH non ubiquitaires, l'apprenant interagit de manière explicite avec un dispositif informatique pour construire son apprentissage. Les environnements d'apprentissage ubiquitaires, *via* un modèle dynamique du contexte, permettent des interactions explicites et implicites. Typiquement, les déplacements de l'apprenant sont pris en compte pour lui proposer des interactions pertinentes en fonction de sa position. Le déplacement de l'apprenant constitue une interaction implicite, capturée par le modèle dynamique du contexte.

La mise en place d'un environnement d'apprentissage ubiquitaire nécessite donc une représentation duale des connaissances : les connaissances propres au domaine d'apprentissage lui-même et les connaissances propres au contexte, reflet de l'état du monde autour de l'apprenant. La difficulté est de relier ces deux modèles de connaissances afin de fournir des informations ou des possibilités d'interaction adaptées à la fois au domaine d'apprentissage et à l'état du contexte.

L'objet de ce travail est de concevoir des modèles sémantiques pour l'apprentissage ubiquitaire et de définir des processus et une architecture informatique permettant la génération d'activités d'apprentissage à partir de ces modèles. L'approche que nous proposons dans la suite a donné lieu à la réalisation de la plateforme CALM. Elle a été appliquée à l'apprentissage ubiquitaire dans le cadre de la visite de musée.

Problématique

Problématique applicative

Notre travail vise à instrumenter, sur un dispositif mobile, des situations de visite de musée. Nous considérons deux types de situation : la visite libre d'un musée et la visite scolaire à but pédagogique. Dans les deux cas, il s'agit d'adapter, de manière transparente pour l'apprenant, les activités et interactions proposées en fonction de ses désirs et de sa situation physique (lieu, œuvre à proximité...). Cette adaptation implique une modélisation fine du domaine d'apprentissage et du contexte de la situation du visiteur ainsi que la conception d'un mécanisme d'adaptation permettant de proposer des informations ou des activités de manière appropriée durant la visite. Dans le cadre d'une visite scolaire, il est de plus souhaitable de proposer des modes de contrôle à l'enseignant lui permettant de spécifier des objectifs pédagogiques.

Notre objectif est donc de déterminer les modèles permettant la représentation sémantique des connaissances patrimoniales et de construire une représentation informatique du contexte de visite, s'adaptant à la situation du visiteur de manière transparente. Nous cherchons également à déterminer de nouveaux types d'interactions entre le visiteur et les œuvres, afin d'instrumenter son apprentissage.

Problématique théorique

Deux principaux problèmes, liés à la représentation des connaissances, ont orienté notre travail. Le premier problème concerne l'articulation de la représentation sémantique d'un domaine avec la représentation du contexte. Le deuxième problème concerne la génération de propositions d'interaction et d'activités d'apprentissage à partir d'une représentation sémantique du domaine et du contexte.

Le premier problème tient aux différences entre la représentation statique des connaissances d'un domaine et la représentation dynamique des connaissances contextuelles permettant la description de la situation réelle dans laquelle se trouve l'apprenant. Notre objectif est de déterminer les formalismes de représentation les plus adaptés pour la description de ces deux catégories de connaissances, les connaissances du domaine et les connaissances contextuelles, et de définir les liens entre ces deux types de connaissances. Nous devons également définir les modalités de peuplement des bases de connaissances associées au modèle du contexte, c'est-à-dire les moyens de transformer les informations issues des dispositifs pervasifs (ex. dispositif de localisation de l'utilisateur) en connaissances contextuelles.

Le deuxième problème est lié de manière plus directe à la problématique applicative. Notre objectif est d'offrir des informations et des propositions d'interaction permettant

l'apprentissage en musée. A cette fin, nous devons définir des opérations sur nos modèles de connaissances permettant de générer effectivement des propositions d'interaction adaptées à l'utilisateur en vue d'un apprentissage. L'utilisation de représentations sémantiques du domaine est assez répandue pour assister des activités d'apprentissage. Cependant, l'utilisation conjointe d'une représentation sémantique du domaine et d'une représentation sémantique du contexte pour proposer des interactions contextualisées a été peu étudiée. Nous souhaitons générer des activités d'apprentissage à partir d'un modèle sémantique du domaine et déterminer l'adéquation de ces activités au contexte de l'utilisateur afin de lui proposer des activités pertinentes sur le plan du domaine d'apprentissage et sur le plan contextuel.

Cadre théorique

Notre cadre théorique mobilise plusieurs domaines de recherche : EIAH, ingénierie des connaissances, interaction homme machine. Il est centré sur la représentation des connaissances, plus particulièrement sur la représentation sémantique des connaissances contextuelles, ainsi que sur les moyens à mettre en œuvre pour la génération et le contrôle d'activités d'apprentissage à partir de représentations sémantiques. Ce travail étant appliqué au cadre de la visite de musée, nous nous sommes également penché sur l'apprentissage en musée, en particulier lorsque cet apprentissage est instrumenté par un dispositif mobile.

En ce qui concerne la représentation du contexte, nous avons considéré les définitions et les propositions d'opérationnalisation du contexte proposées par [Abowd *et al.*, 1999], [Zimmermann *et al.*, 2007] et [Dourish, 2004]. Nous nous sommes intéressés aux différentes techniques de modélisation du contexte : numériques [Choi *et al.*, 2005], logiques [Ranganathan *et al.*, 2002] et plus particulièrement sémantiques [Haake *et al.*, 2010], [Ye *et al.*, 2007].

Afin de générer de manière automatique des propositions d'interactions, nous nous sommes principalement penché sur les calculs de proximités et de similarités sémantiques. Nous avons considérés les approches basées sur la structures hiérarchique des ontologies [Rada *et al.*, 1989],[Wu & Palmer, 1994], les approches basées sur les propriétés [Aimé *et al.*, 2011],[Anyanwu *et al.*, 2005] ainsi que les approches basées sur des corpus [Lin, 1998],[Mihalcea *et al.*, 2006]. Les proximités et similarités sémantiques étant parfois trop contraintes pour retourner des résultats, nous nous sommes intéressés aux techniques de relaxations sémantiques [Hurtado *et al.*, 2006].

Nous avons pris en compte les travaux concernant l'utilisation de technologies mobiles pour l'aide à la visite de musée. Nous avons considéré deux types de travaux : les travaux centrés sur l'aide à la visite informelle de musée, [Oppermann & Specht, 2000, Wang *et al.*, 2009, Damala, 2009] et les travaux centrés sur les visites à but pédagogiques [Anderson *et al.*, 2002, Thom-Santelli *et al.*, 2005].

Enfin, d'un point de vue pédagogique, nous nous sommes intéressés aux travaux sur l'apprentissage authentique [Brown *et al.*, 1989] et l'apprentissage situé [Lave & Wenger, 1991]. Nous nous sommes en particulier penchés sur les liens entre l'apprentissage situé et l'apprentissage en mobilité [Pachler, 2010].

Approche proposée

Afin de proposer des activités pertinentes dans le cadre d'une visite de musée, nous avons étudié les pratiques d'apprentissage en musée, et en particulier l'apprentissage instrumenté par des dispositifs mobiles. Nous nous basons sur une adaptation de formalismes sémantiques de description du patrimoine culturel afin de proposer un modèle sémantique riche des œuvres. En nous inspirant des travaux de [Zimmermann *et al.*, 2007], nous avons choisi de représenter le contexte suivant plusieurs espaces contextuels, chaque espace contextuel étant associé à une ontologie. Ces espaces permettent de faire le lien entre les connaissances purement contextuelles et les connaissances issues de la modélisation du domaine ; ils sont peuplés automatiquement en fonction des déplacements de l'apprenant et de ses interactions avec le dispositif.

Nous proposons un processus d'opérationnalisation des modèles du domaine et du contexte à l'aide de calculs de proximités sémantiques pour la génération d'activités d'apprentissage. Afin de prendre en compte le contexte dans la génération d'activités, nous proposons des calculs de proximités contextuelles mesurant l'adéquation des activités générées au contexte de l'apprenant. Pour permettre aux enseignants de spécifier des objectifs pédagogiques, nous définissons un formalisme de règles pédagogiques offrant deux modes de contrôle : un contrôle thématique et un contrôle contextuel. Ces modes de contrôle permettent de concilier une certaine liberté d'exploration de l'environnement, indispensable dans le cadre d'un apprentissage ubiquitaire, avec un guidage pédagogique défini par l'enseignant.

Organisation du mémoire

Notre mémoire est organisé de la manière suivante : le premier chapitre décrit notre domaine d'application , l'apprentissage informel en musée.

Le mémoire est ensuite organisé en trois grandes parties, la première partie consiste en un état de l'art des domaines couverts par ce travail. Elle comporte trois chapitres. Le chapitre 2 décrit la représentation formelle de connaissances et les métriques utilisables sur ces représentations. Le chapitre 3 traite de la représentation informatique du contexte et des systèmes informatiques ubiquitaires. Enfin, le chapitre 4 détaille les différentes approches et systèmes utilisés pour l'apprentissage en musée.

La deuxième partie du mémoire présente nos contributions. Nous présentons dans le chapitre 5 les modèles que nous avons créés pour la représentation riche des œuvres du contexte de visite. Nous détaillons dans le chapitre 6 les métriques sémantiques et contextuelles que nous avons définies sur ces modèles pour la génération d'activités. Le chapitre 7 présente différents modes de contrôle, permettant notamment d'intégrer des connaissances didactiques dans la génération d'activité.

La troisième partie du mémoire présente les réalisations techniques et la validation de nos résultats. Le chapitre 8 décrit en détail la plateforme CALM. Le chapitre 9 décrit le processus de validation de l'approche et présente les expérimentations réalisées.

Chapitre 1

L'apprentissage en musée

Notre domaine d'application est le support numérique à l'apprentissage informel en musée. La vision des musées en tant que lieux de "délectation des publics" (Colbert) [Gob & Drouguet, 2006], puis comme lieux d'apprentissage est relativement récente. La grande variété des musées actuels, de leurs collections et de leurs modes d'organisation se comprend mieux suivant une perspective historique. Nous diviserons donc ce chapitre en deux sections. La première section est organisée suivant un plan historique et met en avant les évolutions sociales et techniques ayant permis d'aboutir à la conception actuelle du musée, avec en particulier la place des NTIC (Nouvelles Technologies de l'Information et de la Communication) pour l'organisation de l'accès à la culture.

La deuxième section nous permettra d'introduire des éléments plus spécifiques à notre recherche. Nous présenterons les modèles d'études permettant de capturer les variabilités (sociale, comportementale, d'expertise...) des visiteurs, et les formes que prend l'apprentissage suivant ces variabilités. Ces modèles serviront de base à nos propositions pour la modélisation du contexte de la visite de musée.

1.1 Naissance et développement des musées

La définition du musée qui fait référence est celle de l'ICOM (International Council of Museum) : "Un musée est une institution permanente sans but lucratif au service de la société et de son développement, ouverte au public, qui acquiert, conserve, étudie, expose et transmet le patrimoine matériel et immatériel de l'humanité et de son environnement à des fins d'études, d'éducation et de délectation"¹. Cette définition met en avant les trois fonctions principales du musée moderne : la conservation, la recherche et l'exposition. Nous retraçons ici les évolutions qui ont mené à cette conception moderne du musée.

1. <http://icom.museum/la-vision/definition-du-musee/L/2/>

1.1.1 Des trésors aux musées modernes

Les premières mentions de collections remontent au VII^e siècle avant notre ère. Ces collections n'étaient pas accessibles au public et étaient réservées aux riches familles. L'objectif de ces collections était double : d'une part étaler la puissance et la richesse, d'autre part manifester concrètement le lien entre les puissants et leurs ancêtres. La collection était la manifestation physique de la continuité du pouvoir. La collection est alors un trésor, un butin. La notion d'herméneutique, d'analyse savante des objets de la collection est absente. Cependant, la notion de patrimoine, en tant qu'ensemble de traces matérielles du passé digne d'être protégé pour les générations futures, transparaît dans ces premières collections. L'objectif premier de la collection est en effet, paradoxalement, sa transmission. Cette notion de patrimoine sera formalisée plus tard et donnera naissance aux musées modernes.

Le Moyen Âge européen voit se développer un nombre important de lieux d'admiration de "merveilles" à caractère religieux. Ces merveilles, essentiellement placées sur les chemins de pèlerinages, sont également appelées « trésors » et ont, outre leur réalité matérielle, une extension spirituelle. Ces merveilles ont créé les premiers efforts de mise en scène et d'interprétation de l'objet matériel, sous forme d'un discours religieux. Par ailleurs, ces merveilles sont, pour la plupart, accessibles publiquement. Le rôle de la trace matérielle dans l'édification du public apparaît donc très nettement.

C'est à la Renaissance que va se développer une conception "moderne" de l'organisation des traces matérielles. Des "cabinets de curiosités" naissent à travers toute l'Europe et avec eux l'ère des "connoisseurs". Ces cabinets, s'ils peuvent se contenter d'être uniquement des "collections", amas ou accumulation d'objets présentant quelques particularités historique, théorique ou autre, font rapidement l'objet d'une ébauche de théorisation qui formera les bases de la muséologie. Les premiers manuels sur la manière d'organiser les collections apparaissent ainsi au milieu du XVI^e siècle. Ces cabinets restent cependant des objets privés, et, si des visiteurs y sont admis, c'est au bon vouloir de leurs propriétaires.

La fin du XVII^e siècle et le début du XVIII^e sera l'âge de la création des musées publics. Par des legs de collections, des confiscations et des rachats, le patrimoine culturel arrive aux mains des états, l'exemple le plus flagrant étant la confiscation des biens du clergé à la Révolution française. La création des grands musées publics est également l'occasion de réflexions plus théoriques sur l'organisation des collections. D'une esthétique baroque où les œuvres d'écoles et de styles différents étaient disposées pour créer la « délectation artistique », on arrive à des dispositions par artiste ou par chronologie. Ces classifications font écho à l'effort général de systématisation dans les sciences naturelles notamment par Buffon et Linné.

Le XIX^e est marqué par d'importantes créations de nouveaux musées. La tendance est à l'éclectisme avec l'apparition des musées de sciences et techniques (Conservatoire des

Arts et Métiers en 1799), et le développement de musées locaux un peu partout en Europe. Ce siècle est aussi celui du nationalisme avec la création partout en Europe de musées exaltant « l'identité nationale ». Les efforts de systématisation et de classification entamés au siècle précédent se poursuivent avec les classements des peintures par écoles dans les musées d'art et des espèces par genres dans les musées d'histoire naturelle.

Le XX^e siècle verra se produire plusieurs bouleversements tant dans le rôle du musée que dans sa perception auprès du public. L'utilisation des musées pour la propagande nazie pendant la Seconde Guerre mondiale, ainsi que le mouvement de décolonisation des années 1960, alimenteront une forte critique de l'institution muséale. Parallèlement, les mouvements de collaboration internationale se développent et donnent naissance à l'ICOM en 1947. Le milieu et la fin du XX^e siècle voient se développer les recherches sur les publics, avec les études sociologiques de Bourdieu en 1960 et l'introduction de la notion d'évaluation des expositions en 1976. Le comportement du visiteur, sa réception du discours muséal sont scrutés et aboutissent à une nouvelle perception du visiteur comme acteur de sa visite et de son interprétation. L'introduction du multimédia à la fin du XX^e siècle, au début sous forme de bornes interactives, marque une nouvelle étape dans ce processus visant à rendre le visiteur acteur de sa visite, ce dernier étant libre d'explorer comme il le souhaite les collections sur ces bornes.

Cette brève remise en contexte historique de l'histoire du musée nous permet de dégager deux tendances importantes pour notre problématique applicative. La première tendance, initiée au XVII^e siècle, concerne l'effort de classification systématique des collections muséales. Ce travail, correspondant à la fonction de recherche des musées, était à l'origine basé sur des systèmes de fiches. Cependant dès 1960 les premières communautés dédiées à l'informatisation des fiches muséales apparaissent [Marty & Jones, 2012]. Les modes d'indexation et de classification actuels font de plus en plus appel aux technologies issues de l'ingénierie des connaissances, comme le montre par exemple la création récente d'un *endpoint* SPARQL permettant de requêter les collections du *British Museum*². Il existe donc déjà des modèles issus de l'ingénierie des connaissances pour la description du patrimoine culturel, comme par exemple l'ontologie CIDOC-CRM [Doerr, 2003]. Nous utiliserons dans notre travail ces ressources existantes. La deuxième tendance est d'attirer un public de moins en moins expert et de rendre actif ce public dans le musée. Notre travail se situe dans la continuité de ce mouvement en proposant au visiteur des interactions nouvelles avec les œuvres, basées sur des modèles de connaissances formelles.

1.1.2 Mécanismes d'expression du discours muséal

Le rôle d'exposition du musée ne se limite pas à fournir un accès public aux œuvres. Il implique une organisation de cet accès permettant aux visiteurs d'atteindre une certaine

2. <http://collection.britishmuseum.org/Sparql>

compréhension des objets exposés, de leurs relations, des problématiques historiques, artistiques ou sociales qui émergent du choix particulier des œuvres exposées.

La mise en place d'une exposition pour un conservateur ne se réduit pas à choisir des œuvres particulièrement intéressantes, il s'agit plutôt d'organiser un discours à travers et pour ces œuvres. Nous décrivons ici les mécanismes d'expression de ce discours durant la visite. Il est à noter que, de plus en plus, les musées offrent des possibilités d'accès au catalogue via internet, permettant un accès au discours avant et après la visite.

1.1.2.1 Organisation physique de l'espace

L'organisation de l'espace reflète, de manière implicite, le sens que le conservateur veut donner à l'exposition [Poulot, 2010]. Plusieurs possibilités classiques existent pour l'organisation des objets : chronologique, géographique, classification taxinomique (ex. par école). C'est de cette organisation qu'émerge le message de l'exposition. Une organisation chronologique sera par exemple adaptée dans les musées d'histoire ou d'archéologie.

Le parcours dans son ensemble, c'est-à-dire du début de la visite à la sortie du musée, peut également être porteur de sens. Les musées imposant un parcours déterminé peuvent ainsi mettre en place une véritable narration, alimentée par les objets exposés. Cependant imposer un parcours est une contrainte forte pour les visiteurs, qui aiment à faire usage de leur liberté de visiter [Gob & Drouguet, 2006].

Avec l'émergence des technologies numériques, de nouveaux modes d'organisation des parcours apparaissent sous forme d'espaces où sont situées des bornes permettant aux visiteurs d'en savoir plus sur les œuvres exposées. Ces espaces intermédiaires, ni vraiment au contact des œuvres, ni vraiment en dehors du musée, peuvent devenir des lieux d'échanges entre visiteurs, favorisant une sociabilisation chère aux visiteurs novices.

1.1.2.2 Matériaux écrits

La place de l'écrit dans le musée fait encore débat entre les tenants d'une « muséologie de l'objet », visant à minimiser les informations fournies aux visiteurs pour leur permettre de construire seuls leurs expériences muséales, et une « muséologie de l'idée » pour laquelle les objets n'ont pas de sens intrinsèque et où les matériaux écrits explicitent le sens de l'exposition [Poulot, 2001]. Cependant, les expositions sans aucun texte sont l'exception, le texte offrant une aide immédiate à la compréhension des objets et des expositions.

Le problème essentiel du texte dans le musée est que les textes ne sont pas lus dans un ordre donné, comme c'est le cas dans une publication. De plus, moins d'un tiers des textes en moyenne est lu [Gob & Drouguet, 2006]. Les textes doivent donc être indépendants les uns des autres. Par ailleurs, afin de permettre au visiteur de sélectionner rapidement ce qui l'intéresse, les textes doivent présenter une certaine hiérarchie, les titres de salles

ou d'exposition étant très visibles et les ressources plus spécifiques (ex. « pour en savoir plus ») moins visibles.

La muséographie moderne propose souvent plusieurs niveaux de lecture d'une salle. Les titres d'expositions ou de salles, très visibles, donnent des informations très générales. Les cartels des oeuvres donnent une brève description (titre de l'œuvre, artiste). Les informations détaillées sont situées un peu en retrait et précisent des éléments historiques ou techniques.

1.1.2.3 Visite guidée et audioguide

Nous regroupons ces deux modes d'expression du discours muséal car ce sont deux modes d'expressions construits hors de la scénographie. L'organisation des textes et des œuvres est figée et propre au musée, les textes faisant partie de la scénographie, tandis que la visite guidée et l'audioguide vont ajouter de nouveaux éléments d'information ou d'animation à cette scénographie.

La visite guidée offre, au prix d'un parcours imposé, une plus grande adaptation au public (c'est le rôle du guide conférencier) et surtout un espace social d'échange entre les visiteurs (échanges entre les visiteurs et le guide et entre les visiteurs eux-mêmes). Le discours du guide ne suit pas nécessairement le discours imaginé par le conservateur et s'adapte aux attentes et aux intérêts du public, offrant une « animation » au sein du musée.

Aujourd'hui, la plupart des musées proposent au moins des audioguides. Ces derniers fonctionnent, en terme de contrainte, à l'inverse de la visite guidée : au prix d'un discours préconçu et d'une faible interaction avec les autres visiteurs, les audioguides offrent une liberté totale dans les mouvements du visiteur.

Conclusion sur l'organisation du discours muséal

Le musée n'est pas un terrain vierge. Au contraire, plusieurs mécanismes éprouvés permettent l'accès effectif aux œuvres et aident à leur interprétation. Les nouvelles modalités d'interaction avec les œuvres n'ont donc pas à se calquer sur ces solutions mais plutôt offrir une complémentarité prenant en compte l'existant. Ainsi, l'organisation de l'espace et du parcours, porteuse de sens, devra être prise en compte pour le choix des interactions à proposer. La question de l'indépendance des textes et de leur hiérarchie devra également être adaptée au cadre de nouvelles interactions. Nous détaillons dans le chapitre 4 les solutions techniques existantes pour ces nouveaux systèmes.

1.2 Musées et publics : la didactique muséale

1.2.1 Approches pour l'étude des publics muséaux

Le rôle d'exposition du musée, dans sa dimension didactique, ne peut se concevoir qu'en connaissant précisément les récepteurs du discours muséal. Différents modes d'étude et de classification des publics ont pour cela été mis en place. Nous présentons ici les principaux modes d'étude des publics, répondant aux questions suivantes : quels sont les publics des musées ? Comment fréquentent-ils le musée ? Comment comprennent-ils le discours muséal ? Quelles sont les variabilités intra et inter-individuelles dans la visite du musée et la réception du discours muséal ?

1.2.1.1 Etudes sociologiques : publics des musées et leurs motivations

Les études sociologiques sont certainement les plus courantes. Ces études ont essentiellement pour objet de déterminer les profils et les motivations de visite des « visiteurs libres », c'est-à-dire à l'exclusion des publics scolaires. L'objectif de ces études est double : d'une part connaître les profils des visiteurs actuels, leurs motivations et leurs modes de réceptions, d'autre part identifier les segments de populations n'allant rarement ou jamais dans les musées afin de remédier à ces situations [Hirzy, 1992].

L'étude de [Bourdieu & Darbel, 1969] est une des premières références sur la sociologie des musées. Cette étude dresse un constat assez accablant en montrant que les personnes fréquentant les musées sont **déjà** celles qui disposent d'un capital culturel important. Le rôle éducatif des musées semble donc réduit puisqu'il consiste à apporter du patrimoine culturel à ceux qui le possèdent déjà. Par ailleurs, le capital économique ne semble pas être une variable influençant la visite ou non des musées.

Les études suivantes sur la sociologie des publics se sont alors focalisées principalement sur trois critères de distinction : les classes d'âge de visiteurs, le niveau d'expertise du visiteur et l'influence du groupe de visiteurs par rapport au visiteur unique.

La distinction par classe d'âge

On distingue classiquement trois classes d'âge dans l'étude des publics muséaux : l'enfant, l'adolescent et l'adulte [Gob & Drouguet, 2006]. Les études sur l'enfant au musée portent moins sur les motivations des enfants (la visite leur est généralement imposée) que sur les spécificités de réception du discours. Ces études font ressortir la nécessité de contextualiser les informations vis-à-vis de leurs expériences personnelles, en particulier, lier les artefacts muséaux à leurs expériences familiales ou scolaires. Par ailleurs, quels que soient les objets muséaux, la narration semble le meilleur moyen de faire retenir les informations [Anderson *et al.*, 2002].

Les adolescents forment une catégorie de public dont la spécificité n'a été reconnue qu'assez récemment. Dans les relations entre adolescents et musée, le rôle d'intégrateur social du musée tient une place prépondérante. Le musée doit en effet « aider les jeunes de 12 à 17 ans à développer les habiletés cognitives et sociales nécessaires à une bonne adaptation à la vie en société » [Carnegie Council on Adolescent Development, 1994]. Le visiteur adolescent reste cependant rare dans les musées (en dehors des visites scolaires). L'étude de [Lemerise & Soucy, 1999] sur la relation adolescents/musée soulève trois facteurs expliquant ce manque : le caractère trop didactique de la visite de musée (perçue comme une extension de l'institution scolaire), le manque de liberté dans les visites et dans l'accès aux informations et enfin le manque d'intérêt intrinsèque pour les expositions.

Les visiteurs adultes sont les plus nombreux en musée. Les adultes se distinguent par une compréhension fine des distinctions entre les modalités d'apprentissage (comment apprend-on au musée) et l'apprentissage en lui-même (ce qu'on apprend au musée) [Lefebvre & Lefebvre, 1993]. Les visiteurs adultes sont plus attirés par la visite guidée par un professionnel, jugée très satisfaisante en terme d'apprentissage. En particulier, la visite guidée offre la possibilité, via les interactions avec le guide-conférencier, de contextualiser les œuvres les unes par rapport aux autres mais également par rapport à l'histoire du visiteur lui-même. Les interactions avec le guide, qui questionne et stimule directement le visiteur, sont particulièrement appréciées et jugées performantes pour la réalisation d'un apprentissage.

La distinction par niveau d'expertise

Les classements par niveaux d'expertise distinguent essentiellement deux catégories de visiteurs très différentes : le visiteur occasionnel et le visiteur expert. Le classement d'un visiteur dans l'un ou l'autre des groupes est relatif et dépend de la perception qu'a le visiteur de sa familiarité avec les musées. Il semble que le taux de fréquentation des musées soit un très bon indicateur de l'expertise perçue, les visiteurs se percevant comme novices faisant moins de trois visites par an alors que ceux qui en font plus de trois se considèrent comme familiers avec les arts [Davallon, 1992].

Les visiteurs occasionnels ont une approche centrée sur la visite comme recherche de plaisir esthétique, ils ne se considèrent pas suffisamment armés pour l'étude des textes et pour l'interprétation. Le rôle social du musée (lors de la visite en groupe) est central pour eux et d'ailleurs ils visitent plus volontiers le musée en groupe. Leur « plan » de visite est moins structuré que celui des experts et ils auront tendance à rester moins longtemps dans le musée.

Les visiteurs experts quant à eux ont une approche plus centrée sur la recherche de « surprise » dans le sens d'un défi intellectuel qu'il s'agira de relever. Ils savent plus précisément ce qui va les intéresser, ce qui leur permet d'avoir un parcours plus direct.

Ces visiteurs sont attentifs aux interprétations textuelles proposées par le musée, à l'organisation des scénographies. Par ailleurs la visite est pour eux un moyen d'affirmation de l'appartenance à un groupe social, celui des amateurs éclairés.

Le rôle du groupe de pairs

Les experts peuvent retirer du plaisir à apprendre lors d'une visite individuelle, et visitent effectivement souvent les musées seuls. La situation des visiteurs « novices » est très différente et on observe une corrélation inverse entre le niveau d'expertise que s'attribue le visiteur et la probabilité qu'il vienne accompagné. C'est-à-dire que plus le visiteur se considère comme « novice », plus il a de chance d'être accompagné par d'autres visiteurs [Debenedetti, 2003].

Les pairs dans la visite remplissent principalement trois rôles :

- **Le rôle d'enrichissement mutuel.** Face à une œuvre, la première réaction du visiteur novice est souvent de rester interdit : que penser de cette œuvre? La présence de pairs permet de co-construire une interprétation avec des personnes perçues comme des égaux sociaux et donc de diminuer les phases d'ambiguïté que peuvent ressentir les novices face aux œuvres.
- **Le rôle récréatif.** La visite au musée est plus perçue comme un cours par les visiteurs novices que comme une activité de loisir. Les pairs assurent des moments de détente entre les moments de concentration intellectuelle au contact des œuvres.
- **La réassurance ou soutien affectif.** Ce rôle vient du caractère parfois austère des musées et du sentiment que peuvent avoir les novices de « ne pas être à leur place ». Le fait d'avoir un groupe de pairs à leur côtés aide à « dédramatiser » la visite au musée.

Les distinctions qui nous intéressent dans le cadre de notre étude concernent surtout le visiteur adulte, la variabilité la plus importante étant la perception du visiteur de sa propre expertise. C'est cette variabilité qui va orienter le plan de visite, la réception du discours ainsi que le besoin de pairs autour du visiteur. Le rôle des pairs, de leur avis et de leur soutien est particulièrement important. Un dispositif pour l'apprentissage informel doit donc permettre au visiteur de communiquer avec d'autres visiteurs perçus comme ses pairs.

1.2.1.2 Etude des comportements : comment fréquente-t-on le musée?

Les études systématiques des comportements de visite sont relativement récentes. Contrairement aux études sociologiques, essentiellement basées sur des entretiens et des questionnaires, ces études se basent sur l'observation directe des comportements des visiteurs de musées : temps consacré à l'observation des œuvres, lecture ou non des cartels,

modes de déplacement... Ce type d'études a pour objectif d'obtenir des données quantifiables permettant dans une certaine mesure de prévoir les comportements des visiteurs et d'adapter ainsi la mise en scène muséale.

L'étude de [Veron & Levasseur, 1983] distingue quatre types de visiteurs (figure 1) : poissons, fourmis, sauterelles et papillons. Les fourmis vont souhaiter voir l'ensemble de l'exposition, et feront donc à chaque fois de petits trajets en suivant le parcours de visite. Les papillons se déplacent de manière beaucoup plus aléatoire en faisant parfois des retours en arrière. Les sauterelles effectuent des bonds, qui vont les amener directement aux œuvres qui les intéressent sans voir le reste. Enfin les poissons traversent les salles sans s'arrêter en suivant le cours de l'exposition.

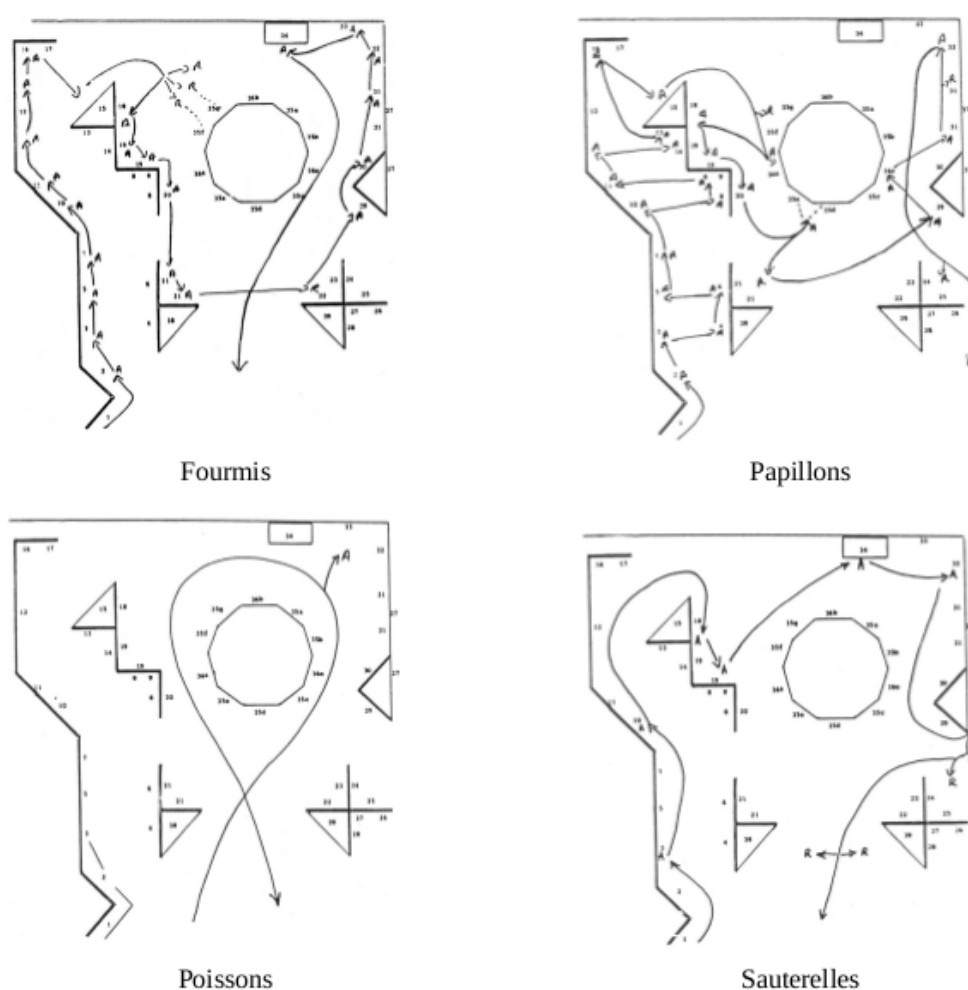


FIGURE 1.2.1: Les quatre types de visiteurs d'après [Veron & Levasseur, 1983]

Des études similaires ont été réalisées par [Sparacino, 2002], qui propose une typologie

des visiteurs en trois catégories : le visiteur « occupé » (busy), « sélectif » (selective) et « avare » (greedy). Le visiteur avare souhaitant l'exhaustivité de la visite, le visiteur occupé faisant une visite rapide de l'ensemble de l'exposition et le visiteur sélectif allant en profondeur uniquement sur certaines questions qui l'intéressent et survolant les autres.

1.2.1.3 Etudes cognitivistes : comment comprend-on le musée?

Les études cognitivistes cherchent à déterminer les variabilités dans la réception du discours muséal, dans l'interprétation des expositions et les critères à prendre en compte pour modéliser ces variabilités. Ces études ont souvent pour objectif la conception de systèmes informatiques d'aide à la visite. En dehors du mode de visite, qui a été évoqué plus haut, deux facteurs de variabilité sont principalement retenus : les intérêts du visiteur et ses connaissances. Les intérêts du visiteur vont permettre d'orienter le discours sur les œuvres, en mettant par exemple en avant des perspectives historique, sociale ou scientifique, l'idée étant alors d'offrir un support à l'interprétation des œuvres par les visiteurs.

Les connaissances du visiteur correspondent à son « background culturel » et aux connaissances qu'il va acquérir durant la visite. La modélisation des connaissances du visiteur permet d'éviter la redondance des informations [Oppermann & Specht, 1999] ou au contraire de la favoriser [Wang *et al.*, 2009] suivant l'approche pédagogique choisie.

1.2.2 Apprendre en musée ?

D'après [Poulot, 2001], le musée peut mobiliser deux ressorts différents : l'émerveillement devant l'œuvre ou sa mise en écho avec d'autres connaissances et d'autres savoirs. Le média informatique est adapté dans le second scénario, en aidant le visiteur à établir les liens conceptuels entre les œuvres exposées suivant différentes modalités (liens historiques, artistiques...). Par ailleurs, apprendre en musée est également apprendre à retirer du plaisir d'une visite. Les médias informatiques, en personnalisant la visite et en proposant des interactions originales, peuvent renforcer ce plaisir de visite.

Une question importante pour qu'on puisse parler d'apprentissage est l'application pertinente des connaissances acquises dans une nouvelle situation. Cette propriété est particulièrement difficile à vérifier dans le cadre des musées, où comme nous l'avons vu, il s'agit plutôt de permettre l'interprétation personnelle, la construction d'un sens de l'exposition que de présenter des connaissances opérationnelles au visiteur. D'ailleurs, les évaluations de visites de musées quantifient le plaisir de visite plutôt que l'apprentissage qui en est retiré.

Le processus par lequel prend forme l'apprentissage en musée peut être représenté par l'approche constructiviste de l'apprentissage humain [Anderson *et al.*, 2003]. Dans ce

cadre, l'apprentissage se produit en deux types d'étapes, les étapes graduelles correspondant à l'acquisition incrémentale de connaissances, les étapes structurelles correspondent à un changement important des conceptions de l'individu. Un point important de l'apprentissage constructiviste est qu'il est propre à chaque individu. Nous retrouvons donc l'importance de contextualiser les informations ou interactions proposées par rapport aux connaissances du visiteur.

1.3 Conclusion : éléments d'un modèle du contexte en musée

Nous avons évoqué l'évolution historique des musées, l'apparition du discours muséal et les formes que peut prendre l'adaptation de ce discours suivant les différents types de publics. Cette étude nous permet de déterminer les éléments qui devront faire partie du modèle de visiteur et le type d'adaptations ou d'interactions à proposer suivant les différentes variabilités. L'objectif d'un tel modèle étant de capturer aussi finement que possible les conditions permettant l'accès à la culture.

Nous retenons deux types de variabilité, celles propres aux visiteurs et celles propres à l'environnement (le musée lui-même, les conditions de visites). Parmi les variabilités propres au visiteur, nous retenons le niveau d'expertise, les connaissances acquises durant la visite ainsi que ses intérêts. Parmi les variabilités propres à l'environnement, nous retiendrons la position du visiteur dans le musée définissant les œuvres autour de lui, ainsi que l'historique de son parcours de visite.

Etat de l'art

Chapitre 2

Représentation des connaissances et métriques sémantiques

Nous détaillons dans ce chapitre des éléments sur la représentation des connaissances et sur les mesures qu'il est possible de définir sur ce type de représentations. Ces éléments nous permettront dans la suite de ce travail de construire nos modèles des œuvres et du contexte ainsi que nos mesures de proximités sémantiques et contextuelles.

L'ingénierie des connaissances a pour objet l'acquisition, la modélisation et le traitement de connaissances dans des environnements informatiques. Ce champ se distingue donc de l'informatique classique, centrée sur le traitement de **données** (structures les plus élémentaires) et d' **informations** (données structurées). Les objets informatiques considérés en ingénierie des connaissances ne sont plus des données/informations mais des **connaissances formalisées**. Pour cela, l'ingénierie des connaissances se base sur la définition de langages formels, permettant d'exprimer de manière computationnelle des savoirs de nature linguistique ou cognitive [Bachimont, 2000]. Ces langages permettent la description de problèmes de haut niveau, par exemple les connaissances de sécurité à mettre en œuvre dans une situation de risque industriel [Edward *et al.*, 2010]. La nature computationnelle des langages formels rend possible la réalisation automatique d'inférences sur les problèmes décrits. L'inférence a pour rôle de déduire automatiquement de nouvelles connaissances à partir de la description initiale, permettant ainsi d'automatiser en partie la résolution des problèmes décrits.

Dans le domaine des EIAH, l'ingénierie des connaissances présente l'intérêt de permettre la représentation formelle d'un domaine d'apprentissage et d'offrir des outils permettant aux apprenants de manipuler les objets de ce domaine pour en acquérir une compréhension appropriée [Lenne, 2009]. La manipulation directe de représentations de connaissances peut cependant s'avérer délicate, la représentation informatique de connaissances donnant lieu à des objets complexes (ontologies, bases de connaissances...). Les proximités et similarités sémantiques permettent d'affranchir en partie l'utilisateur final des

complexités de la manipulation des connaissances formelles en définissant des métriques permettant d'associer automatiquement les connaissances entre elles et de faciliter la navigation entre connaissances. Ainsi, [Aubry, 2007] propose l'utilisation de similarités sémantiques pour la recherche de concepts dans un corpus d'annotations portant sur des représentations 3D de composants mécaniques. L'utilisation de similarités sémantiques permet donc un support fonctionnel aux interactions entre l'utilisateur final et les descriptions formelles de connaissances du domaine.

Nous présentons dans ce chapitre les paradigmes dominants pour la représentation informatique des connaissances. Nous détaillons en particulier les langages RDF, RDF(S) et OWL et leur opérationnalisation dans le cadre du web sémantique. Nous illustrons cette opérationnalisation à travers la description de quelques systèmes centrés sur l'interaction avec des modèles de connaissances, les systèmes à base de connaissances. Nous nous pencherons ensuite sur les différents types de similarités et proximités sémantiques et leur utilisation pour outiller les utilisateurs humains dans la manipulation de connaissances formalisées. L'utilisation de proximités nous permettra dans la suite de ce travail de proposer des interactions adaptées au visiteur d'un musée en fonction des œuvres qui y sont exposées.

2.1 La représentation des connaissances

2.1.1 Représentation des connaissances et langages formels

Définir ce qu'est une connaissance n'est pas chose facile et relève plutôt du domaine de la philosophie ou de l'épistémologie. Nous retiendrons ici la définition de [Bachimont, 2004] "Une connaissance est la capacité d'exercer une action pour atteindre un but." Cette définition met en avant l'idéalité de la connaissance (la capacité n'est pas l'action), et l'importance de la **finalité** de la connaissance, une connaissance visant à la réalisation d'un but. On peut distinguer les connaissances théoriques, qui ont pour finalité l'explicitation dans un code de communication, des connaissances pratiques, qui ont pour finalité la réalisation d'une transformation dans le monde réel.

Plusieurs définitions ont été proposées pour la **représentation informatique** des connaissances. Pour [Winston, 1992], une représentation des connaissances doit permettre de dénoter des objets et de décrire les relations entre eux. Pour [Brachman & Levesque, 2004] il s'agit d'écrire une représentation d'une partie du monde de telle façon qu'une machine puisse parvenir à de nouvelles conclusions sur l'environnement réel en manipulant cette représentation. Suivant [Bachimont, 2000], les connaissances s'appréhendent à travers leurs inscriptions (c'est-à-dire leurs représentations), ces inscriptions étant de nature documentaire (destinées à une interprétation par l'humain) ou formelle (destinée à une interprétation computationnelle). C'est cette deuxième catégorie de représentation qui nous intéresse en particulier.

La représentation des connaissances d'un domaine doit donc refléter ce domaine et permettre la réalisation d'opérations, ces opérations correspondant à la « capacité d'exercer une action » propre aux connaissances humaines. Le rôle fondamental d'une représentation est donc d'être un substitut simplifié du domaine considéré, permettant d'opérer des actions et des raisonnements sur la représentation pour en déduire des conséquences sur le domaine. Pour cela, les représentations de connaissances portent en elles un ensemble d'**engagements ontologiques**, définissant les objets d'intérêts du domaine, leurs relations, les règles qui les lient entre eux. Enfin, une représentation des connaissances définit de manière formelle et computationnelle les mécanismes d'inférences permis sur les objets du domaine [Davis *et al.*, 1993].

Différentes approches ont été proposées pour ces représentations au travers des langages de description des connaissances. Ces langages formels permettent la construction de concepts, de relations entre concepts, d'individus, de règles, ainsi que la description des mécanismes d'inférences permis. On peut distinguer quatre approches principales pour les langages de description des connaissances : les approches issues de la logique du premier ordre, les approches issues de la Frame Logic, les approches issues des logiques de description et les approches issues du Web sémantique (figure 2.1).

Les approches issues de la logique du premier ordre sont basées sur un ensemble de constantes (les « mots » du langage), de variables et la possibilité d'écrire des expressions utilisant la quantification universelle ou existentielle. La description du monde permise par ces langages consiste donc en un ensemble de « phrases » exprimées en logique du premier ordre. Les langages KIF [Ginsberg, 1991] et Cycl [Lenat & Guha, 1991] sont représentatifs de cette famille de langages. Ce type de représentation a été peu à peu abandonné, en effet les langages purement logiques s'avèrent difficiles à manier pour la description des connaissances de haut niveau. De plus la logique du premier ordre peut donner lieu à des bases de connaissances indécidables et donc à construire des systèmes qui ne pourront conclure sur la véracité d'une assertion.

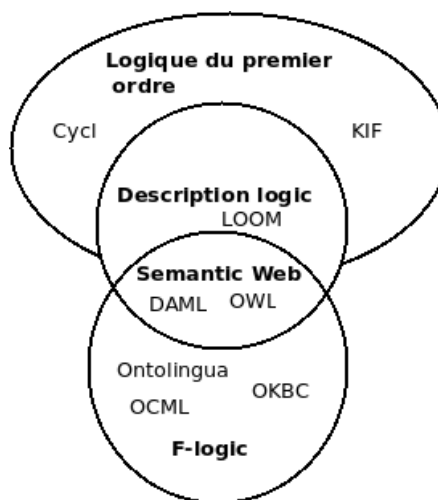


FIGURE 2.1.1: Langages de description des connaissances

Les langages basés sur la F-Logic (frame logic) ont été introduits par [Minsky, 1974]. Ils possèdent de nombreux points communs avec la modélisation orientée objet. Ces langages se basent sur les notions de frames ou classes, sur la définition de liens de généralisation/spécialisation entre classes, ainsi que sur la définition de slots, correspondant

aux attributs des classes. Il est également possible de définir des individus, correspondant en langage objet aux instantiations d'une classe. Les langages basés sur les frames sont donc particulièrement « naturels » à utiliser pour les informaticiens, car ils reprennent l'essentiel des mécanismes de la modélisation objet. Cependant, les langages basés sur la F-Logic sont moins expressifs que les langages basés sur les logiques de descriptions [Baader *et al.*, 2005]. Ainsi, l'utilisation de ces langages rend nécessaire l'écriture de procédures pour implémenter des mécanismes déclaratifs en logique de description [Gruber, 1993].

Les logiques de description sont des sous-ensembles décidables de la logique du premier ordre. Les langages issus des logiques de description offrent une grande expressivité ainsi qu'une garantie de décidabilité. En particulier, ils permettent de définir des concepts en termes de contraintes à satisfaire pour qu'un objet soit une instance d'un concept (description intentionnelle). En reprenant l'exemple fourni par [Baader *et al.*, 2005], nous pouvons ainsi décrire le concept d'homme heureux par la formule suivante :

$$Human \sqcap \neg Female \sqcap \exists married.Doctor \sqcap (\geq 5 hasChild) \sqcap \forall hasChild.Professor$$

Un homme heureux est un humain de sexe masculin marié à un docteur et dont les enfants, au nombre d'au moins 5 sont tous des professeurs. Les logiques de description permettent ainsi d'exprimer de manière concise des concepts complexes. Par ailleurs, les logiques de description introduisent la distinction instance/concept via les notions de T-Box (terminological box), permettant la construction des concepts et de A-Box (assertional box) pour la description des individus. Une limite des logiques de description est liée à cette expressivité importante, l'utilisation de constructeurs et de quantificateurs logiques demandant une certaine expertise. Par ailleurs, la sémantique des logiques de description s'interprète en terme de théorie des modèles, non nécessairement familière aux informaticiens.

Enfin, les langages issus du web sémantique reprennent des éléments des langages issus des logiques de description pour la sémantique et des éléments issus de la F-Logic pour la syntaxe. Ces langages sont donc à la fois relativement simples sur le plan syntaxique, ils permettent d'exprimer les connaissances dans un formalisme proche de la modélisation objet, et sont également particulièrement puissants sur le plan de l'expressivité. Nous décrivons dans la partie suivante les langages de description des connaissances basés sur les recommandations du W3C¹.

1. <http://www.w3c.org>

2.1.2 Ontologies et web sémantique

2.1.2.1 Définitions

La définition la plus rencontrée d'une ontologie est certainement celle de Gruber : *une ontologie est la spécification explicite d'une conceptualisation partagée*. La conceptualisation partagée désigne la représentation des connaissances du domaine (conceptualisation) pour laquelle il existe un accord entre experts de ce domaine (partagée). Une ontologie est une spécification explicite de cette conceptualisation partagée, le terme spécification explicite renvoie au caractère formel de cette représentation, qui est donc écrite dans un langage formel. Une ontologie est donc une manière formelle de décrire un domaine de connaissance. On distingue classiquement trois niveaux d'expressivité, le niveau du méta-modèle, spécification du langage formel (ex. RDF(S) [Guha & Brickley, 2004], OWL [McGuinness & Harmelen, 2004]), le niveau du modèle, description d'une théorie du monde dans le méta-modèle (ex. SKOS², CIDOC-CRM³) et le niveau de la base de connaissances, décrivant des individus appartenant au modèle ontologique (ex. termes SKOS, instances de CIDOC-CRM).

Le web sémantique est une architecture et un ensemble de spécifications permettant la description, l'implémentation et le partage de représentations formelles de connaissances, notamment d'ontologies et de base de connaissances, en utilisant le web et en particulier le protocole http pour la description et l'accès aux ressources. L'idée fondamentale du web sémantique est de proposer une description sémantique de ressources web identifiées par une URI (ex. <http://www.w3.org/2000/01/rdf-schema#Class>). L'URI joue donc le rôle d'identificateur d'une ressource (identifiant) ainsi qu'éventuellement le rôle de chemin d'accès à cette ressource. L'utilisation des URI comme identifiants offre en pratique une grande flexibilité, plusieurs bases de connaissances ou ontologies distinctes pouvant faire des assertions sur les mêmes objets (c'est à dire ressources web) sans accord préalable. Ainsi, un hôtel, identifié par l'URL de sa page web, pourra être décrit en tant que lieu par ses coordonnées géographiques dans une base de connaissance géographique et comme un établissement 3 étoiles par une base de connaissances touristique. Les ressources ne sont pas nécessairement déréférencables, c'est-à-dire qu'un accès à une représentation de la ressource via le protocole http n'est pas garanti, bien qu'il soit considéré comme une bonne pratique qu'elles le soient.

Du point de vue technique, le web sémantique peut être représenté comme un ensemble de couches, c'est le bien connu « semantic web cake » ou « semantic web stack » (figure 2.2). Les niveaux inférieurs définissent l'alphabet (UNICODE) et la syntaxe (XML) utilisés pour les échanges, les niveaux supérieurs introduisent des représentations formelles de plus haut niveau (RDFS et OWL). Nous décrivons dans la partie suivante ces langages basés sur RDF.

2. <http://www.w3.org/2004/02/skos/>

3. <http://www.cidoc-crm.org/>

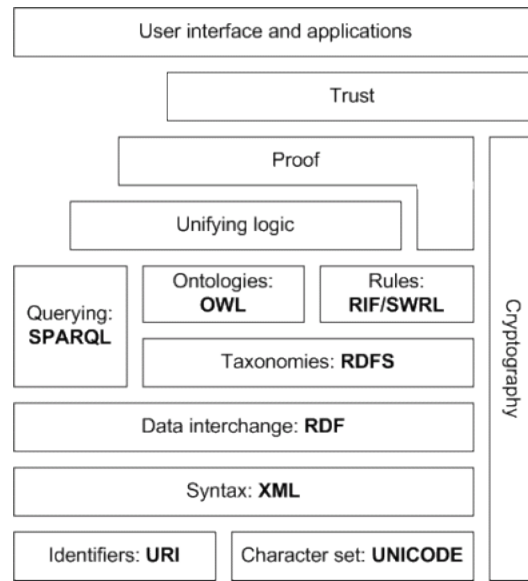


FIGURE 2.1.2: Semantic Web Cake

2.1.2.2 Les langages ontologiques du web sémantique

Notre description se limitera ici aux langages basés sur RDF (Resource Description Framework). Un langage ontologique définit une syntaxe et une sémantique pour la description des ontologies et des bases de connaissances. La syntaxe du langage définit les notations admissibles et la sémantique du langage permet la réalisation d'inférences, c'est-à-dire la dérivation automatique de connaissances nouvelles à partir des assertions d'une ontologie ou d'une base de connaissances.

Le langage RDF est le niveau le plus élémentaire de description des ressources. RDF définit une syntaxe très simple basée sur l'utilisation de triplets. Un triplet est un ensemble ordonné de trois éléments : le sujet, le prédicat (ou propriété) et l'objet (s,p,o). La convention d'écriture de RDF, suivant les recommandations du W3C, est le format XML. Cependant en raison des lourdeurs de cette écriture nous écrirons dans la suite les assertions RDF au format N3⁴ [Berners-Lee, 2006] (figure 3).

RDF permet ainsi de mettre en relation différentes ressources, et d'interpréter ces relations, pour peu qu'on ait convenu auparavant d'une signification (sémantique) pour les prédicats. RDF définit justement un nombre réduit de mots clés, possédant une sémantique particulière et donc permettant une interprétation. Un point intéressant à noter est que RDF permet une interprétation visuelle immédiate sous forme de graphes (figure 3), les noeuds du graphe étant les ressources et les prédicats étant les liens entre ressources.

4. Le format Notation 3 ou N3 est une sérialisation alternative du RDF beaucoup plus compacte et lisible pour les humains

Cette interprétation graphique sera conservée quand nous traiterons d'autres langages ontologiques.

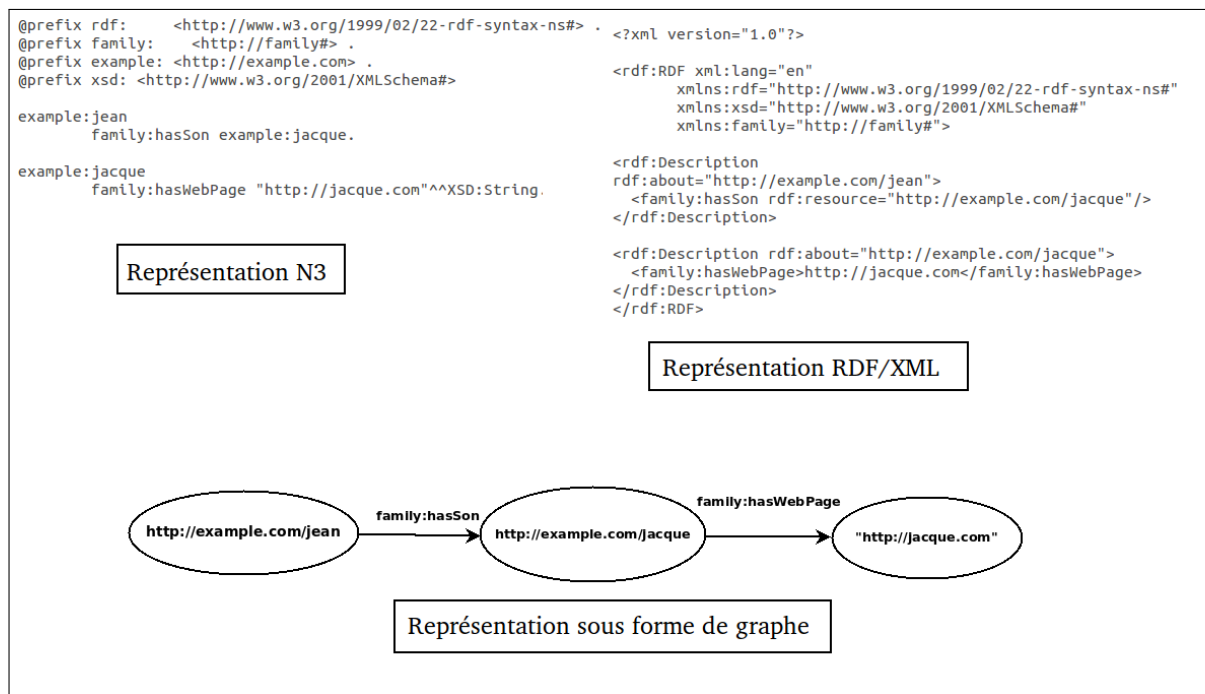


FIGURE 2.1.3: Différentes représentations des mêmes assertions RDF

RDFS (RDF-Schéma) est la première extension de RDF. Tout document RDFS valide est un document RDF valide. RDFS introduit deux notions fondamentales pour la modélisation et la réalisation des inférences : la subsomption avec *rdfs:subClassOf*, et la restriction des valeurs que prennent les propriétés avec *rdfs:range* et *rdfs:domain*. Chacun de ces prédicats possède une interprétation sémantique, correspondant à de nouveaux prédicats à ajouter à la base de connaissances lorsqu'ils sont rencontrés. Le tableau 2.1 suivant illustre les inférences réalisées suivant certains prédicats.

Assertion	Triplet inféré
A <i>rdfs:subClassOf</i> B. a <i>rdf:type</i> A.	a <i>rdf:type</i> B
P <i>rdfs:domain</i> Q. a P b.	a <i>rdf:type</i> Q
P <i>rdfs:range</i> Q. a P b.	b <i>rdf:type</i> Q

TABLE 2.1: Assertions et inférences RDFS

Le langage OWL est basé sur RDFS et utilise d'ailleurs les mots réservés de RDFS. Tout

document OWL valide est un document RDFS valide. Il existe trois versions de OWL, qui sont, par expressivité croissante : OWL-Lite, OWL-DL et OWL-Full. En pratique on utilise essentiellement OWL-DL car l'expressivité de OWL-Lite est faible et OWL-Full est indécidable, dans la suite du texte OWL désignera donc OWL-DL. La différence essentielle entre RDFS et OWL tient au fait que la sémantique de OWL est basée sur une logique de description (la logique SHOIN). Ce langage permet donc une expression beaucoup plus précise des représentations de connaissances, avec en particulier les contraintes de cardinalité sur les propriétés, l'existence du quantificateur existentiel pour les propriétés d'un concept (c'est-à-dire qu'un concept **doit** posséder cette propriété), la définition de propriétés transitives, inverses et fonctionnelles ainsi que la définition intentionnelle des classes (ex. définir la classe A comme la restriction de la classe B pour une valeur donnée d'une propriété). De manière similaire à RDFS, les triplets inférés sont ajoutés à la base de connaissances ou à l'ontologie lorsque certaines assertions sont rencontrées. Le tableau 2.2 suivant illustre certaines de ces inférences.

Assertion	Triplet inféré
P a owl:TransitiveProperty. a P b. b P c.	a P c
Human a owl:Class ; [a owl:Restriction ; owl:onProperty :hasParent ; owl:Cardinality 2] . a rdf:type Human	a hasParent _P1. a hasParent _P2.

TABLE 2.2: Assertions et inférences OWL

2.1.3 Interaction avec des connaissances formalisées : les systèmes à base de connaissances

Les ontologies et les bases de connaissances permettent de représenter avec une granularité arbitrairement fine les connaissances d'un domaine. Cependant, les formalismes de représentation RDFS ou OWL sont plus adaptés à une interprétation par les machines que par les êtres humains. Les interactions directes entre un humain et une base de connaissances sont délicates et impliquent la connaissance de langages de requêtes comme SPARQL.

L'interaction entre humains et connaissances formalisées se fait le plus souvent par l'intermédiaire de systèmes à base de connaissances. Les systèmes à base de connaissance sont une classe de programmes informatiques capables de raisonner sur des abstractions du monde réel pour résoudre des problèmes d'un domaine donné [Stefik, 1995].

La figure 4 illustre les opérations d'abstraction et d'interprétation qui sont la base du fonctionnement des systèmes à base de connaissances.

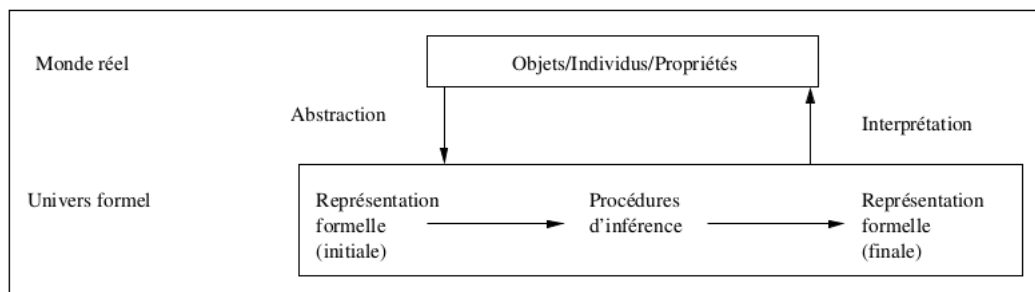


FIGURE 2.1.4: Principe des systèmes à base de connaissance [Le Ber *et al.*, 2006]

Suivant la classification de [O'keefe & Preece, 1996], les systèmes à base de connaissances remplissent cinq types de tâches : la prise de décision cumulative, l'aide à la décision (conseil), l'aide à la recherche de solutions (systèmes heuristiques), l'aide à la gestion de cas et les systèmes critiques (systèmes surveillant les actions de l'utilisateur et lui proposant des alternatives lorsque ses actions ne sont pas optimales). On peut noter qu'avec l'avènement des EIAH et, en particulier, des serious games de nombreux systèmes à base de connaissances sont apparus où les connaissances formalisées sur lesquelles se base le système n'apparaissent pas directement à l'utilisateur. En particulier, dans le cas des environnements virtuels de formation, (ex. V3S [Barot *et al.*, 2011]), les connaissances à faire acquérir à l'apprenant, et sur lesquelles repose le système, ne sont généralement pas présentées directement à l'apprenant.

Un aspect fondamental des systèmes à base de connaissances et qui les différencie d'autres approches de résolution de problèmes (réseaux de neurones, algorithmes génétiques...) est la capacité du système à **justifier précisément ses choix**. En effet, une recommandation ou une décision prise par un système de connaissance pourra toujours être justifiée en présentant l'état initial et l'application des règles qui lui ont permis d'aboutir à une conclusion.

2.1.4 Conclusion sur la représentation des connaissances

Nous avons présenté la représentation informatique des connaissances humaines ainsi que différentes techniques pour ces représentations. Nous avons détaillé les recommandations du W3C : RDF, RDFS et OWL. Ces langages de représentation des connaissances utilisent une syntaxe basée sur la F-Logic et la représentation des ressources par leurs URI, permettant ainsi une grande flexibilité dans les descriptions. Cette syntaxe s'avère

particulièrement utile dans le cadre du web sémantique dans lequel « Anyone can say anything about anything »⁵.

Le choix des formalismes RDFS et OWL dans la suite de notre travail n'est donc pas simplement dû à des questions techniques, mais également au fait que les langages issus du web sémantique sont conçus pour le traitement et la réutilisation des connaissances distribuées. Ainsi, il sera relativement aisé d'intégrer des connaissances publiées au format RDFS ou OWL, notamment celles publiées par différents musées.

2.2 Mesures de similarités et proximités sémantiques

Les similarités et proximités sémantiques sont utilisées dans un large panel d'applications : recherche d'information, alignement d'ontologies, désambiguation, data-mining... Cependant, la manipulation directe de représentations formelles de connaissances pour déterminer les similarités est relativement difficile. En particulier, la détermination de concepts ou d'instances en lien ou proches d'autres ressources nécessite des requêtes SPARQL complexes, comportant plusieurs jointures. Ces requêtes dépendent de l'ontologie considérée et ne sont donc pas généralisables.

Plusieurs méthodes ont été proposées pour faciliter la détermination automatique des proximités et des similarités, tant au niveau des concepts qu'au niveau des instances. Nous définissons tout d'abord formellement les notions de similarités et de proximités sémantiques. Nous proposons ensuite une classification des similarités sémantiques suivant la formalisation utilisée pour décrire les concepts.

2.2.1 Définition

Une fonction de similarité se définit classiquement de la manière suivante :

Soit D l'ensemble des éléments du domaine considéré et (V, \leq) un ensemble totalement ordonné. Une fonction $s : D \times D \rightarrow V$ est une fonction de similarité ssi il existe $0_V \in V$ et $1_V \in V$ tels que :

- $\forall x, y \in D, 1_V \geq s(x, y) \geq 0_V$
- $\forall x \in D, s(x, x) = 1_V$
- La fonction est symétrique : $\forall x, y \in D, s(x, y) = s(y, x)$

En pratique on prendra pour V l'intervalle $[0, 1]$. Il est à noter que la condition de symétrie de la fonction de similarité ne fait pas l'unanimité. En particulier, les travaux de Tversky sur la perception humaine de la similarité tendent à montrer que la notion humaine de similarité n'est pas symétrique. Par exemple, la similarité entre un concept et un concept plus générique est moins importante que la similarité entre un concept et

5. <http://www.w3.org/TR/rdf-concepts/>

une spécialisation de ce concept. Ainsi les jugements humains considèrent généralement « véhicule » plus similaire à « voiture » que « voiture » à « véhicule ».

2.2.1.1 Similarité et proximité

Deux catégories de mesures, respectant la définition de la similarité donnée, permettent de capturer la notion de distance, d'éloignement, entre des ressources d'une ontologie : les mesures de similarités et les mesures de proximités. La similarité entre deux objets quantifie la ressemblance entre ces deux objets. Une mesure de similarité appliquée au domaine des véhicules permettra par exemple de déterminer que les concepts de *camion* et *voitures* sont similaires en ce sens que ce sont tous deux des véhicules, qu'ils ont tous deux des roues, qu'ils sont tous deux utilisés pour des déplacements. La similarité sémantique est donc une similarité dans la structure des concepts, deux concepts similaires pouvant être, dans une certaine limite, échangeables dans une phrase.

Les mesures de proximités quantifient quant à elles les associations entre les concepts. Deux concepts de natures très différentes peuvent être associés de manière très forte. Par exemple le concept de *camion* est sémantiquement proche du concept de *chauffeur de camion*, car ces deux concepts sont fortement liés. Ces concepts sont cependant très différents sur le plan de la similarité sémantique. On ne pourra généralement pas dans une phrase remplacer un concept par un concept proche sans changer fortement le sens de la phrase. Les concepts de proximité et similarité sémantique sont donc proches mais ne sont pas similaires [Aimé *et al.*, 2011]. On peut cependant noter que dans beaucoup de cas des concepts ayant une forte similarité sémantique auront une forte proximité sémantique, mais ce n'est pas une règle générale. De plus, deux concepts ayant une forte proximité sémantique n'ont aucune raison *a priori* d'avoir une forte similarité sémantique.

2.2.2 Mesures de similarités et de proximités

Nous décrivons à présent les mesures classiques de similarités et de proximités classiques. Afin d'organiser cette description, nous décrivons les similarités et les proximités suivant l'espace D dans lequel sont définis les concepts. Nous décrivons les mesures usuelles de similarités sémantiques suivant trois types d'espaces, du plus simple au plus complexe. Le premier espace correspond à une hiérarchie de concepts (les concepts sont liés entre eux par des relations de généralisation ou de spécialisation). Le deuxième espace correspond à une ontologie possédant d'autres types de relations entre concepts (*objectProperty*, *dataTypeProperty*). Enfin, le troisième espace correspond à une ontologie pour laquelle des représentants des concepts sont présents dans un corpus extérieur (de documents textuels par exemple). Pour chaque espace, nous décrivons les principaux modes de calculs de similarités ou de proximités présents dans la littérature.

2.2.2.1 Approches basées sur les hiérarchies de concepts

Les premières approches de similarité sémantique, basées sur des hiérarchies de concepts sont issues des travaux de [Rada *et al.*, 1989]. Le principe de cette approche est de considérer que deux concepts sémantiquement proches sont également proches dans l'ontologie, en terme de parcours d'arcs. La distance de Rada entre deux concepts se calcule donc en mesurant les arcs séparant les deux concepts. Plus précisément, la distance de Rada entre deux concepts C_1 et C_2 est la somme :

$$D_{Rada} = d(C_1, lcs(C_1, C_2)) + d(C_2, lcs(C_1, C_2))$$

Avec $lcs(C_1, C_2)$ le *least common subsumer* de C_1 et C_2 , c'est à dire le concept le plus spécifique subsumant à la fois C_1 et C_2 . Le principe de la distance de [Rada *et al.*, 1989] est donc tout simplement de compter le nombre d'arcs séparants deux concepts. Intuitivement, on peut percevoir que plus le nombre d'arcs est important, plus la distance sémantique entre deux concepts sera importante.

La distance de Rada entre deux concepts pose cependant un problème. En effet, elle ne tient pas compte de la différence de spécificité que peuvent prendre les liens **is-a**. En effet, suivant sa position dans l'ontologie un lien **is-a** porte plus ou moins d'information, typiquement les liens les plus spécifiques (situés en bas de la hiérarchie de concepts) sont plus porteurs d'information que les liens les plus génériques. Par exemple, l'application de la distance de Rada à Wordnet⁶ (une taxonomie comportant l'essentiel des termes de langue anglaise) donne une même similarité pour *plante* et *animal* que pour *zèbre* et *cheval*, alors qu'il est manifeste que zèbre est sémantiquement plus proche de cheval que plante d'animal.

D'autres types d'approches, basées sur une pondération des liens hiérarchique, ont alors été proposées. Les premières approches [Kim & Kim, 1990], [Lee *et al.*, 1993], se sont basées sur une annotation manuelle des arcs de l'ontologie à l'aide de poids, les chercheurs donnant une estimation de la force du liens. Ces approches sont appropriées pour des hiérarchies de petites tailles. Cependant ces approches deviennent très contraignantes lorsque la hiérarchie atteint une taille importante.

[Richardson *et al.*, 1994] ont proposé un calcul automatique appliqué à la hiérarchie de concepts Wordnet et prenant en compte la densité locale de concepts et la profondeur des concepts comparés pour assigner des poids aux liens de l'ontologie. Le principe de cette méthode est que les distances associées aux arcs sont plus faibles dans les parties denses de la hiérarchie de concepts (les parties où les concepts possèdent de nombreux frères) et de la même manière les distances associées aux arcs sont plus faibles lorsque les concepts sont profonds dans l'ontologie. La méthode de calcul proposée par

6. <http://wordnet.princeton.edu/>

[Richardson *et al.*, 1994] prend également en compte le contenu informationnel des concepts en se basant sur les travaux de [Resnik, 1993] que nous évoquerons plus loin. L'idée étant qu'un concept ayant un contenu informel important est un concept « rare », que l'on a peu de chance de rencontrer, et donc que les liens sémantiques sont plus forts entre les concepts ayant des contenus informels moins importants. Ceci est illustré en figure 2.2.1 où le concept *Life_form* est plus fortement lié aux concepts *Animal*, *Person* et *Plant* qu'aux concepts *Aerobe* et *Plankton*, ces derniers étant relativement rares.

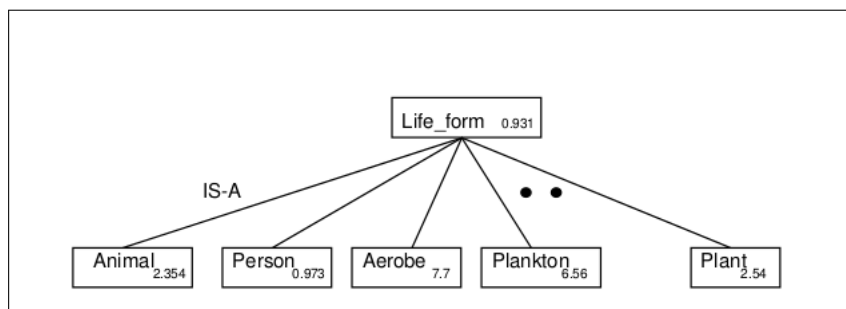


FIGURE 2.2.1: Les contenus informationnels de quelques concepts de Wordnet

On pourra noter que la similarité proposée par [Richardson *et al.*, 1994] n'est pas strictement limitée à une hiérarchie de concepts car elle fait intervenir le contenu informationnel. Nous avons cependant choisi de la faire figurer dans cette section car elle se situe dans la continuité directe des mesures basées sur les calculs d'arcs dans les hiérarchies. On notera cependant que les auteurs ne donnent pas d'expression formelle de la similarité qu'ils proposent, ce qui limite l'applicabilité de la proposition.

Les similarités qui ont suivi sont essentiellement basées sur la forme de la hiérarchie de concepts, en intégrant la notion de profondeur du *least common subsumer* dans les formules. La similarité de [Wu & Palmer, 1994] est définie de la manière suivante :

$$sim_{Wu-Palmer}(C_1, C_2) = \frac{2 \cdot depth(lcs(C_1, C_2))}{depth(C_1) + depth(C_2)}$$

Avec $depth(C_i)$ la profondeur du concept C dans la hiérarchie. Cette formule se comprend assez facilement. Plus les concepts ont un subsumant commun profond dans la hiérarchie, plus la similarité sera importante. De plus, plus les concepts sont proches de ce subsumant commun, donc plus $depth(C_1) + depth(C_2)$ est faible, plus les concepts sont proches.

La mesure proposée par [Leacock *et al.*, 1998] prend en compte la profondeur totale de la hiérarchie de concepts et l'éloignement, en nombre de noeuds, entre deux concepts (ex. deux concepts frères ont un éloignement en nombre de noeuds de trois). Cette mesure est définie par :

$$Sim_{Leacock}(C_1, C_2) = -\log \left(\frac{dist_{node}(C_1, C_2)}{2 \cdot D} \right)$$

Avec $dist_{node}(C_1, C_2)$ la distance en nombre de noeuds entre C_1 et C_2 et D la profondeur maximale de la taxonomie. La mesure est toujours positive, et l'utilisation de la fonction \log sur des réel inférieur à 1 permet une bonne dispersion des résultats.

Nous avons présentés les principales méthodes de calculs de similarités basées sur des hiérarchies de concepts. Ces mesures seront utiles dans la suite de notre travail pour comparer certaines données muséologiques exprimées sous forme de hiérarchies (ex. hiérarchie de styles, hiérarchie de thèmes). Ces mesures ne permettent cependant pas de capturer tous les éléments des similarités sémantiques. Nous poursuivons donc notre description des similarités sémantiques avec les approches basées sur les propriétés.

2.2.2.2 Approches basées sur les propriétés

Dans les approches basées sur les propriétés, l'espace D dans lequel sont comparés les concepts est plus complexe que dans les approches basées sur les hiérarchies. Outre les relations de généralité/spécificité entre concepts (relation is-a), des propriétés sont définies qui associent les concepts entre eux (objectProperty) ou associent des concepts à des classes de littéraux (datatypeProperty).

Les approches basées sur les propriétés permettent le calcul de similarités ainsi que le calcul de proximités. Le principe général de ces mesures est que deux concepts partageant de nombreuses propriétés sont similaires ; par exemple, dans une ontologie des transports, le concept de voiture et le concept de camionnette sont similaires, car ces concepts partagent les propriétés d'avoir quatre roues, un conducteur unique, une valeur de plaque d'immatriculation... Les concepts liés entre eux par des relations sont quant à eux proches ; par exemple il existe une relation fonctionnelle entre le concept de voiture et le concept de conducteur, ces concepts sont donc proches.

Le modèle de similarité de [Tversky, 1977] est typique de l'approche basée sur les propriétés. La formule générale du modèle de similarité de Tversky est :

$$sim_{Tversky}(A, B) = \alpha \cdot comm(A, B) - \beta \cdot diff(A, B) - \gamma \cdot diff(B, A)$$

avec $comm(A, B)$ le cardinal de l'intersection des propriétés de A et B (propriétés communes) et $diff(A, B)$ le cardinal de l'ensemble des propriétés de A privé des propriétés de B (propriétés uniques à A) (figure 2.2.2).

Il est à noter que le modèle de similarité de Tversky ne respecte pas la contrainte de symétrie. Ce modèle a pour objectif de représenter la perception humaine de la similarité et, suivant les travaux de Tversky cette perception n'est pas symétrique. En particulier,

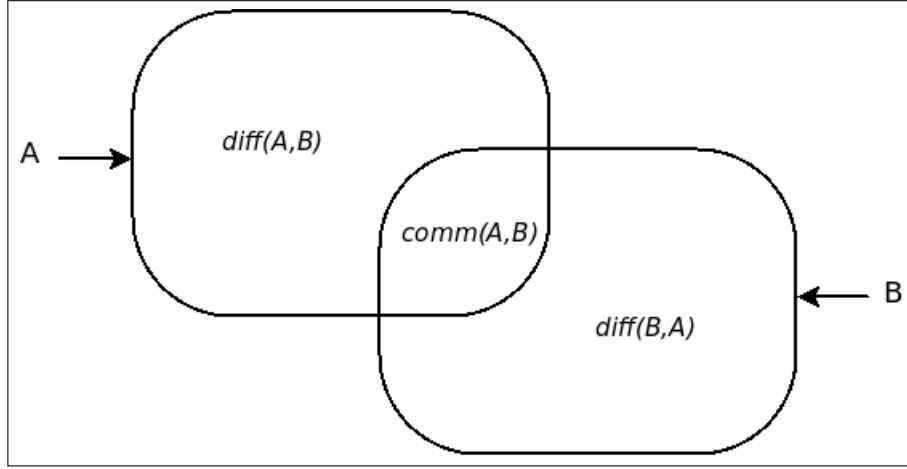


FIGURE 2.2.2: Représentation des similarité dans le modèle de Tversky

comme nous l'avons précisé précédemment, la similarité entre un concept et un concept plus générique $sim(A, B)$, $B \text{ subClassOf } A$ est jugée plus importante que la similarité entre le concept plus générique et le concept $sim(B, A)$, $B \text{ subClassOf } A$.

[Rodriguez & Egenhofer, 2003] proposent une interprétation du modèle de similarité de Tversky pour la mesure de similarité entre concepts de différentes ontologies. Les concepts sont comparés suivant différents aspects, en particulier suivant les *features* (propriétés) communes ou distinctes de ces concepts. Cette mesure se base sur une distinction structurelle entre catégories de propriétés : propriétés de composition, propriétés fonctionnelles et attributs. Les propriétés de type composition correspondent aux méronymies, par exemple le concept de bateau est en relation de composition avec le concept de voile. Les propriétés fonctionnelles relient les concepts à des fonctions (typiquement représentées par des verbes), par exemple le concept de *Bateau* est en relation fonctionnelle avec le concept *Naviguer*. Enfin, les attributs correspondent aux *datatype-Properties*. La formule générale de la similarité par propriété est alors :

$$S_{Features}(a, b) = \omega_p \cdot S_p(a, b) + \omega_f \cdot S_f(a, b) + \omega_a \cdot S_a(a, b)$$

Avec S_p , S_f et S_a les fonctions de similarité respectivement pour les propriétés de composition, fonctionnelles et d'attributs, et ω_p , ω_f et ω_a leurs poids respectifs. Pour chacune des similarités $S_i(a, b)$ ils proposent l'utilisation d'une forme normalisée de la distance de Tversky :

$$S_{Tversky}(a, b) = \frac{|A \cap B|}{|A \cap B| + \alpha(a, b) |A/B| + (1 - \alpha(a, b)) |B/A|}$$

Avec A et B l'ensemble des propriétés des concepts a et b . L'intérêt de cette formulation

de la similarité est qu'elle peut facilement être utilisée dans différents types d'ontologies (ontologie ne présentant que des relations de type part-of par exemple). Il suffira de considérer les similarités d'attributs et fonctionnelles comme nulles. Par ailleurs, l'utilisation de la distance de Tversky offre une certaine garantie que les similarités déterminées par ces formules seront cohérentes avec les jugements humains de similarités. Deux problèmes se posent cependant avec cette formule. Tout d'abord, comment distinguer de manière automatique les relations de type **part-of** et les relations fonctionnelles ? Ensuite comment attribuer les poids ω_i ?

L'équation de similarité $S_{features}(a, b)$ peut s'interpréter comme l'agrégation d'un vecteur de similarités suivant trois dimensions : une similarité suivant l'axe des propriétés, une similarité suivant l'axe des relations et une similarité suivant l'axe fonctionnel. La fonction d'agrégation étant ici une moyenne pondérée. Ce type de formulation, comme agrégation d'un vecteur de similarité a été proposé par [Bergmann & Stahl, 1998] pour le calcul de similarités sur des objets informatiques. Le principe de la similarité de [Bergmann & Stahl, 1998] est de considérer le *least common subsumer* des deux classes et de comparer ces classes suivant les propriétés de ce *least common subsumer*. En effet le *least common subsumer* ne comporte que des propriétés communes aux deux classes. La formulation générale proposée par les auteurs est la suivante :

$$sim(q, c) = \Phi(sim_{A_1}(q.A_1, c.A_1), sim_{A_2}(q.A_2, c.A_2) \cdots sim_{A_N}(q.A_N, c.A_N))$$

Avec q et c deux classes, les $\{A_1 \dots A_N\}$ les attributs du *least common subsumer* de q et c et les $\{Sim_{A_1} \dots Sim_{A_N}\}$ des fonctions de similarités propres à ces attributs, $\Phi()$ étant une fonction d'agrégation.

Si [Bergmann & Stahl, 1998] ne donnent pas de critères généraux pour le calcul des sim_{A_i} , [Aimé, 2011] propose une approche comparable basée sur la définition de vecteurs prototypes. La formule générale du calcul proposé par Aimé est :

$$sim_{Aimé}(C_1, C_2) = 1 - dist(\vec{p}_{C_1}, \vec{p}_{C_2})$$

\vec{p}_{C_i} étant le vecteur prototype du concept C_i c'est à dire le vecteur ayant pour valeur de composante k l'importance de la propriété k pour ce concept.

Il est à noter que les calculs basés sur les propriétés permettent également de déterminer la **proximité** entre concepts. [Aimé, 2011] propose ainsi une mesure permettant de quantifier la proximité entre C_1 et C_2 en considérant le nombre de liens existant dans l'ontologie entre les concepts C_1 et C_2 . Le calcul de proximités correspond alors à la fraction de concepts liants C_1 à C_2 et C_2 à C_1 sur le nombre total de relations impliquant C_1 et C_2 , ainsi deux concepts ayant de nombreuses autres relations auront une faible proximité alors que deux concepts ayant uniquement des relations l'un vers l'autre auront une forte proximité.

Une approche plus générale de la proximité par propriétés entre ressources (une ressource étant un concept ou une instance) est proposée par [Anyanwu *et al.*, 2005]. Les auteurs introduisent la notion de chaîne de propriétés entre deux ressources. Cette notion est assez naturelle et correspond à la suite de propriétés nécessaires pour passer d'une ressource A à une ressource B. Ils proposent ensuite d'associer des poids aux propriétés suivant la probabilité d'occurrence de ces dernières. Par exemple un individu de type *human* possède une chance relativement forte d'avoir la propriété *hasCar*, cependant il possède de faibles chances d'avoir la propriété *hasPlane*. Le calcul d'Anyanwu et al permet alors de déterminer les individus proches en terme de relations « probables » et les individus proches en terme de relations « improbables ».

Nous pouvons constater que l'utilisation de propriétés dans le calcul des similarités apporte une grande richesse par rapport à l'utilisation unique de la hiérarchie de concepts, au prix de l'introduction de plusieurs paramètres à fixer, comme la pondération des propriétés. Ces mesures sont par ailleurs plus adaptées aux cadre actuel du web sémantique, dans lequel les objets possèdent quasiment toujours différents types de propriétés.

2.2.2.3 Approches basées sur des corpus

Dans le cadre des approches basées sur les corpus, les concepts d'une ontologie ont des représentants dans différents documents. Le principe de ces approches est alors de déduire la similarité entre deux concepts des co-occurrences ou des liens entre leurs représentants. L'exemple le plus simple du représentant d'un concept est un mot désignant ce concept, ce mot étant présent dans des documents textuels. Ainsi, si dans un corpus, deux termes désignant deux concepts sont fréquemment cités ensemble, les deux concepts sont considérés comme proches.

Le cadre conceptuel le plus utilisé dans la formulation de similarités basées sur des corpus est celui du contenu informationnel (CI). Le contenu informationnel est une mesure issue de la théorie de l'information qui permet de mesurer la quantité d'information portée par un concept. Le contenu informationnel de l'événement A est défini comme le négatif du logarithme de la probabilité de A : $IC(A) = -\log(P(A))$.

Les premières approches de l'utilisation du contenu informationnel [Resnik, 1995] étaient basées sur la hiérarchie Wordnet. [Resnik, 1995] évaluait le contenu informationnel d'un concept C comme la probabilité de l'événement « Le mot ω est un représentant de C, $\omega \in Corpus$ » avec Corpus un corpus de texte. Cette probabilité étant simplement estimée par $TF(\omega, Corpus)$, c'est à dire la fréquence (Term Frequency) du terme dans le corpus. La similarité de deux termes est alors évaluée par le contenu informationnel de leur *least common subsumer*. Ainsi, les termes possédant un *least common subsumer* doté d'un contenu informationnel important (c'est à dire peu présent dans le corpus) auront une similarité importante.

[Lin, 1998] a proposé une approche dans la lignée de celle proposée par Resnik. L'approche proposée par Lin est très générique et ne se limite pas aux similarités sémantiques. La seule hypothèse de Lin pour le calcul de proximité dans un espace donné est qu'il est possible d'avoir une mesure de probabilité sur les objets de cet espace. Dans le cas d'une mesure conceptuelle, la mesure de Lin prend la forme suivante :

$$Sim_{Lin}(C_1, C_2) = \frac{2 \cdot \log(P(C_0))}{\log(P(C_1)) + \log(P(C_2))}$$

Avec C_o le *least common subsumer* de C_1 et C_2 et $P()$ une mesure de probabilité sur la hiérarchie de concepts.

Ce type de calcul, basé sur la fréquence des concepts dans un corpus a été repris par [Aimé *et al.*, 2011] dans le cadre de la mesure de similarité extensionnelle entre concepts. L'extension d'un concept, pour une base de connaissance donnée, est définie comme l'ensemble de ses instances. La similarité extensionnelle de C_1 , C_2 est fonction du nombre d'instances communes aux deux concepts sur le nombre total d'instances des deux concepts.

Les approches basées sur les corpus sont intéressantes car elles permettent de faire intervenir d'autres sources de connaissances (les corpus) dans le calcul de la similarité. De nombreux corpus déjà existants peuvent ainsi être facilement réutilisés. Une limite de ces méthodes est qu'il sera plus difficile de justifier de la similarité de deux concepts en utilisant une similarité par corpus. En effet, dans le cas des similarité sur des hiérarchies ou sur des propriétés, il est possible de justifier les similarités en considérant par exemple les propriétés communes. Les méthodes par corpus, plus calculatoires, rendent plus difficile une justification directe.

2.2.3 Conclusion sur les mesures de similarités et de proximités

Nous avons décrit différentes similarités sémantiques suivant l'espace de définition des concepts. On peut constater que ces mesures sont plutôt complémentaires que concurrentes. Les mesures de similarité hiérarchique permettent de comparer la position de deux concepts dans une ontologie, les mesures de similarités par propriétés permettant de comparer la structure de ces concepts, enfin, les mesures par utilisation de corpus permettant d'intégrer des connaissances extérieures aux mesures. Plusieurs tentatives ont été effectuées pour déterminer quelles étaient les meilleures mesures par rapport au jugement humain (voir par exemple [Warin *et al.*, 2005] qui considère la mesure de Leacock comme la plus proche du jugement humain). Cependant ces tentatives se limitent souvent aux mesures basées sur les hiérarchies de concepts, les autres mesures comportant trop de paramètres pour pouvoir être comparées simplement.

Dans le cadre de notre travail, l'utilisation de similarités a pour objectif d'assister la navigation au sein d'un corpus d'œuvres et de présenter de nouvelles connaissances

à l'utilisateur. Il est alors important de pouvoir « justifier » des similarités calculées par le système, c'est-à-dire d'expliquer pourquoi l'objet recommandé est similaire ou proche de l'objet considéré par l'utilisateur. Par exemple justifier de la similarité de deux peintures par la similarité de leurs styles. Ce type de justification, explicite, se retrouve essentiellement dans les calculs basés sur la hiérarchie de concepts ou les calculs basés sur les propriétés. Nous utiliserons donc dans la suite ce type de similarités pour déterminer les justifications adéquates lors des suggestions.

Par ailleurs, nous souhaitons prendre en compte le contexte de visite pour déterminer les éléments les plus appropriés à recommander. Dans ce cadre, notre représentation du contexte jouera le rôle d'un corpus. En effet, les différentes dimensions contextuelles (localisation, contexte historique) comporteront des instances, représentations des œuvres que l'utilisateur a autour de lui ou qu'il a déjà vues. Nos calculs de proximités contextuelles consisteront donc à déterminer dans quelle mesure une œuvre est présente dans le contexte utilisateur. Cette approche, typique de l'approche basée sur les corpus, nous permettra de recommander des œuvres présentes par exemple dans le contexte de localisation de l'utilisateur, c'est-à-dire autour de lui.

Chapitre 3

Représentation informatique du contexte

La notion de contexte en informatique, en tant qu'environnement réel extérieur au système informatique et pouvant cependant communiquer avec lui, est intimement liée au concept d'informatique ubiquitaire. Le concept d'informatique ubiquitaire a été introduit par [Weiser, 1991], comme étant le paradigme où les dispositifs informatiques possèdent la capacité de déterminer certains éléments de leur contexte d'utilisation (ex. position de l'utilisateur, nombre de personnes dans une salle...) et peuvent adapter leur mode de fonctionnement et les interactions proposées à ce contexte. Un des exemples présentés par Weiser concerne des badges qui, portés par les employés d'une entreprise, permettent de diriger les appels téléphoniques concernant un employé dans la salle où il se situe. Si, avec l'arrivée des téléphones mobiles, cet exemple n'est plus d'actualité, l'idée d'une technologie calme (*calm technology*), s'adaptant de manière automatique et transparente au contexte d'utilisation (en particulier dans le cadre de dispositifs mobiles), est au cœur de la recherche en informatique ubiquitaire.

Dans ce chapitre, nous reviendrons tout d'abord sur les définitions du contexte et leur caractère opérationnalisable. Nous décrirons ensuite les différents paradigmes de représentation et de traitement des informations contextuelles et l'apport de la sémantique pour ces représentations. Enfin nous décrirons les différents types de systèmes sensibles au contexte en terme d'architecture et d'interaction avec l'utilisateur. Nous porterons une attention particulière aux systèmes pour l'apprentissage humain en positionnant notre contribution parmi ces systèmes.

3.1 Définitions du contexte

Le contexte d'une situation ou d'une activité est une notion vaste pour laquelle il est particulièrement difficile de donner une définition générale et opérationnelle. Suivant [Abowd *et al.*, 1999], deux catégories de définitions se retrouvent principalement : les définitions extensionnelles qui tentent de lister l'ensemble des caractéristiques contextuelles pertinentes pour une application et les définitions tautologiques qui définissent le contexte par des synonymes. Le problème qui se pose est que les définitions extensionnelles ne sont pas généralisables (elles décrivent le contexte pour une application donnée) et les définitions tautologiques ne sont pas opérationnalisables (elles décrivent le contexte en termes de trop haut niveau). [Abowd *et al.*, 1999] proposent alors la définition suivante qui est l'une des plus largement acceptées :

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Cette définition a été critiquée par [Zimmermann *et al.*, 2007] comme étant trop générale et non opérationnelle. En particulier, les termes « *any information* » et « *characterize the situation* », sans plus de précision, ne permettent pas de construire une représentation informatique du contexte. [Zimmermann *et al.*, 2007] proposent une définition plus spécifique du contexte, basée sur des catégories d'informations contextuelles permettant la caractérisation des entités :

« *Context is any information that can be used to characterize the situation of an entity. Elements for the description of this context information fall into five categories: individuality, activity, location, time, and relations. The activity predominantly determines the relevancy of context elements in specific situations, and the location and time primarily drive the creation of relations between entities and enable the exchange of context information among entities* »

La figure 3.1.1 représente une entité ainsi que ses différentes catégories contextuelles suivant l'approche de Zimmerman.

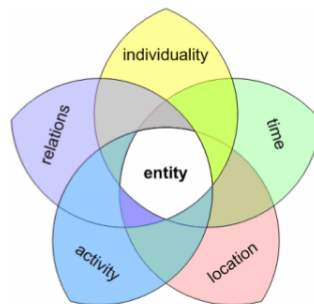


FIGURE 3.1.1: Une entité et son contexte d'après [Zimmermann *et al.*, 2007]

Cette définition du contexte est plus précise sur le plan formel, avec les définitions de cinq catégories contextuelles, et plus spécifique sur le plan opérationnel avec la description des modes de création et d'échange d'informations contextuelles. [Zimmermann *et al.*, 2007] définissent par ailleurs une série d'opérations contextuelles « canoniques », sans faire d'hypothèses sur la situation modélisée.

La figure 3.1.2 représente une de ces opérations contextuelles, l'opération « création d'une relation » entre deux entités. Deux entités *A* et *B* existent indépendamment l'une de l'autre (à gauche de la figure), puis un changement dans l'environnement crée un recouvrement au niveau des contextes spatiaux et temporels (au centre de la figure). Ce recouvrement peut correspondre par exemple à deux personnes regardant un tableau au même moment, ou à deux personnes consultant la même page Web au même moment (dans ce cas le contexte de localisation correspond à la page consultée). À l'issue de ce recouvrement une nouvelle relation contextuelle est créée entre les entités *A* et *B*, cette relation pouvant subsister même si les contextes spatiaux et temporels des deux entités ne sont plus en correspondance (partie droite de la figure).

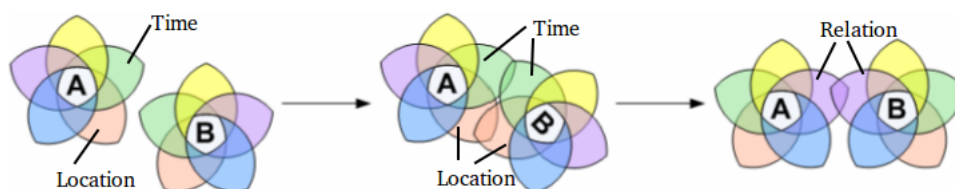


FIGURE 3.1.2: Création d'une relation entre deux entités

Dans la suite de ce travail, nous prendrons comme définition du contexte une version affaiblie de la définition proposée par [Zimmermann *et al.*, 2007], en limitant les entités possédant un contexte aux êtres humains (une peinture par exemple n'aura pas de contexte). Par ailleurs, nous proposerons de considérer les catégories contextuelles de manière plus générale, comme des espaces structurés par des ontologies (chapitre 5). Cette nouvelle caractérisation du contexte nous amènera à proposer des catégories contextuelles différentes de celles définies par [Zimmermann *et al.*, 2007].

3.2 Représentation et traitement des informations contextuelles

Suivant l'étude de synthèse menée par [Baldauf *et al.*, 2007], le processus global mis en place dans les systèmes sensibles au contexte, indépendamment de l'architecture sous-jacente, comporte essentiellement quatre étapes.

Ces étapes, présentées figure 3.2.1, sont : l'acquisition des données brutes issues de capteurs physiques ou logiques, la transformation des données pour alimenter un modèle du contexte, le traitement du modèle du contexte et les transformations ou ajustement applicatif suivant ce modèle. Les deux premières étapes, si elles sont indispensables sur le plan technique, sont très dépendantes de l'application contextuelle. Nous décrirons donc dans la suite les moyens de mettre en oeuvre les deux dernières étapes, c'est à dire les stratégies de modélisation du contexte et l'opérationnalisation des modèles du contexte.

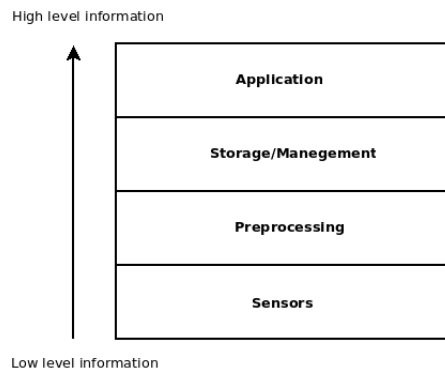


FIGURE 3.2.1: Acquisition et traitement du contexte d'après [Baldauf *et al.*, 2007]

3.2.1 Stratégies de modélisation du contexte

Suivant [Strang & Linnhoff-Popien, 2004], on peut distinguer six stratégies de modélisation du contexte. Nous présentons ces stratégies par expressivité croissante.

Le type de modèle du contexte le plus simple est le **modèle clé-valeur**. Dans ce modèle les entités représentées ont un nombre défini de propriétés (clés) pouvant prendre certains types de valeurs. [Schilit *et al.*, 1994] proposent cette approche dans le cadre d'une application contextuelle où les entités contextuelles (utilisateurs et machines) sont décrites par leur localisation. L'intérêt de cette modélisation est sa simplicité : il est en effet très simple de calculer des appariements (*matching*) entre les utilisateurs et machines suivant la clé localisation. Ce modèle est cependant très limité en terme d'expressivité et l'évolution du modèle pour intégrer de nouveaux types de données contextuelles se révèle complexe.

Les **modèles du contexte à balises** (*markup scheme model*) se basent sur une structure hiérarchique comportant des tags de balisages associés à des valeurs d'attributs. Ces modèles du contexte sont souvent des extensions du modèle *Composite Capabilities/Preference Profiles*¹ (CC/PP) proposé par le W3C. Le modèle CC/PP a été proposé par

1. <http://www.w3.org/TR/CCPP-struct-vocab/>

le W3C pour répondre au problème de l'accès à des pages ou des services Web par des dispositifs de plus en plus variés (ex. smartphone, tablette, télévision...). Un profil CC/PP définit alors de manière simple les capacités et préférences d'un client en fonction d'un certain nombre d'attributs, comme la taille d'affichage préférée, les caractéristiques du navigateur, etc... L'intérêt des modèles à balises, dérivés ou non de CC/PP, par rapport aux modèles clés/valeurs est de pouvoir définir clairement des propriétés ou sous-propriétés contextuelles dans un schéma XML réutilisable. Cependant, ces modèles possèdent des limitations intrinsèques ne leur permettant pas d'être suffisamment expressifs : en particulier l'expression de contraintes sur les valeurs prises par les propriétés ainsi que les liens entre propriétés sont particulièrement difficiles à exprimer dans les modèles issus de CC/PP [Indulska *et al.*, 2003].

Les **modèles graphiques du contexte** sont basés sur des langages formels graphiques et ont l'avantage de permettre une visualisation immédiate du modèle contextuel. Ces modèles sont particulièrement bien adaptés pour un aspect important du contexte : la temporalité. Ainsi [Bauer, 2003] propose l'utilisation des diagrammes de séquences d'UML pour représenter les interactions et rôles de différentes entités lors du contrôle d'une perturbation aérienne. Le langage CML (Context Modeling Language, figure 3.2.2) [Henricksen *et al.*, 2002] est une extension du langage graphique Object Role Modelling (ORM) permettant une représentation riche des relations et rôles des différentes entités (grâce à ORM) tout en ajoutant des propriétés spécifiques aux informations contextuelles. Ainsi, il est possible dans CML de spécifier le type des sources d'informations contextuelles (statiques, issues d'un capteur, dérivées ou fournies par l'utilisateur), de capturer la qualité des informations fournies et gérer les cas d'informations contextuelles contradictoires ainsi que de capturer l'évolution du contexte à travers le temps (historisation). Cependant, ce type d'approche est moins utilisé à présent, les représentations sémantiques offrant une alternative permettant des raisonnements généraux et automatiques plus complexes sur les données contextuelles.

Les **modèles orientés objets** utilisent les principes de la POO (programmation orientée objet) pour la modélisation des données contextuelles. L'intérêt de ces modèles découle directement de l'approche orientée objet : encapsulation des données et des fonctions et réutilisabilité du code (héritage). Les traitements complexes effectués sur les données contextuelles sont cachés par l'encapsulation et sont accessibles *via* des interfaces spécifiques. Ce type de modélisation permet donc de créer des objets contextuels complexes tout en cachant la complexité des traitements aux autres composants du système. Le problème de l'approche objet est son manque de généralité ; en effet même en produisant des classes très génériques, la POO ne possède pas une sémantique basée sur une logique formelle. L'ajout de nouvelles catégories contextuelles demande donc de profondes modifications applicatives.

Nous regroupons les deux dernières stratégies (approches logique et approches ontologique) proposées par [Strang & Linnhoff-Popien, 2004] sous le nom d'**approches sémantiques**. L'idée étant d'utiliser un langage ontologique pour la description du contexte. Ce type

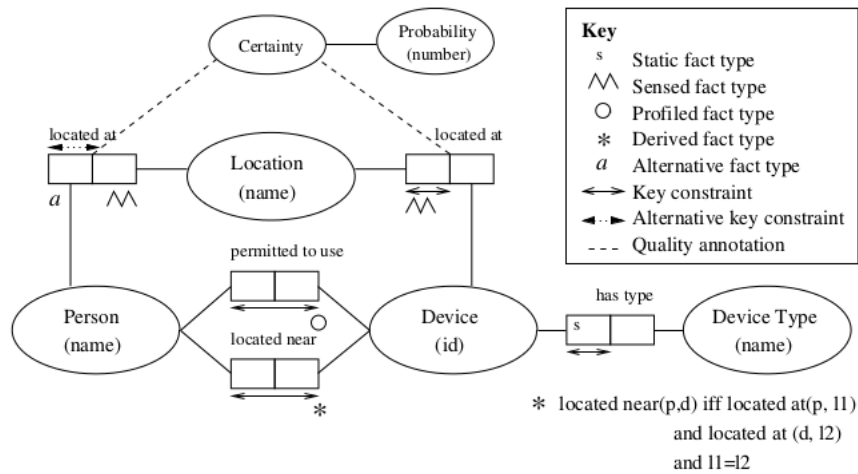


FIGURE 3.2.2: Modélisation du contexte par ORM/CML [Henricksen & Indulska, 2004]

d'approche s'est révélée particulièrement fertile ces dernières années avec de nombreuses propositions d'ontologies tant spécifiques à un domaine que génériques pour la modélisation du contexte [Baumgartner & Retschitzegger, 2006]. Les intérêts de la stratégie sémantique de modélisation sont en effet nombreux. Tout d'abord, en termes techniques, il existe des langages ontologiques de références (OWL, RDFS) possédant une sémantique non ambiguë et centrés sur la mise à disposition d'information distribuées depuis plusieurs sources (ce qui est typiquement le cas des informations contextuelles, agrégées depuis des capteurs logiques ou physiques). Par ailleurs, ces langages ontologiques disposent d'une sémantique formelle clairement définie, assurant ainsi que les inférences sur des contextes identiques seront identiques, ce qui favorise également le partage des informations contextuelles. Dans la suite de notre travail nous utiliserons pour ces raisons un modèle sémantique du contexte.

Le tableau 3.1 présente de manière synthétique les différentes stratégies de modélisation que nous avons décrites.

Modélisation	Expressivité
Attribut-Valeur	Faible
Modèle à balise	Propriété/Sous-propriété
Modèle graphique	Liens entre propriétés, Temporalité, Sources de données
Modèle orientés objets	Liens entre données, Temporalité, Sources des données, Traitements complexes
Modèles sémantiques	Liens- Spécifique/Générique, Temporalité, Source des données, Inférences

TABLE 3.1: Les stratégies de modélisation et leur expressivité

3.2.2 Opérationnalisation des modèles contextuels

Le choix d'une stratégie de modélisation impose déjà un traitement des données, correspondant à l'étape de récupération et de traitement initial des données pour l'alimentation du modèle du contexte (première et deuxième étapes de la figure 3.2.1). Cependant les opérations réellement intéressantes du point de vue fonctionnel sont les opérations de plus haut niveau effectuées sur le modèle du contexte. Ce sont ces opérations qui permettront de lier la logique de l'application au modèle contextuel.

On peut distinguer deux grandes catégories d'opérations sur les modèles du contexte permettant leur opérationnalisation [Zimmermann *et al.*, 2007] (figure 3.2.3) : les **opérations de transition/évolution** du contexte d'une entité (opération d'un contexte dans lui-même) et les **opérations de requête et de partage d'informations contextuelles** (opération impliquant un ou plusieurs contextes et une entité extérieure au modèle du contexte). Les opérations de transition/transformation impliquent typiquement la mise en place de règles ou d'algorithmes numériques permettant le passage d'un contexte à un autre ou la déduction d'un contexte de haut niveau (ex. activité de l'utilisateur) à partir d'un contexte de bas niveau (ex. position de l'utilisateur). Ces opérations correspondent donc à des raisonnements sur le contexte. Les opérations de partage d'informations contextuelles impliquent typiquement des requêtes d'informations contextuelles sur des entités, des comparaisons entre ces requêtes et des modifications applicatives (ex. proposition d'interaction). Nous présentons dans la suite les principales méthodes pour ces opérations et donc pour l'opérationnalisation du contexte.

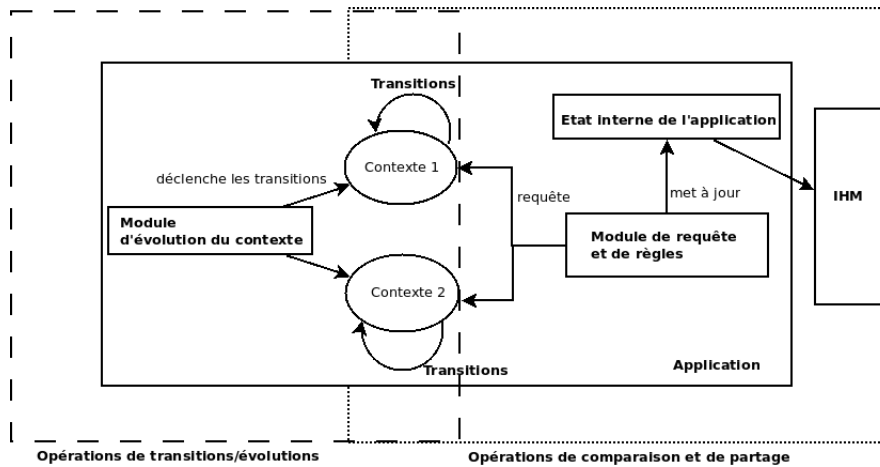


FIGURE 3.2.3: Transitions et requêtes d'information contextuelles

Transitions et évolution du contexte

On peut distinguer deux classes de méthodes permettant l'évolution d'un contexte ou la déduction d'éléments de haut niveau à partir d'éléments de plus bas niveau : les méthodes numériques et les méthodes logiques. Dans le cas des méthodes numériques, les règles régissant l'évolution du contexte ne sont pas explicitées, l'évolution du contexte est gouvernée par l'application de modèles numériques probabilistes (ex. réseau bayésien pour [Castro *et al.*, 2001]) ou de méthodes de classification supervisée (ex. réseaux de neurones pour [Choi *et al.*, 2005], figure 3.2.4). L'intérêt des méthodes numériques d'évolution du contexte est qu'elles sont bien adaptées à un problème récurrent sur les informations contextuelles : le problème de la qualité des informations renvoyées par les capteurs. En effet, comme le soulignent [Bettini *et al.*, 2010], les systèmes sensibles au contexte doivent souvent faire face à des données contextuelles imprécises, voire contradictoires dues par exemple à des imprécisions dans les mesures fournies par les capteurs. Dans ce type de situation, l'utilisation, par exemple, de réseaux de neurones offre des réponses opérationnelles pour la classification des contextes (et donc la déduction d'information de plus haut niveau à partir d'informations de bas niveau) même dans le cas où les informations sont incomplètes, imprécises ou contradictoires.

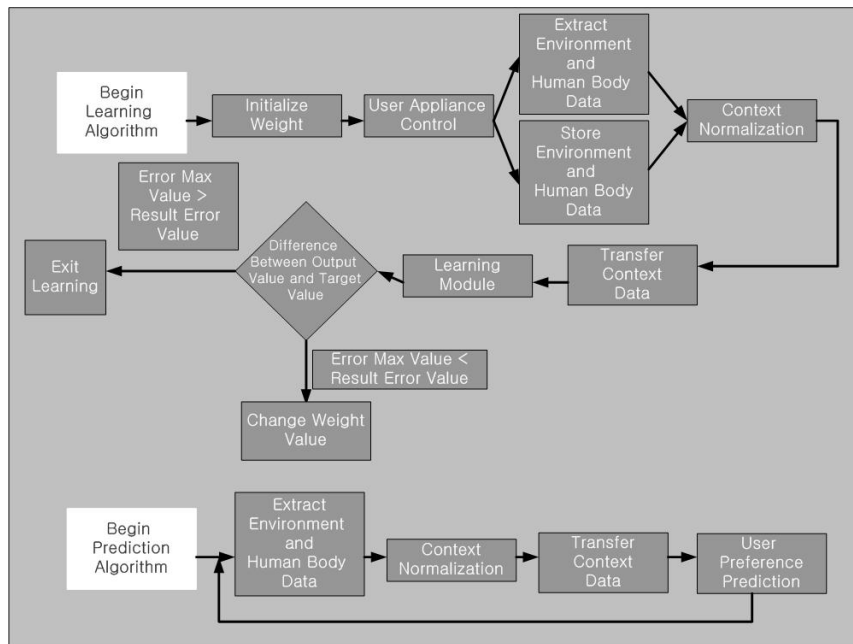


FIGURE 3.2.4: Algorithme numérique pour l'apprentissage du contexte d'après [Choi *et al.*, 2005]

L'autre classe de méthodes permettant de faire évoluer le contexte ou de déduire des contextes de haut niveau à partir de contextes de bas niveau sont les méthodes logiques ou sémantiques. Ces méthodes nécessitent au préalable une modélisation sémantique du contexte. L'utilisation d'ontologies pour la représentation d'informations contextuelles comporte de nombreux avantages [Baumgartner & Retschitzegger, 2006] : les ontologies sont un moyen efficace d'intégrer des informations hétérogènes sur les objets, la sémantique des ontologies permet de définir explicitement et sans ambiguïté les situations à traiter (ex. situation à problème) enfin les ontologies et plus particulièrement le web sémantique permettent le partage simple d'informations entre des agents intelligents à différents niveaux d'abstraction (un agent pourra souhaiter connaître la température, un autre savoir simplement si « il fait chaud »).

Les possibilités d'inférences propres aux langages ontologiques permettent l'évolution du contexte par application directe de la sémantique de l'ontologie. Ainsi, pour l'ontologie SOUPA [Chen *et al.*, 2004] modélisant les principes du *Region Connection Calculus*, des inférences peuvent être faites sur les positions des personnes (ex. si une personne est dans une pièce, elle est dans le bâtiment qui contient cette pièce). Dans les cas complexes où l'évolution du contexte ne peut pas être déduite directement de la sémantique de l'ontologie, des règles sémantiques peuvent être mises en place. Ces règles utilisent typiquement le formalisme Semantic Web Rule Language (SWRL) proposé par le W3C qui permet d'associer un ensemble d'antécédents à une ou plusieurs conséquences. La

figure 3.2.5 illustre la mise en place de règles de ce type dans le projet SAWA (Situation Awareness Assistant)[Matheus *et al.*, 2000], un système d'aide à la décision destiné aux militaires visant à simplifier l'intégration d'une multitude de données issues de capteurs sur le terrain pour offrir une vue synthétique de la situation et des problèmes potentiels (ex. Est ce que l'unité X peut-être ravitaillée?).

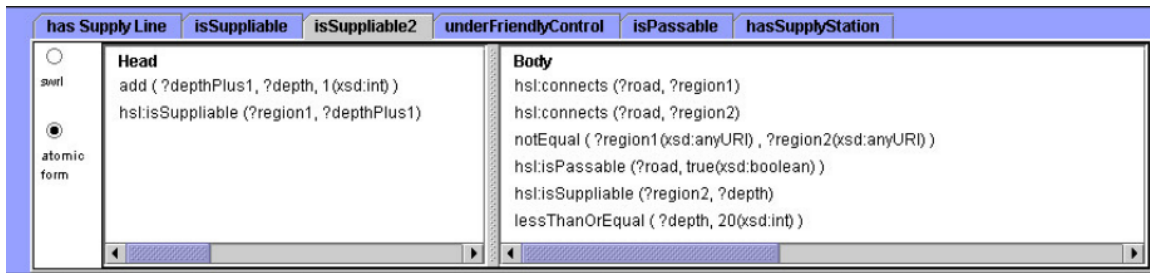


FIGURE 3.2.5: Edition d'une règle SWRL dans SAWA [Matheus *et al.*, 2000]

Echange et requêtes d'informations contextuelles

La question de la comparaison des contextes et du partage d'informations contextuelles est cruciale dans les systèmes sensibles au contexte. On peut distinguer deux cas : le partage de contexte entre différents agents (humains ou machines) et la comparaison ou les calculs contextuels à visée applicative (ex. proposer une interaction particulière suivant le contexte de l'utilisateur).

Les modèles du contexte permettant le partage d'informations sont le plus souvent des modèles sémantiques, ce type de modèle permettant en effet d'avoir simplement un accord préalable sur la définition des informations contextuelles (l'ontologie). L'utilisation d'une ontologie permet en effet de définir le vocabulaire commun entre plusieurs entités possédant une représentation du contexte, permettant ainsi d'obtenir la même interprétation des données contextuelles. Ainsi dans le cadre de *Context-Aware Chat*, [Ranganathan *et al.*, 2002] proposent l'utilisation d'un langage logique (proche d'une logique de description) pour la description du contexte. Ce langage est partagé entre deux utilisateurs du *chat* situés dans deux environnements physiques différents. Le *Context-Aware Chat* définit deux canaux de transmission de l'information : le canal classique d'échange de texte et le canal contextuel qui permet l'échange d'informations exprimées dans le langage logique. Les éléments transférés dans le canal contextuel sont envoyés automatiquement et permettent à chaque utilisateur du *Context-Aware Chat* d'avoir une vision sur ce qui se passe « de l'autre côté » par exemple l'entrée d'une nouvelle personne ou le début d'une réunion.

Les opérations sur le contexte à visée applicative sont le plus souvent basées sur la formulation de règles. Les règles de la forme SI-ALORS ont pour préconditions un état

donné du contexte et déclenchent une modification de l'état de l'application lorsqu'elles sont exécutées. Les premiers systèmes sensibles au contexte utilisaient des opérations de *matching* simples entre attributs et valeurs pour exécuter ces règles contextuelles. Avec l'arrivée des nouveaux formats de modélisation du contexte, basés sur des ontologies, les règles ont pu gagner en complexité. Les règles modernes peuvent utiliser le langage SPARQL pour déterminer les conditions d'applications de la règle (SI) et déterminer les changements d'état de l'application ou les propositions d'interactions. Ainsi, [Haake *et al.*, 2010] proposent la règle contextuelle de la figure 3.2.6 pour déterminer les artefacts informatiques « importants ».

```
CONSTRUCT {?sub ?pred ?obj}
WHERE {
  ?actor1 hasLocation ?location .
  ?actor2 hasLocation ?location .
  ?actor1 worksOn ?artifact .
  ?actor2 worksOn ?artifact .
  ?location definesPlaceOf ?shareableDisplay .
}
```

FIGURE 3.2.6: Règle contextuelle proposée par [Haake *et al.*, 2010]

Cette règle contextuelle est basée sur l'utilisation de CONSTRUCT, un opérateur SPARQL permettant de créer de nouveaux graphes RDF lorsqu'il existe un matching entre la base de connaissances et les clauses définies après le WHERE. Ici, le SI de la règle correspond donc à l'existence d'un tel matching.

3.3 Informatique ubiquitaire et systèmes sensibles au contexte

3.3.1 Architecture des systèmes sensibles au contexte

Deux principaux types d'architecture ont été proposés pour les systèmes sensibles au contexte: une approche basée sur des **widgets contextuels** et une approche basée sur un **modèle unifié du contexte**. L'approche basée sur les widgets, introduite par [Salber *et al.*, 1999] consiste à définir des briques logicielles réutilisables (les widgets) chargées de requêter les données contextuelles issues des capteurs et de les présenter dans un formalisme de haut niveau aux applications. Le contexte est ainsi fragmenté en un ensemble d'informations indépendantes et des règles fines peuvent être établies sur chaque type de données contextuelles comme le proposent [Gandon & Sadeh, 2004] dans l'application *eWallet*. Un exemple de règle proposée dans l'application *eWallet* sont les

règles de partage des informations contextuelles, l'utilisateur pouvant décider de manière très fine quels agents (humains ou logiciels) auront accès à sa localisation. L'intérêt de cette approche est sa modularité, de nouveaux widgets pouvant facilement être introduits pour capturer de nouvelles données contextuelles. Mais le principal problème de ce type d'architecture est également lié à sa modularité. Le contexte étant fragmenté en de multiples valeurs indépendantes, il est difficile de raisonner sur la situation, en tant que regroupement des données contextuelles.

Les approches basées sur un modèle unifié du contexte utilisent une représentation informatique de l'environnement reflétant en permanence l'état de l'environnement physique et de l'utilisateur (profil, tâches...). Ces modèles sont essentiellement basés sur des ontologies de domaine, représentant les relations spatiales et temporelles entre entités de l'environnement, ainsi que les informations et tâches courantes de l'utilisateur. La représentation informatique de l'environnement peut alors être interrogée par des applications sensibles au contexte via des agents comme proposé par [Chen *et al.*, 2003]. L'intérêt de ce type de représentation est que l'on retrouve la notion de situation de l'utilisateur à travers le modèle unifié du contexte et qu'il est alors possible d'établir des règles de haut niveau sur cette situation.

3.3.2 Interaction avec les systèmes sensibles au contexte

La particularité, en terme d'interaction homme-machine, des systèmes sensibles au contexte est de placer l'interaction sur deux plans : les interactions classiques et les interactions implicites. Dans un article fondateur, [Schmidt, 2000], Schmidt établit une analyse de l'interaction entre les humains et les systèmes perceptifs, c'est-à-dire les systèmes sensibles au contexte. Schmidt établit une analogie entre l'interaction homme-machine et les communications entre humains. Une particularité importante des communications entre humains est que le sens de la communication est en grande partie implicite, par la prosodie, le langage corporel, les expressions faciales... ces informations qui permettent de donner du sens aux communications entre humains sont le plus souvent inconscientes, tant pour l'émetteur que pour le récepteur.

Suivant Schmidt, de manière similaire les systèmes perceptifs ont pour finalité d'intégrer des interactions implicites, c'est-à-dire des actions de l'utilisateur qui n'ont pas comme finalité première d'interagir avec le système mais permettent cependant ces interactions. Il introduit pour cela le terme de IHCI (Implicit Human Computer Interaction), on retrouve dans ce concept l'idée de *calm technology*, s'adaptant de manière autonome à l'utilisateur [Weiser, 1991]. Cependant, ces technologies adaptatives ne sont pas sans poser problème en terme d'ergonomie : les mêmes actions dans différents contextes n'auront pas forcément les mêmes résultats, ce qui pose un problème dans l'évaluation ergonomique des systèmes sensibles au contexte. L'évaluation des interfaces mobiles devient délicate puisqu'il s'agit d'étudier à la fois le comportement et les interactions

explicités de l'utilisateur ainsi que le contexte dans lequel ces interactions se produisent [Plouznikoff & Robert, 2004].

Pour concevoir ces nouveaux modes d'interactions entre l'utilisateur et les systèmes sensibles au contexte, [Plouznikoff *et al.*, 2006] proposent un modèle de la médiation homme-machine basé sur deux axes : **l'évaluation**, axe quantifiant comment la perception du réel par l'utilisateur est enrichie par la machine, et **l'exécution** axe quantifiant le plan où se situent les interactions entre l'homme et la machine (plan réel ou plan virtuel). Ce modèle peut être vu comme une interprétation, dans le cadre de l'informatique ubiquitaire, des distances d'évaluation et d'exécution de la théorie de l'action de [Norman, 1986]. Suivant ces deux axes, [Plouznikoff *et al.*, 2006] définissent trois catégories d'interactions entre un utilisateur et un dispositif mobile : **l'altération virtuelle**, la **virtualisation** et la **diffusion** (figure 3.3.1).

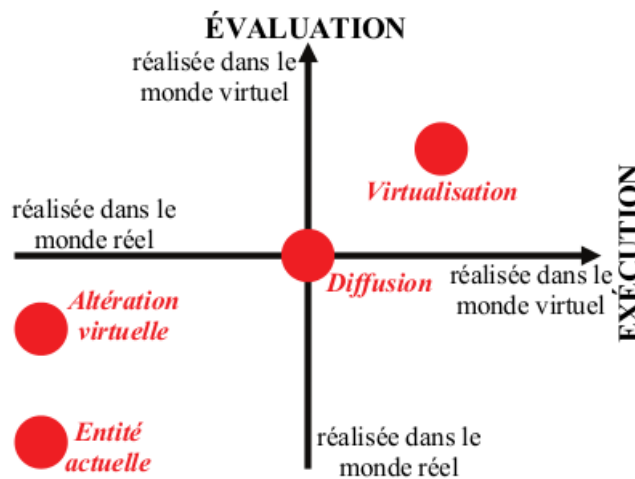


FIGURE 3.3.1: Médiation utilisateur-monde réel [Plouznikoff *et al.*, 2006]

Cette classification s'inscrit initialement dans le cadre où l'utilisateur a une tâche précise à accomplir utilisant un ou plusieurs appareils dans le monde réel, tâche qui est assistée par l'ordinateur porté. La classification est cependant suffisamment générique pour s'inscrire dans le cadre applicatif de ce travail, l'assistance à la visite de musée. Nous détaillons dans la suite ces trois types d'interactions en positionnant notre travail dans cette classification.

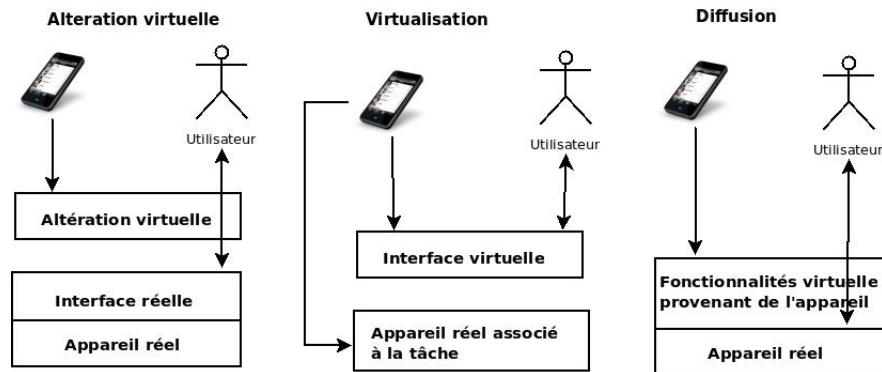


FIGURE 3.3.2: Les trois catégories de médiation proposées par [Plouznikoff *et al.*, 2006]

Les trois types de médiation sont représentés en figure 3.3.2. Dans **l'altération virtuelle**, les interactions se font directement dans le monde réel. L'utilisateur dispose cependant *via* un périphérique mobile d'informations complémentaires sur le monde réel afin de l'assister dans ces actions, par exemple via la réalité augmentée. Une illustration de ce niveau est l'application d'aide aux sapeurs-pompiers proposée par [Luyten *et al.*, 2006]. Les sapeurs-pompiers disposent via un périphérique mobile d'informations signalant leurs positions respectives et les zones à risques (fumées, feux...) mais agissent effectivement dans le monde réel (figure 3.3.3).



FIGURE 3.3.3: Situation d'altération virtuelle pour un sapeur-pompier en exercice [Luyten *et al.*, 2006]

Dans **la virtualisation**, l'utilisateur est en relation avec un ou plusieurs objets du monde réel mais l'évaluation et l'exécution se font uniquement par l'intermédiaire d'un appareil mobile. Notre approche pour l'assistance à la visite de musée se situe dans ce cadre. Les utilisateurs n'interagissent pas directement avec les œuvres, mais le périphérique mobile leur permet d'obtenir des informations complémentaires sur les œuvres (évaluation) ainsi que dans une certaine mesure d'interagir avec les représentations sémantiques de ces œuvres (*via* par exemple l'utilisation d'annotations numériques sur les œuvres physiques).

Enfin le troisième type de médiation, **la diffusion des interfaces** s'apparente dans une certaine mesure aux interfaces tangibles. Il s'agit de distribuer les fonctions nécessaires à l'accomplissement d'une tâche au travers des objets présents autour de l'utilisateur. L'utilisateur interagira donc bien avec les objets réels associés à la tâche mais ces objets seront augmentés par des fonctions proposées par l'ordinateur porté. Cette approche est bien illustrée par le système proposé par [Pham-Nguyen, 2010]. Les utilisateurs du système sont des vendeurs spécialisés dans le domaine de l'audiovisuel. L'appareil mobile utilisé est un PDA sur lequel les vendeurs peuvent consulter des informations sur les produits qu'ils proposent en fonction de questions des clients. En fonction du contexte de l'utilisateur (présence ou non d'écran LCD autour de lui), ce dernier peut faire passer les informations affichées du PDA à l'écran LCD, afin de faire visualiser au client des éléments pertinents sur le produit qui l'intéresse. On a donc bien une diffusion des fonctionnalités du PDA sur d'autres objets afin d'assister l'utilisateur dans la réalisation de sa tâche.

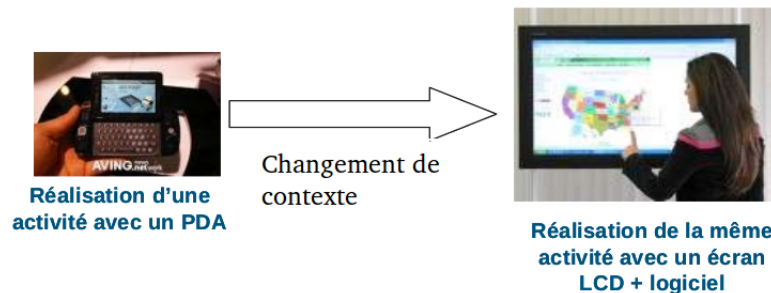


FIGURE 3.3.4: Système de diffusion d'interface pour l'assistance à la vente en audiovisuel [Pham-Nguyen, 2010]

3.3.3 Systèmes sensibles au contexte pour l'apprentissage humain

Ces dernières années, l'utilisation de périphériques mobiles et l'informatique ubiquitaire ont suscité l'intérêt croissant des concepteurs d'EIAH. Plusieurs termes sont apparus pour qualifier ces nouvelles formes d'apprentissage : *mobile learning*, *pervasive learning*, *ubiquitous learning*... sans que les définitions de chacun de ces termes soient nécessairement cohérentes d'un auteur à l'autre. Nous reprendrons ici les définitions les plus communément acceptées en considérant que le *mobile learning* (apprentissage mobile) relève du *e-learning* pratiqué à l'aide d'un dispositif mobile. Le *pervasive learning* (apprentissage pervasif) est l'apprentissage dans des environnements réels augmentés de capacités sensibles et l'*ubiquitous learning* (apprentissage ubiquitaire) est la conjonction du *mobile learning* et du *pervasive learning*, c'est à dire l'utilisation de dispositifs mobiles comportant des capacités sensibles (ou dans un environnement comportant des

capacités sensibles) pour mettre en oeuvre l'apprentissage [Ogata & Yano, 2003]. Nous nous intéresserons ici plus particulièrement aux systèmes d'apprentissage ubiquitaires.

L'intérêt pédagogique principal de l'apprentissage ubiquitaire est la possibilité de mettre en place des situations d'apprentissage authentiques. En effet, l'acquisition de nouveaux savoirs est plus pérenne lorsque les connaissances peuvent être contextualisées au travers de situations authentiques, définies par [Brown *et al.*, 1989] comme les situations où l'apprenant peut évaluer par lui-même l'utilité et l'application dans le monde réel de ces connaissances. L'informatique ubiquitaire permet la mise en place de situations où l'apprenant se voit proposer des connaissances adaptées à son contexte immédiat et opérationnalisable directement.

Afin d'appréhender de manière systématique les environnements d'apprentissage ubiquitaires, on peut positionner ces environnements sur un axe correspondant au degré de spécification de la tâche que doit réaliser l'apprenant (figure 3.3.5). La distinction est en effet très nette entre les systèmes proposant un modèle formel de la tâche ou des tâches à accomplir, l'apprenant devant en permanence être situé sur un scénario d'exécution et les systèmes faisant appel à la sérendipité, proposant des informations ou interactions en fonction du contexte sans que l'apprenant soit situé dans un modèle de tâche.



FIGURE 3.3.5: Environnements d'apprentissage et spécificité des tâches à accomplir

Nous décrivons rapidement les systèmes présentés dans la figure 3.3.5 qui nous fourniront également un bref panorama des formes que peut prendre l'apprentissage ubiquitaire. Au niveau le plus à gauche de l'axe se trouvent les systèmes où aucune tâche n'est définie. Le système CLUE [Ogata & Yano, 2003] est un système d'aide à l'apprentissage du Japonais pour des étudiants américains. Le principe de fonctionnement du système est de proposer des mots ou expressions japonaises sur un dispositif mobile en fonction de la position de l'étudiant sur le campus (ex. cafétaria). Les étudiants peuvent également s'envoyer des messages concernant des expressions inconnues, les messages étant alors orientés vers l'étudiant ayant déjà eu accès à l'expression. Ce système d'apprentissage juste à temps est typique des environnements d'apprentissage où la notion de tâche ou de scénario d'exécution est absente. Le système CHIP [Hage *et al.*, 2010], que nous évoquerons plus longuement dans le chapitre suivant est une assistance mobile à la visite de musée. L'utilisateur se voit proposer des suggestions d'œuvres à considérer au fur et à mesure de son parcours et suivant son profil. Ces deux systèmes sont typiques d'une approche informelle de l'apprentissage ubiquitaire où le dispositif est un support à

l'activité que décide d'entreprendre l'utilisateur et non un prescripteur de cette activité.

Au centre de l'axe se trouvent les systèmes d'apprentissages dans lesquels les tâches sont de haut niveau et peu formalisées. Ce sont le plus souvent des systèmes pour l'apprentissage scolaire. Les tâches en question sont proposées par l'enseignant mais sont représentées très simplement en interne (ex. pas de hiérarchie de tâches). MECOCO [George & Lekira, 2009] est un système d'aide à la communication contextualisée entre étudiants dans le cadre d'un parcours de musée. Les étudiants ont pour tâche spécifiée de visiter le musée et annoter les œuvres. La représentation interne de la tâche est de très haut niveau (ex. visiter le Musée du Louvre, guider les visiteurs). Le système *Museum Detective Guide* proposé par [Thom-Santelli *et al.*, 2005] est un support interactif à la visite de musée destiné aux plus jeunes. Lors du parcours du musée, les élèves se verront proposer des exercices élémentaires n'ayant pas de lien les uns avec les autres. Ici encore la tâche globale (visiter le musée en utilisant le système Museum Detective) est prescrite par l'enseignant mais les utilisateurs-apprenants conservent une grande liberté d'action. Ce type de système s'inspire d'une approche constructiviste de l'apprentissage en laissant ouvertes les possibilités et en laissant l'apprenant construire son savoir à travers les interactions, non nécessairement prévues initialement, avec le dispositif.

Les systèmes à droite de l'axe sont ceux où le modèle de tâche est représenté informatiquement de manière explicite et souvent complexe (ex. tâches/sous-tâches, préconditions et postconditions sur les tâches). Ce type de système se destine le plus souvent à l'apprentissage professionnel sur le terrain, pour lequel il est important de situer précisément l'apprenant dans l'ensemble du processus afin de lui offrir les informations suffisantes et adaptées à sa situation. En effet dans le cadre d'un apprentissage sur le terrain il sera très difficile à l'apprenant de prendre du temps pour rechercher l'information précise dont il a besoin sur le dispositif. Il s'agit donc de lui fournir cette information *just in time* en fonction de l'état de la tâche. Deux exemples issus du projet Européen PLearnNet illustrent particulièrement bien cette approche orientée tâche. Le premier système concerne la formation de postiers en utilisant des PDA [Caron *et al.*, 2008]. Les procédures métiers (et les liens entre ces différentes procédures) sont modélisées de manière fine et associées à des positions géographiques, capturées par le GPS du PDA. Lorsque l'utilisateur consulte le dispositif, il est donc situé de manière précise dans un scénario d'exécution et le dispositif peut lui proposer les informations concernant les tâches ou sous-tâches qu'il doit réaliser. Le deuxième système est proposé par [Pham-Nguyen & Garlatti, 2008] et concerne la formation continue des vendeurs en électronique dans un hypermarché. Les auteurs proposent une hiérarchie de tâches, chaque tâche étant associée à une ou plusieurs méthodes de résolutions. De la même manière que pour la formation des postiers, l'apprenant est en permanence situé dans la hiérarchie de tâches, ce qui permet de lui proposer des méthodes appropriées. Ces derniers systèmes laissent donc moins de liberté réelle à l'utilisateur et se rapprochent plus d'une formation sur le terrain traditionnelle, avec un formateur humain assistant l'apprenant dans sa démarche. L'intérêt de ces systèmes n'est cependant pas de remplacer complètement

le formateur mais plutôt de permettre à un formateur de suivre plusieurs apprenants en même temps.

3.4 Conclusion

Nous avons présenté dans ce chapitre les stratégies de modélisation du contexte et les solutions pour opérationnaliser ces modèles. Il apparaît que la modélisation sémantique offre de nombreux avantages en terme d'expressivité et de généralité. Par ailleurs, la modélisation sémantique facilite l'opérationnalisation des modèles du contexte (évolution/transition du contexte, comparaisons ou création de liens entre plusieurs contextes). Nous décrirons dans le chapitre 5 notre proposition pour la modélisation sémantique du contexte.

Par ailleurs, l'interaction homme-machine dans les systèmes sensibles au contexte doit être pensée de manière différente de l'interaction homme-machine traditionnelle. En particulier, il convient de définir les interactions implicites entre l'utilisateur et le système sensible au contexte. Dans le cadre de notre travail, nous avons choisi une stratégie de virtualisation, dans laquelle l'utilisateur interagit avec des représentations augmentées des objets réels (les œuvres).

Enfin, du point de vue pédagogique, nous nous situons dans une approche informelle de l'apprentissage. Notre cadre applicatif est en effet l'assistance à la visite de musée, il est important pour le visiteur de conserver un sentiment de liberté tout au long la visite (cf. chapitre 1). L'idée n'est donc pas d'imposer un scénario et des activités de visite, mais de proposer un système offrant des informations et des activités complémentaires suivant la situation et les désirs du visiteur.

Chapitre 4

Systèmes sensibles au contexte pour l'assistance à la visite de musées

La recherche sur les systèmes mobiles d'assistance à la visite de musée a été particulièrement importante ces dernières années. Le musée est en effet un terrain d'expérimentation très intéressant de par son accessibilité, la variété des thèmes et des types de musées (histoire, sciences et techniques, arts...) et la variabilité des visiteurs (classes, groupes, visiteurs individuels). Par ailleurs, les études systématiques sur les modes de visite, sur les typologies de visiteurs et sur l'apprentissage en musée se sont développées (chapitre 1) et donnent des directions relativement précises pour la mise en place de ces systèmes.

Deux catégories de travaux émergent pour l'assistance mobile à la visite de musée : les travaux centrés sur des « tâches » à effectuer dans le musée et les travaux centrés sur la navigation. Les travaux centrés sur les tâches sont typiquement destinés à des élèves du primaire ou du secondaire en groupe organisé, l'idée dominante dans ces systèmes étant d'informatiser des activités de visite préexistantes ou de créer de nouveaux scénarios de visite. Les systèmes centrés sur la navigation concernent typiquement le visiteur adulte seul ou en petit groupe. Ces systèmes ambitionnent de fournir un accès alternatif aux collections. Ils utilisent parfois les technologies de l'ingénierie des connaissances pour sélectionner et proposer des informations d'intérêt à l'utilisateur. Notre approche de l'assistance à la visite se place dans cette deuxième catégorie de travaux.

Dans ce chapitre, nous présenterons tout d'abord les deux types d'approche. Nous nous centrerons ensuite plus spécifiquement sur les systèmes basés sur les connaissances pour l'assistance à la visite de musée. Nous préciserons les limites des systèmes actuels en terme de prise en compte de la didactique muséale et des interactions possibles.

4.1 Approches pour l'assistance à la visite de musées

4.1.1 Approches orientées tâche

Nous regroupons dans les *approches orientées tâche* les systèmes de visite essentiellement centrés sur des activités proposées par le dispositif. Ce dernier n'est plus uniquement un support informatif, mais un prescripteur de l'activité du visiteur. Il propose généralement une série d'activités impliquant l'étude des artefacts et des échanges entre apprenants. Ces systèmes sont essentiellement à destination des publics scolaires. Ils comprennent le plus souvent des outils pédagogiques permettant à l'enseignant de mettre en œuvre les trois temps de la visite : la préparation de la visite, la visite en elle-même et le retour d'expérience après la visite. Les phases de préparation et de retour d'expérience étant typiquement mises en œuvre via des sites Web, qui offrent des ressources pédagogiques permettant, par exemple, de revenir sur les œuvres vues [Semper & Spasojevic, 2002].

Museum Detective Guide [Thom-Santelli *et al.*, 2005] illustre tout à fait l'approche orientée tâche. Le dispositif est destiné aux enfants et reprend des exercices pédagogiques existant sur support papier en les portant sur un dispositif mobile (tablette). Les élèves sont groupés par paires, et le parcours est imposé. Devant chaque œuvre, une série de questions à choix multiples doit encourager les élèves à considérer correctement l'œuvre en question (ex. *What do you think this object is made of?*). Les bonnes réponses apportent un complément d'information et les mauvaises réponses des indices pour déterminer la bonne réponse. Outre les traditionnels QCM, les jeunes visiteurs peuvent prendre part à des jeux plus élaborés, comme la reconstruction de la « Tour de garde », figure 4.1.1. Des questions plus ouvertes comme « Que pensez-vous que la statue dirait si elle pouvait parler ? » permettent de déclencher des débats entre apprenants sur les objets exposés. L'évaluation du projet a été effectuée par comparaison avec un système similaire à *Museum Detective Guide* mais utilisant uniquement le papier/crayon comme support (pour les QCM et les activités de jeux). Les résultats suggèrent fortement un meilleur rappel des informations par le groupe utilisant la version électronique.



FIGURE 4.1.1: Construction d'une tour de garde dans *Museum Detective Guide*

Le système *MYST* [Laine *et al.*, 2010] est également centré sur l'accomplissement de diverses missions par l'utilisateur. Ces missions sont cependant introduites à travers un scénario plus global, faisant appel à une scénarisation pédagogique et à des NPC (*Non Player Character*) guidant le jeune visiteur à travers les diverses missions suivant des dialogues prédéfinis. Cette scénarisation est effectuée via les dialogues avec les NPC, ces derniers représentant des personnes en lien avec le thème du musée (ex. scientifique dans le cas d'un musée des sciences). Par ailleurs, *MYST* offre la possibilité au visiteur d'enregistrer (de manière vocale) ses impressions sur les œuvres ou sur la galerie. Ces enregistrements audios seront ensuite disponibles sur un site spécialisé. Les interactions entre utilisateurs du système ont lieu via des *battles*, c'est-à-dire des successions rapides de quizz (figure 2) où chaque joueur se voit attribuer un score de classement global. L'évaluation du système n'a porté que sur le ressenti des utilisateurs et la facilité à utiliser le système. Ces évaluations sont extrêmement positives en terme de satisfaction, cependant on peut se demander dans quelle mesure l'attrait de la nouveauté joue dans la satisfaction des jeunes visiteurs.

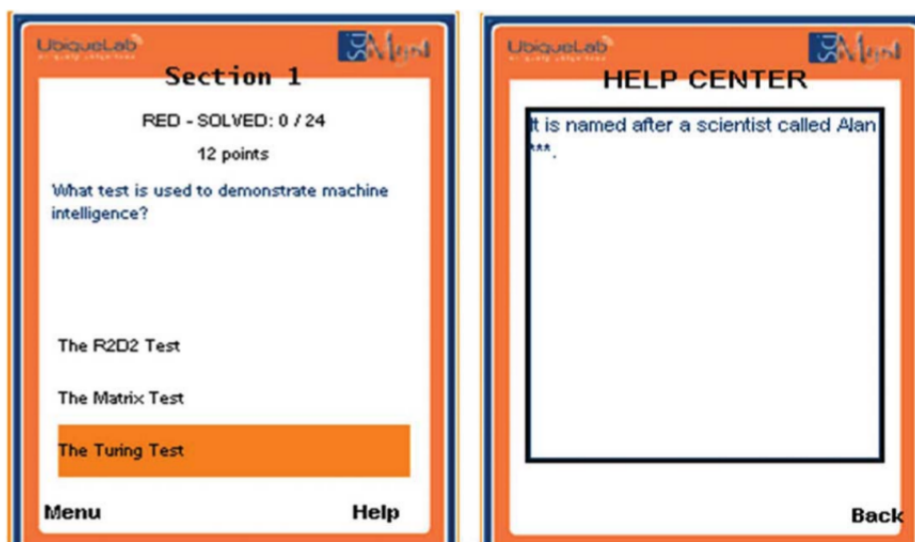


FIGURE 4.1.2: Une énigme dans MYST

De nombreux autres systèmes mobiles ont été conçus pour l'assistance mobile orientée tâche (non nécessairement limités à la visite de musée), CAESARUS [Naismith *et al.*, 2005], LORAMS [Ogata *et al.*, 2009], IPerG¹ (2008)... Différents types de scénarios peuvent être envisagés : jeux de rôles, simulations, puzzles... On pourra trouver une revue de ce type de systèmes dans [Damala, 2009]. Malgré des différences de surface, ces systèmes partagent un certain nombre de points communs. Ils se situent dans une perspective

1. <http://www.pervasive-gaming.org/>

constructiviste de l'apprentissage en permettant au visiteur de construire ses représentations à travers l'interaction. Ils proposent tous des jeux, sous différentes formes, ils favorisent l'interaction sociale entre les utilisateurs et ils permettent généralement de préparer et de revenir sur la visite *via* des sites spécialisés.

Ce type d'approche est intéressante car elle permet de maintenir l'intérêt tout au long de la visite du musée, ce qui est parfois délicat pour les jeunes visiteurs. Par ailleurs ces systèmes, de par leur *design*, facilitent les interactions sociales qui se construisent autour des œuvres, par exemple en proposant aux jeunes visiteurs des questions ouvertes sur les œuvres qui leur permettent de débattre. Ces interactions sociales, à travers l'échange de points de vue, rendent les œuvres plus accessibles. L'œuvre de musée a en effet un statut quasiment mythique, ce statut pouvant bloquer la réflexion ou créer une gêne pour le visiteur qui ne se sent pas capable de commenter ou de donner ses impressions. Par des *design* bien étudiés, les systèmes orientés tâche permettent de surmonter cette gêne et offrent un accès alternatif aux œuvres. Enfin, ces systèmes permettent généralement l'enregistrement de commentaires associés aux œuvres, ces commentaires permettant à l'enseignant de revenir sur les différents points de la visite en classe.

Ces systèmes semblent donner de bons résultats en termes d'implication du public et de plaisir ressenti à la visite. Cependant ils présentent plusieurs problèmes. Le premier problème est que l'attention de l'utilisateur est en grande partie focalisée sur le dispositif. Ce problème est général dans le cadre de l'assistance à la visite de musée par un dispositif mobile mais apparaît de manière beaucoup plus intense dans les systèmes orientés tâche, l'utilisateur étant sollicité de manière régulière pour accomplir des tâches sur le dispositif. [Thom-Santelli *et al.*, 2005] notent que dans le cas du système *Museum Detective Guide*, certains des jeunes utilisateurs passait nettement plus de temps à interagir avec le dispositif qu'à contempler les œuvres.

Le deuxième problème réside dans l'évaluation de l'apprentissage induit par le dispositif. Les systèmes orientés tâche ont en effet clairement un objectif pédagogique, mais les évaluations réalisées jusqu'à présent portent essentiellement sur la facilité d'interaction, le ressenti de la visite... On peut se demander si les systèmes orientés tâche ne dénaturent pas la visite de musée. Les jeunes visiteurs ayant utilisé ce type de système ne risquent-ils pas de trouver une visite classique sans intérêt par rapport à leurs expériences précédentes ? De la même manière, si le ressenti de la visite est très positif, n'est ce pas dû essentiellement à l'attrait de la nouveauté pour les dispositifs mobiles ?

Enfin, le dernier problème, plus fondamental au niveau informatique, est que les systèmes orientés tâche sont prévus pour un musée en particulier. La scénarisation et le *design* des activités d'apprentissage sont spécifiquement conçues pour ce musée. Le passage à un autre musée implique donc une refonte complète des scénarios et des activités, rendant ce type de systèmes difficilement généralisable.

4.1.2 Approches orientées navigation

Les approches orientées navigation ont pour objectif de conserver la liberté de visite tout en apportant des compléments d'information adaptés au bon endroit et au bon moment. Elles se distinguent des approches orientées tâche par le public cible (essentiellement des adultes), par la liberté plus importante laissée au visiteur et par les interactions proposées, le plus souvent implicites alors qu'elles sont explicites dans les systèmes orientés tâches.

Nous décrivons ici brièvement quelques systèmes, représentatifs de cette approche, pour aboutir dans la partie suivante aux systèmes basés sur les connaissances, généralement également orientés navigation.

Le système LISTEN, proposé par [Zimmermann & Lorenz, 2003] illustre particulièrement bien l'approche orientée navigation. Dans ce système, l'utilisateur est équipé d'un casque comme pour les audio-guides classiques, cependant, le visiteur ne dispose d'aucun moyen de contrôle direct sur les informations sonores qui lui sont fournies. Le contrôle se fait exclusivement par les mouvements du corps et l'orientation de la tête du visiteur dans l'espace. Pour cela, les auteurs proposent une partition de l'espace muséal, divisé en un ensemble de zones, chaque zone étant associée à un ou plusieurs artefacts. L'arrivée du visiteur dans une zone constitue une interaction implicite déclenchant un contenu sonore (figure 4.1.3) associé à cette zone. L'idée est donc de proposer une navigation dans un ensemble de documents sonores uniquement à travers les interactions implicites du visiteur.

Afin de proposer des informations pertinentes et intéressantes pour l'utilisateur, un modèle dynamique de ses intérêts est alimenté au cours de ses déplacements. Les seules formes d'interactions entre le visiteur et le système étant des interactions implicites (déplacements et arrêts du visiteur devant une œuvre), le système utilise les informations sur le temps resté devant chaque œuvre et le temps passé à écouter les commentaires pour construire un modèle des intérêts du visiteur. Par ailleurs, le système se base sur les mouvements du visiteur pour lui associer un stéréotype de visite (proche des modèles du visiteur proposés par Véron et Levasseur présentés dans le chapitre 1), permettant par exemple d'encourager les « sauts » d'une œuvre à une œuvre lointaine pour les utilisateurs de types *sautering around*. L'évaluation du système a consisté à mesurer le retour en terme de satisfaction de 15 utilisateurs experts. Si la démarche synesthésique (associant au mouvement des informations sonores) a été perçue de manière positive, l'absence totale de possibilité de feedback explicite sur les informations transmises a été jugée assez négative par les utilisateurs.

Le projet PEACH, proposé par [Zancanaro *et al.*, 2003] se base également sur la localisation du visiteur mais propose une interaction multimodale via un PDA. L'originalité de ce projet est la génération automatique de clips vidéos suivant l'œuvre ou la partie d'œuvre vers laquelle l'utilisateur pointe son PDA (figure 4.1.4). Ces vidéos permettent de

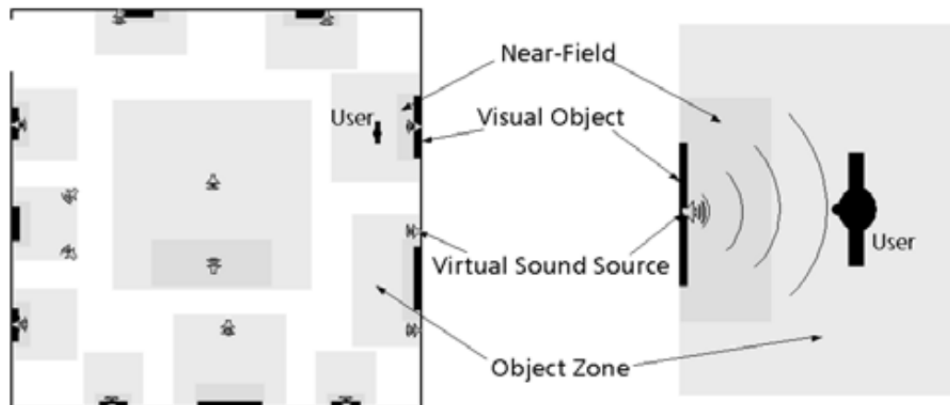


FIGURE 4.1.3: Interactions implicites dans LISTEN

souligner des points de détails de l'œuvre, de montrer l'œuvre dans son ensemble ou d'effectuer des zooms ou des recadrages. L'originalité principale de ce système est d'intégrer des règles de montage propres à la cinématographie (*shot segmentation, camera movement, transition effect*) afin de générer de manière automatique des séquences vidéos accordées aux textes de présentation des œuvres. Le projet PIL (Personal experience with active cultural heritage-Israel) proposé par [Rocchi *et al.*, 2004] est directement issu du projet PEACH. La différence essentielle entre les deux projets réside dans une modélisation plus fine des intérêts de l'utilisateur dans le projet PIL, grâce à un questionnaire initial permettant de choisir des thèmes ou des artefacts *a priori* intéressants. Par ailleurs, le projet PIL permet l'envoi de message d'un utilisateur à l'autre, sous une forme similaire à celle des SMS.

Le système *Leaving a trace*, proposé par [Stevens & Martell, 2003] ne se base pas sur un dispositif mobile mais sur des « kiosques », accessibles à proximité des œuvres. Ces kiosques consistent en des ordinateurs permettant d'obtenir assez classiquement des informations complémentaires sur les œuvres. L'originalité principale du système réside dans la possibilité de réaliser des commentaires audio ou vidéo sur les œuvres. Ces commentaires peuvent être organisés de manière spatiale et temporelle, l'utilisateur « pointe » une zone de la représentation de l'œuvre sur le kiosque et associe à cette zone le commentaire qu'il souhaite. Les commentaires vidéos concernent plutôt les expositions proposant des expériences scientifiques, ou le visiteur peut réaliser l'expérience tout en posant des questions à l'attention des futurs visiteurs. Les auteurs mettent en avant une forme de discussion indirecte permise par le système de trace, certains commentaires répondant directement aux commentaires précédents. Par ailleurs, les collections de traces peuvent aider les conservateurs ou les guides-conférenciers à mieux comprendre comment les visiteurs appréhendent les artefacts, et donc mener à une muséographie différente ou à des explications plus claires des œuvres exposées.

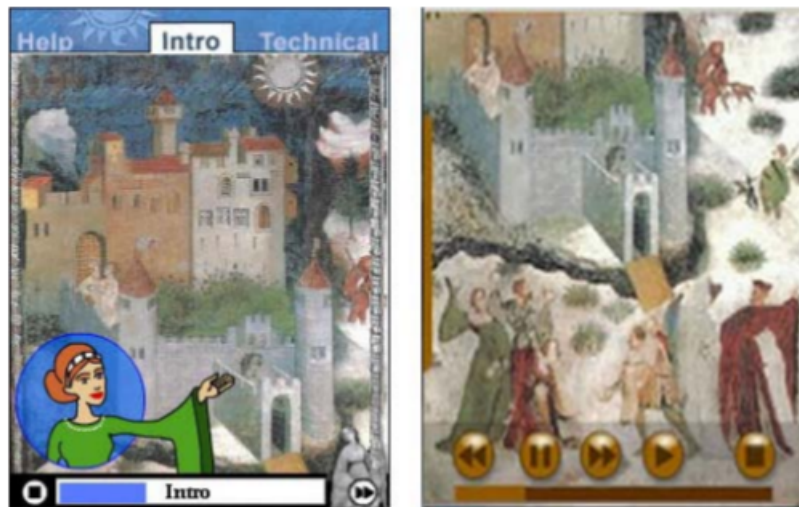


FIGURE 4.1.4: Captures d'écrans des vidéos générées dans le projet PEACH

De nombreux autres systèmes pour l'assistance à la navigation en musée ont été proposés. On pourra trouver une revue détaillée de ce type de système dans [Kuflik *et al.*, 2011]. Le point essentiel à retenir dans pour ces systèmes est le désir des concepteurs de minimiser les interactions explicites entre l'utilisateur et le dispositif pour les remplacer par d'autres formes d'interaction (mouvements du corps, de la tête, pointage...), ainsi que dans une certaine mesure, l'adaptation des informations proposées à un profil du visiteur, déduit de ses interactions implicites.

Les limites des systèmes décrits correspondent pour la plupart aux limites déjà identifiées sur les systèmes basés sur les tâches. Tout d'abord, les dispositifs numériques peuvent détourner l'attention du visiteur des œuvres exposées. Ensuite, si les évaluations de ces systèmes par les utilisateurs sont généralement très positives, il faut prendre garde au rôle que joue l'attrait de la nouveauté dans l'évaluation. Par ailleurs, l'objectif de ces systèmes est souvent ambigu, se situant entre une aide à la visite, un apport pédagogique, ou parfois une toute nouvelle expérience muséale (ex. LISTEN). Ces positionnements intermédiaires peuvent rendre les évaluations des systèmes délicates. Il n'existe d'ailleurs pas, à notre connaissance, de protocole expérimental faisant consensus pour les systèmes d'aide à la navigation. Par ailleurs, on doit noter que les interactions entre le dispositif et l'utilisateur sont très limitées dans ce type de système. Le seul type d'interaction où le visiteur est réellement acteur de sa visite est l'utilisation d'annotations.

4.2 Systèmes basés sur les connaissances pour l'assistance à la visite de musée

De par leur fonction de conservation et de préservation, les musées ont souvent d'immenses collections non accessibles au public ou accessibles lors d'expositions temporaires. Si jusque dans les années 1950-1960 l'indexation de ces collections était essentiellement basée sur des systèmes manuels (ex. fiches), l'informatisation a rapidement joué un rôle important dans l'indexation des artefacts.

Les premiers systèmes, fondés sur des bases de données classiques commencent à laisser la place à une description des œuvres basée sur les connaissances. Une réflexion globale sur les modalités d'indexation et de description sémantique des œuvres a d'ailleurs émergé, visant à unifier les différents vocabulaires de description du patrimoine, ou du moins à définir des langages « pivot » permettant de passer d'un mode de description à un autre, comme le démontre par exemple le projet Europeana².

Par ailleurs, ces descriptions sémantiques offrent des opportunités nouvelles pour l'accès et le parcours parmi des connaissances muséales : les systèmes basés sur les connaissances pour l'assistance à la visite de musée.

4.2.1 Systèmes basés sur les connaissances

Plusieurs systèmes basés sur les connaissances ont été proposés pour faciliter l'accès aux ressources culturelles, en mobilité, dans le musée. Nous présentons ici une revue des principaux systèmes existants et des modèles sous-jacents.

Le projet HIPPIE [Oppermann & Specht, 2000] a été un des premiers systèmes utilisant des techniques de raisonnement automatique couplées à un dispositif mobile dans le cadre de visite de musée. HIPPIE utilise une caractérisation des œuvres basée sur la taxonomie ICONCLASS (an ICONographic CLASsification System) qui est une classification des différents thèmes picturaux de l'art occidental. L'utilisateur est caractérisé dans HIPPIE par son intérêt pour ces différents thèmes, ainsi que par son mode de visite, basé sur la classification proposée par Veron et Levasseur (1983). Lorsque l'utilisateur se déplace dans le musée, le système, détectant sa position par une localisation radio, est capable de l'informer d'œuvres liées à ses centres d'intérêt situées à proximité et de présenter des informations liées à ces œuvres de manière appropriée. La modélisation sous-jacente à HIPPIE reste relativement simple pour les œuvres, les données métier provenant essentiellement de la taxonomie ICONCLASS.

Le projet CHIP [Wang *et al.*, 2009] a pour objectif initial de constituer un système de recommandation d'œuvres en fonction des intérêts de l'utilisateur. Une œuvre y est

2. <http://www.europeana.eu/>

décrite à l'aide de SKOS (Simple Knowledge Organization System)³, une ontologie permettant de décrire des vocabulaires contrôlés de type liste, taxonomie ou thésaurus. Le modèle d'œuvre intègre des informations thématiques issues de ICONCLASS et de trois taxonomies artistiques publiées par le Getty Vocabularies Program⁴ : ULAN (Union List of Artist Names), TGN (Thesaurus of Geographic Names) et AAT (Art and Architecture Thesaurus). L'utilisateur peut, pour une œuvre donnée, attribuer une note d'intérêt global ainsi qu'une note d'intérêt pour les différentes caractéristiques définissant l'œuvre (artiste, style, thématique, etc.). Les différents termes des vocabulaires sont associés à des concepts d'une ontologie liés par des relations pondérées, le coefficient de pondération étant fixe et déterminé par le degré d'intérêt de la relation (par exemple, la relation entre un artiste et son maître a une pondération élevée car il est intéressant de connaître le maître lorsque l'on s'intéresse à l'artiste). Le système peut alors proposer à l'utilisateur des concepts ou des œuvres susceptibles de l'intéresser également (par exemple un visiteur aimant Magritte se verra suggérer les surréalistes), l'utilisateur pourra alors donner un score d'intérêts à ces recommandations (figure 4.2.1). Ce système a en outre fait l'objet d'une implémentation pour une utilisation mobile en musée, SPACE-CHIP [Hage *et al.*, 2010], permettant de construire dynamiquement un parcours en fonction des intérêts de l'utilisateur.

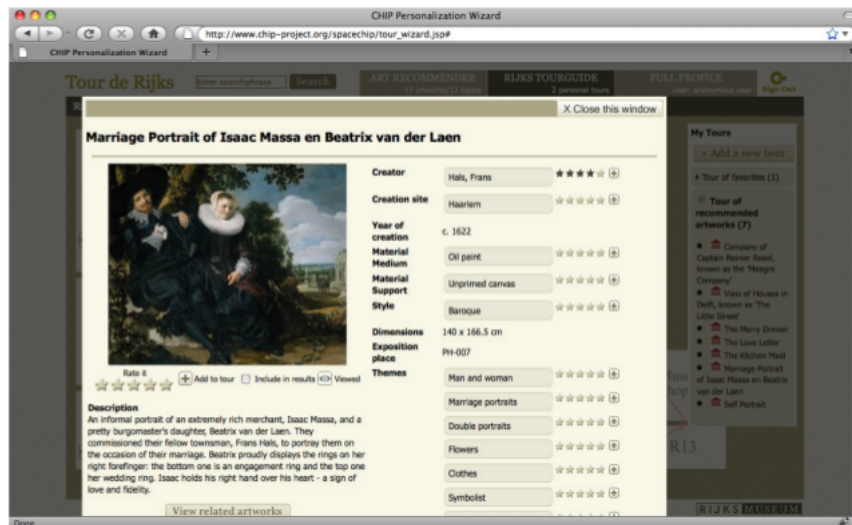


FIGURE 4.2.1: Attribution des notes d'intérêt dans le projet CHIP

Le système ec(h)o [Chou *et al.*, 2005] se distingue par l'utilisation unique du média sonore pour fournir des informations au visiteur. Les objets sonores et les artefacts muséaux sont décrits par des annotations sémantiques précisant les concepts, sujets et thèmes de ces objets. Le visiteur est, de manière similaire à HIPPIE, décrit par ses

3. <http://www.w3.org/2004/02/skos/>

4. <http://www.getty.edu/research/tools/vocabularies/>

intérêts pour ces différents concepts et par son mode de visite. Lorsque l'utilisateur s'approche d'une œuvre, des règles sémantiques sont utilisées pour déterminer parmi le corpus d'objets sonores, lesquels seront présentés à l'utilisateur. Ces règles permettent d'ordonner les objets audio à présenter au visiteur, en favorisant par exemple ceux qui n'ont pas déjà été présentés.

[Chou *et al.*, 2005] proposent un système de guidage et de communication entre visiteurs de musée basé sur l'architecture eWallet [Gandon & Sadeh, 2004]. L'originalité de ce système est de prendre en compte les préférences du visiteur en terme de vie privée. Le visiteur est décrit par ses intérêts, sous forme d'un ensemble de concepts reprenant les thématiques du musée, sa position et le temps qu'il souhaite consacrer à sa visite. Il peut alors autoriser ou interdire l'accès à ces informations aux autres visiteurs ou aux applications commerciales mises en place par le musée. Ainsi, il est possible de partager sa position avec les compagnons de visites mais d'en interdire l'accès à l'application de suggestions commerciales de la cafétéria du musée.

Le tableau 4.1 présente une synthèse des différents systèmes basés sur les connaissances suivant la représentation sémantique des œuvres, le modèle utilisateur et les mécanismes d'adaptation utilisés.

Projet	Modèle d'œuvre	Modèle utilisateur	Mécanisme d'adaptation
CHIP	Artiste, Thème, Lieu	Ensemble de concepts pondérés	Pondération des liens entre concepts
HIPPIE	Thème	Concepts d'intérêt, Mode de visite, Mode de visualisation, Position	Calcul de subsomption sur la hiérarchie ICON-CLASS
ec(h)o	Thème, Concept, Position	Concepts d'intérêt, Mode de visite, Position	Règles sémantiques permettant d'ordonner les documents
eWallet	Thème	Concepts d'intérêt, Temps à consacrer à l'exposition	Règles sémantiques pour le partage d'information contextuelles

TABLE 4.1: Modèles et mécanismes d'adaptation dans les système basés sur les connaissances

4.2.2 Limites des systèmes basés sur les connaissances

4.2.2.1 Limite en termes d'interaction

La première limite des systèmes présentés concerne l'interaction avec ces systèmes à base de connaissances. En effet, cette interaction se limite essentiellement à la suggestion automatique d'œuvres ou de documents (qu'il s'agisse de document sonores ou écrits).

Ces limites rejoignent celles identifiées dans les approches orientées navigation, le visiteur reste un récepteur passif du discours muséal.

L'annotation est le seul moment où le visiteur est réellement actif et acteur. Cependant il semble qu'il soit possible de trouver une juste mesure entre les systèmes centrés sur les tâches, monopolisant l'attention du visiteur et les systèmes centrés sur la navigation, proposant des interactions très limitées. Nous tenterons à travers notre proposition de trouver ce juste milieu en conservant la liberté de navigation tout en proposant des opportunités d'auto-évaluation.

4.2.2.2 Limites pédagogiques

La deuxième limite est liée à la première et concerne l'apport en terme d'apprentissage des projets présentés. Bien que pour chacun de ces projets l'apprentissage soit considéré comme informel, il semble délicat de montrer que les visiteurs comprendront mieux le musée en utilisant ces systèmes. Un aspect assez significatif est la rigidité du système et l'absence de possibilité d'intégration de connaissances pédagogiques. Il est en effet difficile voire impossible pour le conservateur ou le guide-conférencier de paramétrer ou modifier le système pour l'adapter à son expertise, (ex. faire ressortir certaines informations ou suggestions à certains moments plutôt qu'à d'autres).

Outre la limite didactique qu'implique un système fonctionnant sans apport extérieur, on peut se poser la question de l'acceptabilité de ces systèmes pour les conservateurs et guides-conférenciers. L'absence de moyens de contrôle sur le système risque d'être un frein à l'adoption de ces systèmes. On peut dresser dans une certaine mesure un parallèle avec les EIAH plus classiques. Il est à présent généralement admis que l'adoption d'un EIAH dépend des possibilités de « bricolages pédagogiques » que l'enseignant pourra effectuer sur le système [Caron, 2007]. De la même manière un EIAH ubiquitaire doit offrir à l'enseignant des possibilités de contrôle et d'adaptation.

4.3 Conclusion

Nous avons présenté les différentes approches pour le support numérique à la visite de musée. On peut distinguer deux axes d'approche, l'approche orientée tâche et l'approche orientée navigation. La première approche, plutôt destinée aux plus jeunes, est centrée sur un ensemble d'activités ou de « missions » à remplir par l'apprenant. La deuxième approche est plutôt réservée aux visiteurs adultes et centrée sur des suggestions de documents ou d'œuvres susceptibles d'intéresser le visiteur.

Nous avons explicité les limites de ces systèmes en terme d'interaction et de prise en compte de la didactique muséale. Nous tenterons à travers notre proposition de dépasser ces limites en proposant un système situé entre les systèmes orientés tâche et les systèmes

orientés navigation. Nous porterons une attention particulière aux moyens de contrôle du système par les experts (conservateurs et guides-conférenciers) et les enseignants afin d'enrichir l'aspect didactique tout en favorisant l'acceptabilité du système.

Notre approche

Chapitre 5

Représentation sémantique du domaine et du contexte

Notre proposition pour l'assistance mobile à la visite de musée repose sur la mise en place de proximités sémantiques et contextuelles. Ces proximités permettent d'assister l'utilisateur tout au long de sa visite en lui fournissant des informations et des possibilités d'interaction en fonction de son parcours et de son profil. Ces informations et possibilités d'interaction comprennent la suggestion d'œuvres, d'auteurs, de styles... la possibilité d'auto-évaluer ses connaissances par des mini-jeux ainsi que la possibilité de communiquer avec d'autres visiteurs par le biais d'annotations contextualisées. Ces interactions seront décrites en détail dans le chapitre 6.

L'utilisation de proximités sémantiques, que nous détaillerons au chapitre 6, implique nécessairement un modèle sémantique du domaine. Nous détaillons ici notre proposition pour une représentation sémantique riche des œuvres (permettant par exemple de représenter le fait que deux œuvres sont issues d'un courant commun). En outre, afin que les interactions proposées soient contextuellement adaptées à l'utilisateur (par exemple en mesure de lui recommander de considérer une œuvre qui se situe près de lui), nous proposons une modélisation sémantique du contexte de la visite de musée.

Dans ce chapitre nous détaillons les méthodes et modèles employés pour la mise en place des représentations sémantiques des œuvres et du contexte de visite. En particulier, nous détaillons l'utilisation de l'ontologie CIDOC-CRM (Center for Intercultural Documentation-Conceptual Reference Model) [Doerr *et al.*, 2007] comme base de modélisation des œuvres. Cette ontologie étant destinée aux experts en muséologie, nous avons dû effectuer plusieurs transformations afin, d'une part, de proposer des mesures de proximités sémantiques ayant du sens pour des utilisateurs novices, et d'autre part, de rendre l'ontologie compatible avec nos sources de connaissances issues du web de données (chapitres 8 et 9). Nous présentons ensuite notre modèle sémantique du contexte, structuré en différents espaces contextuels alimentés de manière implicite lors

des déplacements du visiteur et de ses interactions avec le dispositif. Ces modélisations serviront de base pour les proximités sémantiques et contextuelles que nous décrirons au chapitre suivant.

5.1 Modélisation sémantique du domaine

Notre terrain d'application est l'apprentissage informel en musée et, en particulier, l'utilisation de connaissances formalisées pour faciliter cet apprentissage. De nombreuses propositions ont été faites pour la modélisation des artefacts muséaux : Europeana Data Model [Doerr *et al.*, 2010], LIDO [Autere & Vakkari, 2011], Art Museum Image Consortium (AMICO¹).

Ces propositions prennent source dans les importants besoins d'indexation des musées. En effet, la taille des collections et la grande variabilité des objets exposés ont amené les musées à définir de nombreux systèmes d'organisation permettant l'indexation et la recherche d'artefacts muséaux précis. Ces systèmes d'organisation étaient, avant les débuts de l'informatisation, constitués de fiches physiques. Cependant, dès le début de l'informatisation, les musées ont fait appel aux technologies de l'information pour l'indexation et la recherche d'informations dans leurs collections [Tallon & Walker, 2008].

A l'heure actuelle, les systèmes d'information propriétaires (essentiellement des bases de données) cohabitent avec des systèmes plus ouverts, basés sur des représentations sémantiques des artefacts muséaux. Les systèmes de représentation sémantiques actuels sont l'aboutissement d'un long processus impliquant l'ensemble des acteurs du monde muséal. Cependant, ces représentations ne sont pas destinées au grand public, elles sont plutôt destinées aux experts (ex. les conservateurs) ou aux échanges d'informations entre musées. Notre terrain d'application est l'apprentissage en musée ; le public cible n'est donc pas constitué de spécialistes. Par ailleurs, nous souhaitons utiliser des proximités sémantiques pour définir de nouveaux modes de navigation parmi les œuvres. Un certain nombre de questions portant sur la modélisation du domaine (c'est-à-dire le patrimoine culturel en général) vont donc se poser : quelles représentations des connaissances choisir ? Comment intégrer des connaissances "extérieures" (ex. vocabulaires contrôlés) au sein de la représentation des connaissances choisies ? Quelles modifications apporter aux représentations existantes afin d'intégrer notre objectif de définition de proximités sémantiques pour l'interaction ?

5.1.1 Sources de connaissances utilisées

L'un des objectifs de notre système est d'aider l'utilisateur à comprendre les différents types de liens existant entre les œuvres. Nous avons donc besoin d'une représentation

1. <http://www.amico.org>

riche et précise des œuvres qui permette de mettre en lumière les différents aspects suivant lesquels les œuvres peuvent être comparées. En considérant les travaux existant dans le domaine, nous avons retenu l'ontologie CIDOC-CRM et les vocabulaires contrôlés ICONCLASS (ICONography CLASSification) et Getty-AAT (Art and Architecture Thesaurus) pour la représentation des connaissances muséales (tableau 1).

Nom	Modèle	Relations	Exemple d'éléments
CIDOC-CRM	Ontologie (RDFS)	Hierarchique Propriété Sous-Propriété	Concept d'artefact (<i>cidoc:man-made-thing</i>)
ICONCLASS	Taxonomie	Hierarchique	Scène biblique
Getty-AAT	Thésaurus	Hierarchique Association	Style Louis XIV

TABLE 5.1: Sources de connaissances utilisées

- L'ontologie CIDOC-CRM est l'ontologie de référence pour la description sémantique du patrimoine culturel. Elle définit les concepts d'événement, d'œuvre, de personne, de lieu... en utilisant le formalisme RDFS. Un intérêt particulier de CIDOC-CRM concerne la réutilisation et l'import de connaissances issues d'autres formalismes de représentation de connaissances (que ces formalismes soient basés sur des ontologies ou non). En effet, cette ontologie est devenue un standard (ISO 21127) et des alignements ont déjà été définis permettant l'import, dans une base de connaissances basée sur CIDOC-CRM, de connaissances muséales décrites dans la plupart des autres formalismes². Cette généralité, ainsi que la pérennité de CIDOC-CRM (garantie par son statut de standard ISO depuis 2006) nous a amené à choisir cette ontologie comme base de notre modèle sémantique du domaine. L'ontologie CIDOC-CRM est organisée suivant une dualité entre les objets physiques, instances de *cidoc:PersistentItem*, dotés de propriétés de spatialisation et les concepts temporels (*cidoc:TemporalEntity*) la rendant particulièrement adaptée à la description des relations patrimoniales. Un artefact muséal peut ainsi avoir une extension temporelle décrivant précisément son mode de création, les activités dans lesquelles il a été impliqué, ses possesseurs successifs, etc. Cependant, le caractère très générique de l'ontologie impose de la compléter par l'utilisation de vocabulaires contrôlés. En effet, CIDOC-CRM ne comprend pas de vocabulaire spécifique à l'art (ex. courant et thème d'une œuvre). Nous avons donc étendu CIDOC-CRM par l'ajout de vocabulaires contrôlés. Par ailleurs, CIDOC-CRM est relativement complexe à utiliser par rapport à nos objectifs de calculs de proximités sémantiques, nous détaillons dans la section suivante les modifications que nous lui avons apportées.
- La taxonomie ICONCLASS permet de caractériser les thèmes des œuvres les uns par rapport aux autres. C'est une classification exhaustive des thèmes artistiques de l'art

2. http://www.cidoc-crm.org/crm_mappings.html

occidental. Les termes y sont organisés sous forme d'une hiérarchie comportant au plus 10 niveaux, il n'existe pas de relation en dehors de la relation Parent → Enfant (pas de relations transversales) . La figure 5.1.1 représente la description du thème « Lois Juives sur le blasphème ». Ce thème est décrit suivant trois axes : un ensemble de termes pouvant le qualifier en plusieurs langages (via la relation *dc:subject*), un ensemble de termes préférés (via la relation *skos:preflabel*) et un ensemble de relations de type *broader*, établissant des relations de type générique/spécifique avec d'autres entités.

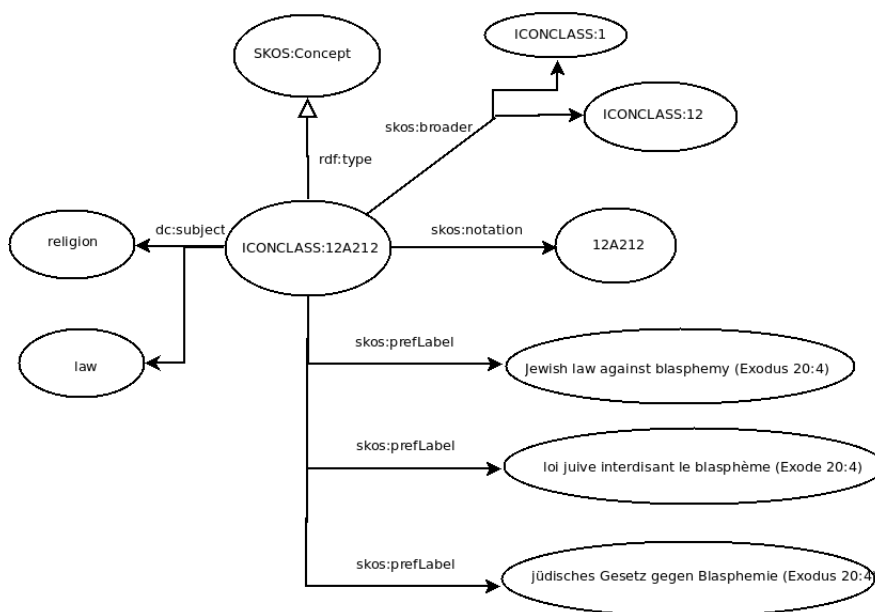


FIGURE 5.1.1: Description d'un élément d'ICONCLASS

Il est à noter que la taxonomie ICONCLASS est extrêmement précise et que les domaines couverts sont très larges. Ainsi il existe un concept exprimant le thème : « représentation d'une personne soulevant son chapeau pour saluer quelqu'un » (concept 33A11 de la taxonomie, figure 5.1.1).

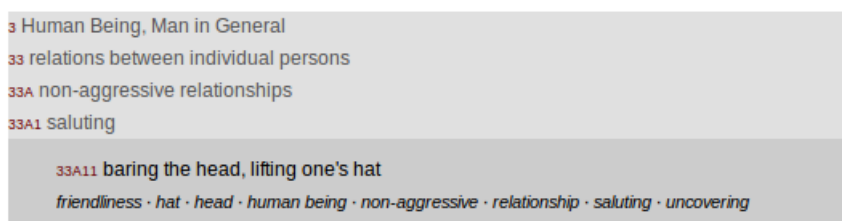
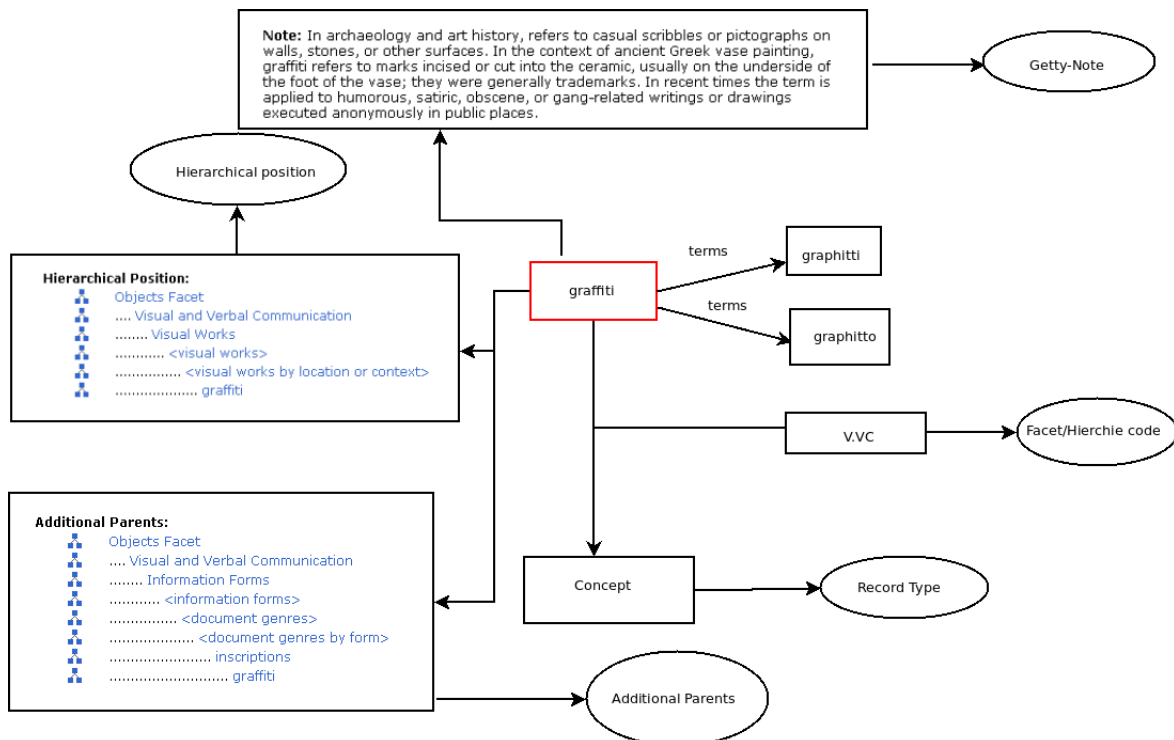


FIGURE 5.1.2: Concept 33A11 de la taxonomie ICONCLASS : soulever son chapeau

- Le vocabulaire Art and Architecture Thesaurus est développé par le Getty Institute. Il s'agit d'un thésaurus à facettes fournissant un vocabulaire contrôlé pour la description des qualités des œuvres et de leur mode de production. Les sept facettes du thésaurus sont organisées sous forme de plusieurs hiérarchies (une facette est associée à une ou plusieurs hiérarchies), des relations d'association et d'équivalence établissant des liens entre les hiérarchies. Les descriptions de Getty-AAT sont sensiblement plus complexes que les descriptions de concepts d'ICONCLASS. La figure suivante illustre la description du terme *graffiti*.

FIGURE 5.1.3: Représentation du concept de *graffiti*

Plusieurs différences existent avec la description des concepts dans ICONCLASS. Tout d'abord la présence d'une note au format textuel décrivant précisément l'élément considéré en langage naturel. Ensuite, chaque élément appartient à deux hiérarchies, la hiérarchie principale, *hierarchical position*, et la hiérarchie secondaire, *additional parents*. Cette structure s'explique car un concept de Getty-AAT peut avoir plusieurs parents, la hiérarchie principale donnant les parents « préférés ». Enfin, une dernière différence notable avec ICONCLASS est que les éléments de Getty-AAT ont un *Record Type*, caractérisant le type des éléments de la hiérarchie. Ici le *Record Type* est de type concept, mais il en existe trois autres types : *Guide Term*, *Hierarchy Name* et *Facet*. Un élément de type *Guide Term* permet de grouper des concepts liés entre-eux sans être lui-même un concept. Par exemple le *Guide Term* *<planographic printing processes>*, sans être

lui-même un concept, va permettre de grouper l'ensemble des concepts liés au processus d'impression de type planographie qui est un procédé d'impression sans creux ni reliefs. Les *Guide Term* servent donc essentiellement à structurer la hiérarchie. *Hierarchy Name* réfère simplement au concept de plus haut niveau d'une des hiérarchies. Enfin *Facet* identifie la facette du thésaurus à laquelle appartient la hiérarchie.

Dans notre application, nous n'utilisons que la facette *Style and Period*. La figure 5.1.2 offre une illustration de cette facette en représentant la position du concept *Style Louis XIV* dans la hiérarchie de Getty-AAT.



FIGURE 5.1.4: Position du concept de *Style Louis XIV* dans le thésaurus Getty-AAT

5.1.2 Modifications du modèle de CIDOC-CRM

L'ontologie CIDOC-CRM est en grande partie centrée sur le concept d'événement, toutes les entités physiques, sous-concepts de *cidoc:PhysicalObject* (ex. personne, artefact, bâtiment, collection) étant liées entre elles via des entités temporelles, sous-concepts de *cidoc:TemporalEntity*. Ainsi, il n'y a pas de relation directe pour spécifier qu'un artiste est l'auteur d'une œuvre. Pour exprimer une telle assertion, il faut associer l'artiste à un événement de type production (*cidoc:Production*) et associer l'œuvre à l'événement production comme étant le produit de cette production.

Ce type de modélisation, utilisant une instance intermédiaire entre deux entités, est assez classique. Il est en effet ainsi possible de créer une relation complexe entre les deux entités, via l'instance intermédiaire. Ainsi l'événement de type *cidoc:Production* sera associé à un lieu (lieu de création de l'œuvre), à une date (date de création de l'œuvre) et éventuellement à d'autres événements qui ont pu influencer la création. Cette modélisation est bien adaptée dans le cas de musées ou d'institutions patrimoniales. Ces institutions peuvent souhaiter, par exemple, retracer la succession d'événements (ex. legs, prêts, saisies) dans laquelle est impliqué un artefact muséal. La figure 5.1.3 illustre la modélisation de l'œuvre *Epitaphios* (une tapisserie religieuse du XVIIe siècle) en utilisant le modèle original; il s'agit d'une adaptation d'une des descriptions proposées dans la documentation de CIDOC-CRM.

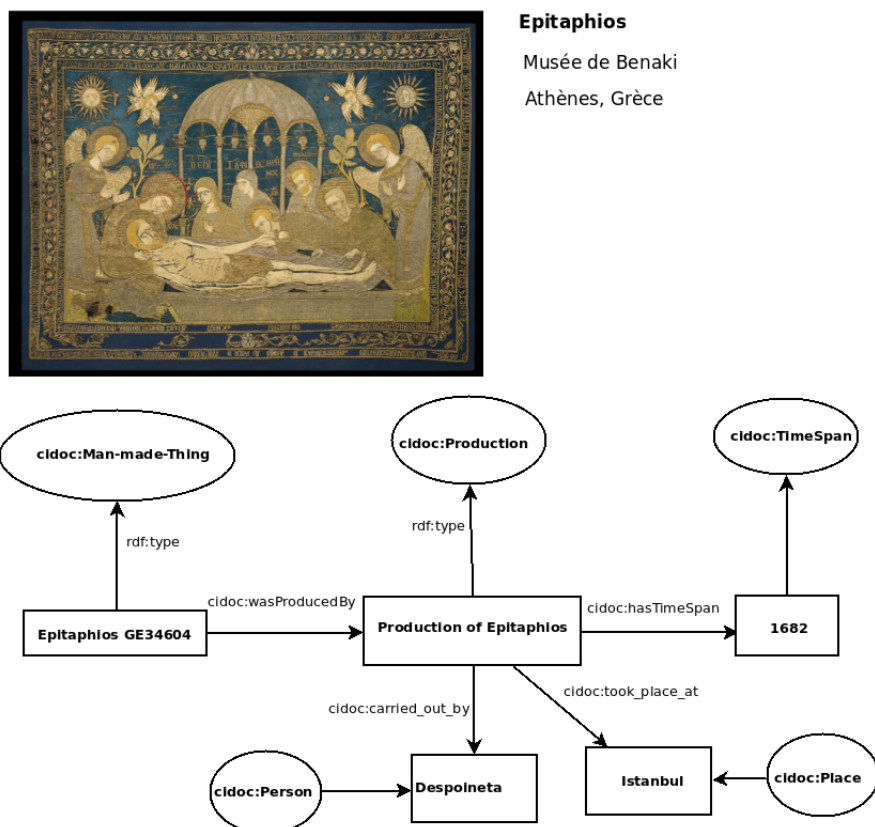


FIGURE 5.1.5: Une œuvre et sa modélisation dans CIDOC-CRM

On constate que l'élément Epitaphios est une instance de *cidoc:Man-made-Thing*. Cet élément physique est en lien avec l'entité temporelle *Production Of Epitaphios* (instance du concept temporel *cidoc:Production*), qui décrit les éléments propres à cette production : la date via la relation *cidoc:hasTimeSpan*, le lieu via la relation *cidoc:took_place_at* et l'artiste via l'événement *cidoc:carried_out_by*.

Ce modèle est riche ; il est cependant relativement difficile de le mettre en oeuvre directement. Le peuplement de la base de connaissances par les sources disponibles sur le web de données s'avère complexe si l'on suit le modèle original. En effet les bases de connaissances que nous avons considérées (chapite 9) sont moins riches en terme d'expressivité que le modèle de CIDOC-CRM. Par exemple le concept de production d'une oeuvre n'est pas décrit. Pour suivre le modèle CIDOC-CRM, il est donc nécessaire de créer une instance « artificielle » qui assurera le lien entre l'artiste et l'oeuvre.

Par ailleurs, comme nous l'avons précisé plus tôt, l'objectif de notre modélisation est de permettre des calculs de proximités sémantiques intelligibles pour les utilisateurs. Nous proposons donc plusieurs simplification du modèle CIDOC-CRM. L'objectif des simplifications du modèle, outre de faciliter l'intégration de connaissances externes, est

que ces nouvelles propriétés « parlent » mieux aux utilisateurs lors des comparaisons de deux œuvres : par exemple, au lieu de décrire deux tableaux comme ayant des événements de création proches, décrire deux tableaux comme étant proches au niveau de leurs dates de création.

Afin de faciliter le peuplement de la base de connaissances ainsi que de rendre les proximités sémantiques plus « naturelles » pour l'utilisateur final, nous proposons de réaliser un certain nombre de compositions de propriétés. C'est à dire que nous avons remplacé certains couples de triplets $(a P_i b)$ et $(b P_j c)$ par un triplet $(a P_k c)$. La figure suivante 5.1.6 illustre ce processus.

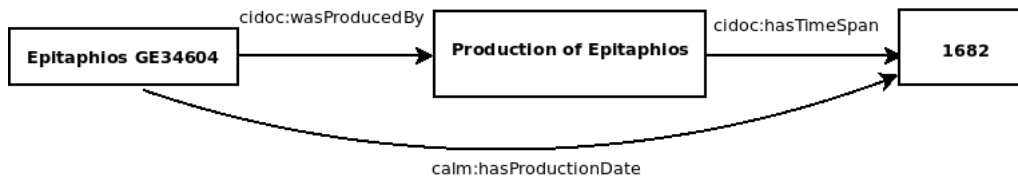


FIGURE 5.1.6: Composition de propriétés

Nous avons effectué des compositions de propriétés similaires pour les autres propriétés impliquant l'événement temporel *Production of Epitaphios*. Le résultat de ces compositions est la description simplifiée de la figure 5.1.7.

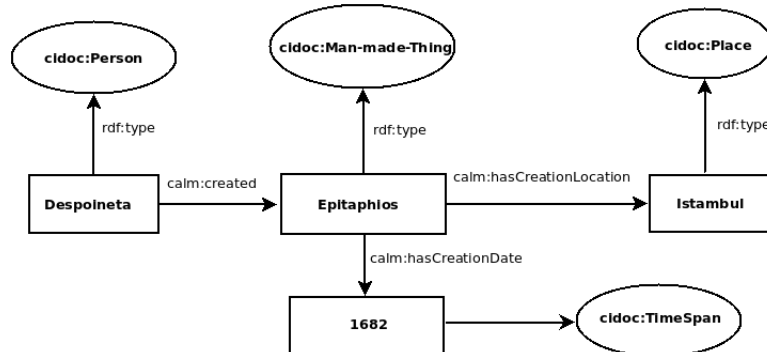


FIGURE 5.1.7: Liens entre l'œuvre et l'artiste après modifications

La disparition de l'instance *Production of Epitaphios* dans la figure 2 limite nécessairement les assertions réalisables. Par exemple, il n'est plus possible de lier différentes activités (ex. commande initiale du dessin, broderie) à la production de l'œuvre. Cependant les informations portées par l'instance temporelle sont pour la plupart trop spécifiques pour intéresser le visiteur du musée et cette perte d'expressivité est compensée par la simplification du modèle.

Nous avons effectué des simplifications similaires dans plusieurs autres cas. Nous illustrons ces modifications sur deux cas : le cas de la naissance ou de la mort d'une

personne et le cas où un artiste est influencé par un autre artiste. Le tableau 5.2 illustre ces modifications. La colonne **Modélisation dans CIDOC-CRM** décrit le modèle original, impliquant une instance de type temporel entre les deux informations qui nous intéressent (par exemple l'événement *cidoc:Death* entre une instance de *cidoc:Person* et sa date de mort). Pour des raisons de simplicité nous avons décrit les relations entre propriété, domaine et codomaine par la forme suivante : (*cidoc:Domain cidoc:Property cidoc:Range*).

La colonne **Modélisation simplifiée** présente la modélisation simplifiée, résultat de la composition des deux propriétés de la colonne de gauche.

Relation	Modélisation dans CIDOC-CRM	Modélisation simplifiée
Naissance	cidoc:Person cidoc:broughtExistence cidoc:Birth. cidoc:Birth cidoc:hasTimeSpan cidoc:TimeSpan.	cidoc:Person calm:birthDate cidoc:TimeSpan.
Mort	cidoc:Person cidoc:tookOutExistence cidoc:Death cidoc:Death cidoc:hasTimeSpan cidoc:TimeSpan.	cidoc:Person calm:dieDate cidoc:TimeSpan.
Influence	cidoc:Person cidoc:wasInfluencedBy cidoc:Activity. cidoc:Activity cidoc:performedBy cidoc:Person.	cidoc:Person calm:hasInfluenced cidoc:Person.

TABLE 5.2: Modification apportées à CIDOC-CRM

Il est à noter que le modèle simplifié, s'il n'est pas compatible directement avec CIDOC-CRM, peut aisément être transformé en un modèle compatible. Il suffit pour cela de créer les instances temporelles correspondantes et de leur associer les propriétés que nous avons fait porter aux objets physiques. De la même manière, il est aisé d'adapter un modèle CIDOC-CRM existant pour qu'il soit conforme à notre modèle.

5.1.3 Modèle sémantique des œuvres : intégration des taxonomies dans CIDOC-CRM

L'intégration des vocabulaires contrôlés dans CIDOC-CRM pose deux problèmes : tout d'abord l'expression des taxonomies sous forme de hiérarchies intégrables dans CIDOC-CRM, ensuite l'établissement des nouveaux liens entre ces taxonomies et CIDOC-CRM.

En ce qui concerne ICONCLASS, il existe déjà une version de cette taxonomie exprimée en SKOS. Nous avons donc directement utilisé cette hiérarchie.

Concernant le thésaurus Getty-AAT, ce dernier est décrit dans un formalisme de base de données propriétaire. L'ontologie SKOS est un bon candidat pour le mapping de ce vocabulaire. En effet, SKOS est déjà utilisé pour ICONCLASS et est spécifiquement

conçu pour permettre l'intégration de systèmes d'organisation de termes (taxonomie, thésaurus, etc.) sous forme d'ontologies. Nous avons donc effectué un mapping de Getty-AAT vers SKOS. Pour ce mapping, nous avons conservé les éléments dont le *record type* était *Concept* ou *Guide Term* (5.1.1) que nous avons transformés en instances de *skos:Concept*. Nous avons reproduit la hiérarchie définie par les relations *hierarchical position* et *additional parent* via les relations *skos:broader* et *skos:narrower*.

La deuxième étape consiste à intégrer les vocabulaires exprimés grâce à SKOS dans CIDOC-CRM. L'objectif de cette intégration étant de pouvoir réaliser des assertions du type : (*cidoc:LaJoconde calm:hasTheme iconclass:Portrait*). Nous avons alors utilisé un mécanisme prévu par les concepteurs de CIDOC-CRM en utilisant la classe *cidoc:E55.Type*, cette classe étant spécifiquement conçue pour servir d'interface entre des vocabulaires métier et les autres classes de CIDOC-CRM. Chaque élément des taxonomies ICONCLASS et Getty-AAT est donc une instance de *cidoc:E55.Type*. De plus, CIDOC-CRM fournit la propriété générique *cidoc:hasType*, ayant pour domaine le concept racine de l'ontologie *cidoc:E1.CRM-Entity* et donc l'ensemble des concepts. Cette propriété permet de spécifier un type pour chaque instance de CIDOC-CRM, ce type correspondant à une caractérisation fine issue d'un vocabulaire métier. Nous avons donc déclaré les concepts issus du mapping SKOS des vocabulaires comme étant des instances de *E55.Type* et défini deux sous-propriétés de *cidoc:hasType* : *calm:hasTheme* et *calm:hasStyle*.

Après intégration des vocabulaires, nous aboutissons à la représentation des œuvres (instances de *cidoc:Man-Made-Thing*), dont un modèle simplifié est présenté figure 5.1.8. Une œuvre est produite par un artiste ayant éventuellement un maître, elle possède un style et un thème, et fait référence à des événements et à des personnes historiques. Les personnes peuvent être impliquées dans des événements, ces derniers étant liés entre eux par des relations de causalité. Pour des raisons de lisibilité, nous n'avons pas représenté sur la figure les relations présentées dans la figure 5.1.7.

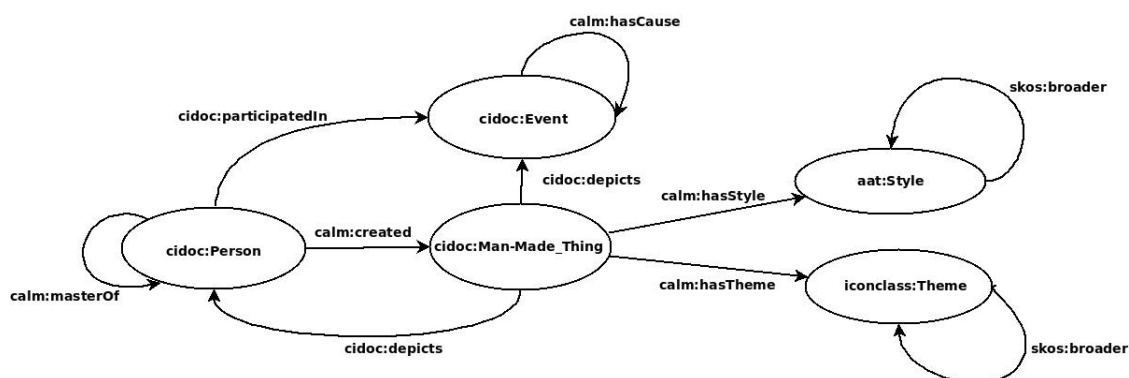


FIGURE 5.1.8: Modèle sémantique de l'œuvre

5.1.4 Conclusion sur la modélisation sémantique du domaine

Nous avons présenté notre proposition pour la modélisation sémantique du patrimoine culturel. Cette modélisation se base sur l'ontologie de référence CIDOC-CRM. Il peut sembler paradoxal de tirer des conclusions génériques à partir de la modélisation d'un domaine particulier ; cependant, les contraintes de modélisation que nous nous étions fixées dépassent largement le domaine du patrimoine culturel. Ces contraintes étaient l'utilisation de modèles de connaissances pérennes et facilitant l'interopérabilité, l'intégration de connaissances extérieures aux modèles (vocabulaires contrôlés) et la question des modifications à apporter aux formalismes afin de faciliter leur utilisation dans le cadre d'une application grand public utilisant la proximité sémantique.

Nous avons choisi de baser notre modélisation sur l'ontologie CIDOC-CRM qui, de par les multiples *mappings* existants, peut être vue comme un "pivot" parmi les nombreux formalismes de descriptions des connaissances culturelles. Nous avons décrit l'intégration de vocabulaires contrôlés, plus spécifiques à notre domaine d'application, *via* l'utilisation d'un *mapping* entre ces vocabulaires et l'ontologie de description des connaissances SKOS. Enfin, nous avons proposé un certain nombre de modifications sur l'ontologie CIDOC-CRM ; ces modifications permettent d'une part de représenter les proximités entre objets modélisés de manière plus "naturelle" pour le grand public et d'autre part de faciliter l'import de données depuis les sources de connaissances du web de données.

5.2 Modélisation sémantique du contexte

Notre modélisation du contexte s'inspire de la définition de [Zimmermann *et al.*, 2007] (chapitre 3). Nous considérons le contexte d'une entité (cette entité étant ici le visiteur), comme un ensemble d'espaces structurés. Cependant, contrairement à Zimmermann nous ne considérons pas une définition « canonique » des catégories contextuelles (temporalité, relations, activité, spatialisation et individualité pour Zimmermann). En effet, un modèle du contexte basé sur ces catégories serait trop rigide et nous obligerait à des choix de modélisation arbitraires (ex. où placer dans ce modèle les intérêts de l'utilisateur ?). Nous proposons plutôt une structure générique des espaces contextuels et définissons grâce à cette structure trois espaces contextuels permettant de capturer finement le contexte d'une visite de musée : le contexte de localisation, le contexte historique et le contexte d'intérêt.

Nous décrivons dans la suite le principe général de la modélisation d'un espace contextuel avant d'explicitier les trois espaces contextuels. Contrairement au modèle des œuvres, qui est statique, le contexte est par essence dynamique. Nous décrivons donc les modalités de peuplement de ces espaces en fonction des interactions explicites et implicites de l'utilisateur avec le dispositif. Nous proposons finalement une illustration de notre modèle du contexte sur un exemple simple.

5.2.1 Structure du modèle sémantique du contexte

L'objectif de notre modélisation du contexte est de fournir à l'utilisateur des informations ou des propositions d'interactions qui soient contextuellement adaptées. Par exemple, si l'interaction proposée est un jeu de type QCM, il s'agit de proposer à l'utilisateur des questions en rapport avec les œuvres qu'il a vues précédemment ou avec les œuvres autour de lui. Les informations importantes du modèle du contexte sont issues de la base de connaissance muséale, l'espace contextuel servant uniquement à structurer ces informations de façon à ce que l'on puisse effectuer des requêtes du type « Quelles œuvres ont intéressé l'utilisateur ? » ou « Parmi les œuvres situées à proximité de l'utilisateur, quelles sont celles qu'il ne connaît pas et sont susceptibles de l'intéresser? ».

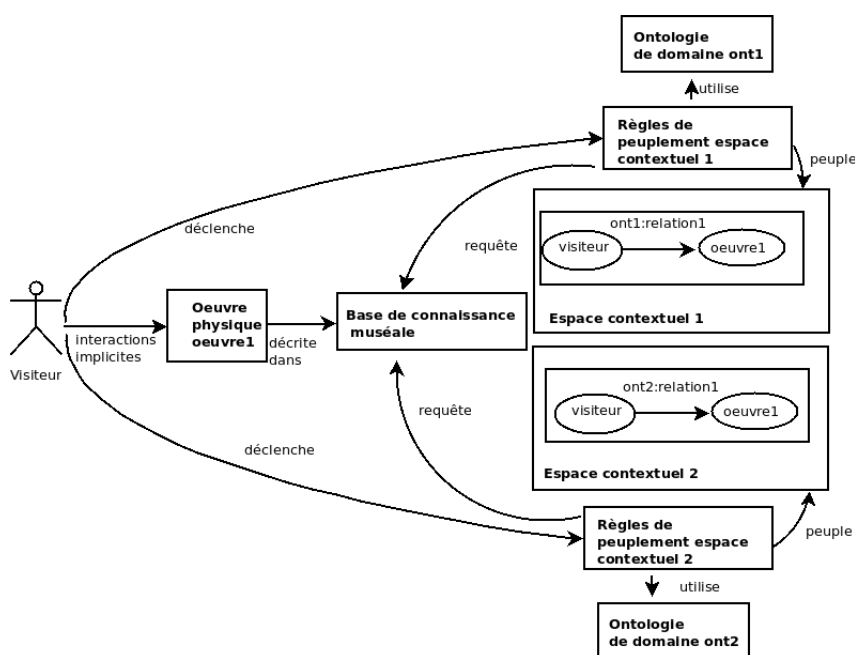


FIGURE 5.2.1: Peuplement des espaces contextuels

Le principe général de notre modélisation du contexte est qu'un espace contextuel est un espace structuré par une ontologie de domaine, cette ontologie de domaine définissant les relations entre le visiteur et les instances de la base de connaissances muséales. L'ontologie de domaine correspond donc au choix d'une catégorie contextuelle, décrivant une modalité d'interaction entre le visiteur et l'œuvre. Lors des interactions explicites entre le visiteur et le dispositif ou lors de ses interactions implicites (ex. déplacement), l'espace contextuel est alimenté d'assertions décrivant l'interaction entre le visiteur et l'œuvre. Un espace contextuel est donc une base de connaissances représentant un des modes de relation entre l'utilisateur et les objets du domaine muséal.

La figure 5.2.1 explicite ce processus : le visiteur déclenche une interaction implicite (ex.

arrêt devant l'œuvre) ou explicite (ex. consultation d'informations sur l'œuvre) avec une œuvre. Cette interaction va activer les règles de peuplement des espaces contextuels nommés **1** et **2** dans l'exemple. Les règles de peuplement utilisent une ontologie de domaine qui définit le mode de relation entre le visiteur et l'œuvre (ex. contempler l'œuvre, lire la notice...) et ajoutent de nouvelles assertions à l'espace contextuel.

En considérant les travaux dans le domaine de la muséologie et de la didactique muséale (chapitre 1), nous avons retenu trois espaces contextuels : le contexte de localisation, le contexte historique et le contexte d'intérêt. Nous décrivons dans la suite les modèles employés ainsi que la façon dont les contextes évoluent dynamiquement.

5.2.2 Les espaces contextuels

Contexte historique

Le contexte historique de l'utilisateur a pour objectif de capturer la temporalité dans la visite, c'est-à-dire la succession ordonnée d'œuvres qu'aura consultées l'utilisateur et les activités qu'il aura effectuées, ainsi que les coordonnées spatiales et temporelles de ses consultations ou activités.

Afin de modéliser le contexte historique, nous avons choisi d'étendre l'ontologie SEM (*Simple Event Model*) [van Hage *et al.*, 2011]. Cette ontologie est bien adaptée à notre problème car elle offre un ensemble de concepts fondamentaux (*sem:Event*, *sem:Place*, *sem:Actor*, *sem:Activity*) permettant la description sémantique du contexte d'un événement. De plus, l'ontologie SEM a été conçue pour être étendue via un mécanisme de typage des concepts fondamentaux qui peuvent être associés à des classes ou des instances de classes d'autres ontologies (c'est-à-dire être typés). Ce mécanisme permet d'assurer une extension aisée et de créer de nouveaux types d'événements.

Afin de modéliser l'historique de visite, nous avons spécialisé le concept *sem:Event* en *calm:ConsultationEvent*. Lorsque le visiteur s'arrête devant une œuvre, une instance d'événement consultation est créée et ajoutée au contexte historique. L'événement de consultation prend la forme suivante :

```
_b rdf:type sem:ConsultationEvent.  
_b sem:hasDate currentDate().  
_b sem:hasLocation <LOCATION>.  
_b sem:hasActor <VISITOR>.  
_b calm:involvesEntity <ARTWORK>.
```

Où *currentDate()* est une *built-in* qui renvoie la date et l'heure de la consultation et LOCATION, VISITOR et ARTWORK les URI respectives des instances représentant la localisation, le visiteur et l'œuvre considérée. Le contexte historique est constitué de

l'ensemble des événements de consultation ayant eu lieu durant la visite. Ce contexte est donc peuplé de manière transparente pour l'utilisateur durant la visite. De la même manière que pour le contexte de localisation, l'URI de l'œuvre considérée assure le lien avec la base de connaissances muséales.

Contexte d'intérêt

Afin de représenter les intérêts de l'utilisateur, nous avons, de la même manière, créé une spécialisation du concept *sem:Event*, *calm:InterestEvent*. Un événement d'intérêt représenté par une instance qui établit un lien entre le visiteur et un élément de la base de connaissance muséale :

```
_b rdf:type sem:InterestEvent.  
_b sem:hasDate currentDate().  
_b sem:hasActor <VISITOR>.  
_b calm:involvesEntity <MUSEUM_ENTITY>.
```

La différence entre un événement d'intérêt et un événement du contexte historique repose sur deux points : la description sémantique de ces deux événements n'est pas la même et la condition de création d'un événement d'intérêt est également différente. Concernant la description sémantique, les événements d'intérêt ne sont pas liés à la localisation, l'intérêt d'un utilisateur envers une entité étant supposé indépendant de sa localisation. Par ailleurs, un événement d'intérêt peut porter sur d'autres types d'entités que les œuvres (*via* la relation *calm:involvesEntity*) par exemple un artiste ou un style.

Concernant la condition de création, un événement d'intérêt est créé à la suite d'une interaction explicite entre l'utilisateur et le dispositif. Lorsque l'utilisateur effectue une manipulation de demande d'informations concernant une instance spécifique de la base de connaissances (ex. auteur, œuvre, thème...) un événement d'intérêt est ajouté au contexte d'intérêt liant l'utilisateur à cette instance précise.

Modèle de l'environnement physique, contexte de localisation

Notre modèle de l'environnement physique permet de représenter les différentes salles du musée, l'idée étant de décrire le musée sous forme d'un ensemble de mailles élémentaires, chaque maille étant un conteneur pouvant être associé à une ou plusieurs œuvres. Nous avons choisi pour modéliser cet environnement d'utiliser l'ontologie spatiale publiée par le laboratoire DAISy³. La maille élémentaire est une instance de *daisy:Location*, les relations entre les mailles sont assurées par la relation *daisy:adjacentTo*. Les liens entre les œuvres (instances de la base de connaissances muséales) et les mailles *daisy:Location*

3. <http://daisy.cti.gr/svn/ontologies>

sont assurées par la relation *daisy:IsInside*. La figure 5.2.2 illustre ce maillage de l'espace du musée.

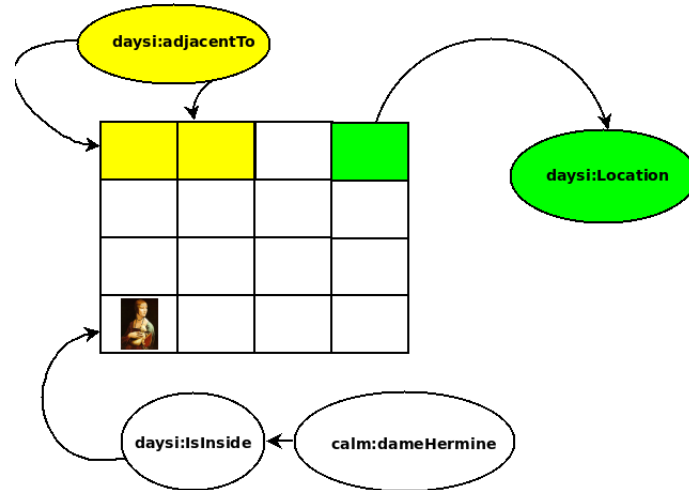


FIGURE 5.2.2: Modèle logique de l'espace physique du musée

Lors de ses déplacements dans le musée, le visiteur est en permanence associé à une maille de *daisy:Location* et aux mailles adjacentes. Le **contexte de localisation** du visiteur est constitué de cet ensemble de mailles, qui évolue au fur et à mesure des déplacements. Il est à noter que les instances d'œuvres liées aux instances de *daisy:Location* sont des éléments de la base de connaissances muséales. Connaître le contexte de localisation de l'utilisateur permet donc également de connaître les styles, auteurs, thèmes... présents autour de lui. C'est cette organisation qui sera la base dans le chapitre suivant des calculs de proximités contextuelles.

La figure 5.2.3 représente les différents espaces contextuels associés au visiteur.

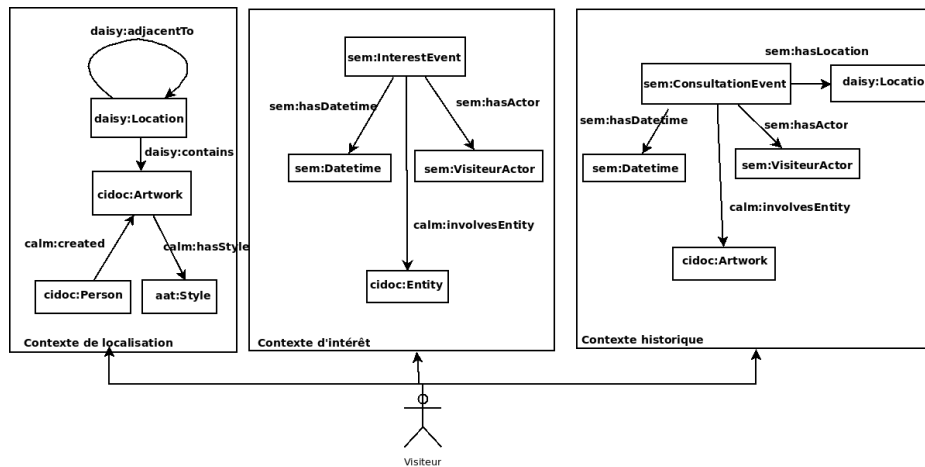


FIGURE 5.2.3: Espaces contextuels associés au visiteur

5.2.3 Illustration

Nous illustrons ici l'emploi de notre modèle du contexte sur un exemple simple. La figure 5.2.4 représente le parcours d'un visiteur dans le musée et les modifications des espaces contextuels suivant ses interactions implicites et explicites. La figure est organisée temporellement de la gauche vers la droite, la première étape étant le début de la visite.

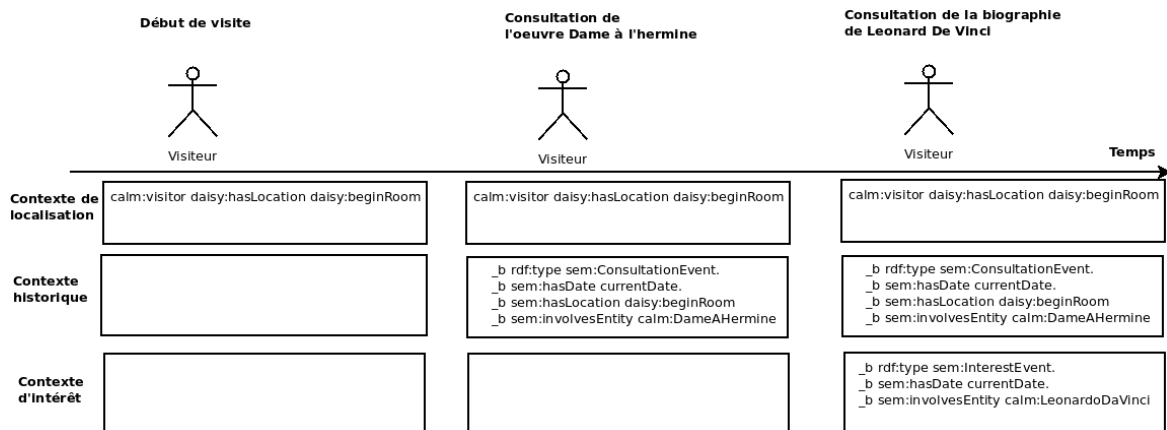


FIGURE 5.2.4: Évolution des espaces contextuels dans les premiers temps de la visite

Un point important à noter dans l'évolution des contextes est la différence de traitement entre le contexte de localisation, le contexte d'intérêt et le contexte historique. En effet, le nombre d'assertions décrivant le contexte de localisation n'augmente pas lorsque

l'utilisateur se déplace. L'ensemble des assertions associant l'utilisateur à une maille est supprimé et les nouvelles assertions sont ajoutées. Le nombre d'assertions du contexte d'intérêt augmente au fur et à mesure de la visite, les anciennes assertions n'étant pas supprimées lorsque le visiteur déclenche un événement d'intérêt. Cependant cette augmentation est relativement lente, le visiteur ne demandant pas *a priori* d'informations complémentaires pour chaque consultation. Enfin le nombre d'assertions du contexte historique augmente rapidement, les nouveaux événements de consultation étant ajoutés sans que les anciens soient supprimés. Ces propriétés des contextes de localisation, historique et d'intérêt seront utilisées dans le chapitre 7 avec l'ajout des règles pédagogiques de focalisation et relaxation contextuelles.

5.3 Conclusion

Nous avons présenté dans ce chapitre une modélisation des artefacts muséaux basée sur l'ontologie CIDOC-CRM et sur l'ajout de vocabulaires contrôlés dans cette ontologie. Nous avons proposé l'ajout de nouvelles propriétés à CIDOC-CRM visant à simplifier la représentation des liens entre artistes et œuvres. Ces simplifications rendent le nouveau modèle moins expressif, mais permettent cependant une meilleure intégration avec les connaissances sémantiques accessibles via le Web de données (chapitre 9). Par ailleurs, ces simplifications vont faciliter le calcul des proximités sémantiques que nous définirons dans le chapitre suivant.

Nous avons également proposé une modélisation sémantique du contexte de visite. On notera que si les entités "importantes" de ces espaces sont bien les instances de la base de connaissances muséales (ex. auteurs, œuvres, etc.), ces entités sont structurées différemment au sein de chaque espace (ex. temporellement via l'ontologie SEM pour la dimension historique). C'est cette structure qui nous permet dans la suite de définir des règles basées sur le contexte pour la suggestion d'œuvres et de manière plus générale pour l'ensemble des activités qui seront proposées à l'utilisateur.

Notation utilisées

Nous décrivons ici les choix de notation utilisés dans la suite du document.

Concept : les concepts d'une ontologie sont écrits en italique et commencent par une majuscule, un concept peut ne comporter qu'une lettre (ex *C*).

instance : les instances de concepts (c'est à dire les individus de la base de connaissance) sont écrit en italique en minuscule.

int[~]*xsd:integer* : les littéraux sont écrits en italique, commencent par une minuscule et sont suivis de leurs types.

P : représente une propriété quelconque de l'ontologie

P_k : indique une propriété particulière de l'ontologie, si $j \neq k$, $P_j \neq P_k$

P_k(a) : représente l'ensemble des entités liées à l'instance *a* par la propriété *P_k*, c'est à dire l'ensemble des entité *i* de la base de connaissance telle qu'il existe un triple (*a P i*)

|*E*| : représente le cardinal de l'ensemble *E*.

Max(v₁...v_n): désigne la fonction de \mathbb{R}^n retournant le maximum parmi (*v₁...v_n*)

M[,] : représente une matrice

M[i,j] : représente la composante de la matrice située sur la ligne *i* et la colonne *j*

Chapitre 6

Proximités sémantiques et contextuelles pour l'interaction

Les proximités sémantiques sont des outils puissants pour instrumenter les interactions entre un utilisateur et une base de connaissances. En effet, elles permettent de nouveaux modes de parcours parmi des connaissances structurées, *via* la réalisation automatique d'associations porteuses de sens entre des objets décrits sémantiquement. Il peut s'agir de descriptions sémantiques d'objets du monde réel, en l'occurrence les œuvres d'un musée. Les proximités sémantiques permettent donc d'instrumenter des interactions entre l'utilisateur et des objets réels, à travers l'utilisation d'un dispositif mobile approprié. En outre, l'utilisation de proximités sémantiques permet, dans une certaine mesure, de fournir des explications sur les associations proposées, par transformation des assertions de la base de connaissances en phrases exprimées en langage naturel. Ces proximités peuvent être exploitées dans d'autres cas que nous étudierons, comme la génération automatique d'activités d'auto-évaluation de différentes natures (QCM, classements....) ou fournir une aide à la navigation dans un corpus d'annotations.

L'utilisation de proximités sémantiques doit nous permettre d'offrir des parcours alternatifs et de nouveaux modes d'interaction au visiteur. Les objets sémantiquement décrits (œuvres, artistes...) sont complexes, tant au niveau des objets eux-mêmes que de leur description sémantique. Afin de réduire en partie la complexité et d'offrir des interactions plus directement adaptées à l'utilisateur, nous introduisons le concept de proximité contextuelle. Ces proximités contextuelles permettent de répondre à des questions telles que « *Dans quelle mesure le Caravage est-il présent dans le contexte historique du visiteur ?* ». Nous cherchons à travers l'emploi de proximités sémantiques et contextuelles à réduire en partie la complexité des liens entre les objets muséaux pour offrir au visiteur une vue adaptée à sa situation, synthétique et compréhensible des relations entre les artefacts muséaux.

Nous décrivons dans ce chapitre notre mode de calcul de proximités sémantiques entre

instances, par la création d'un vecteur de proximités basé sur les propriétés des instances et l'agrégation de ce vecteur de proximités. L'originalité de notre approche est de prendre en compte différents modes de calcul de proximités sémantiques suivant la nature des propriétés des instances qui sont comparées. Notre méthode de calcul de proximités est une forme d'agrégation des différentes similarités et proximités sémantiques que nous avons décrites au chapitre 2.

L'utilisation de similarités sémantiques dans notre calcul de proximités est justifiée par l'utilisation des vocabulaires contrôlés que nous avons intégrés à notre modèle de connaissance. Ces vocabulaires sont exprimés sous forme de taxonomies dans le modèle des œuvres que nous avons proposé (cf. chapitre 5). Le mode de comparaison entre deux entités d'une taxonomie correspond à un calcul de similarité sémantique. Ainsi, pour déterminer la proximité entre deux œuvres, nous calculerons la proximité sémantique entre les deux artistes (correspondant globalement aux liens existant entre ces artistes) mais nous utiliserons une similarité sémantique pour comparer les thèmes (correspondant globalement à la "distance" entre ces thèmes dans la taxonomie). Après avoir décrit notre mode de calcul des proximités sémantiques, nous décrivons ensuite notre proposition pour le calcul des proximités contextuelles, basée sur le modèle du contexte que nous avons présenté dans la section 5.2.

Ces deux outils, proximités sémantiques et contextuelles, servent de base aux trois propositions d'interactions situées décrites dans la suite : la suggestion d'œuvres, les activités d'auto-évaluation et les activités de communication entre visiteurs par le biais d'annotations contextualisées des œuvres. Ces trois types d'interaction couvrent l'essentiel des interactions actuellement proposées en musée que nous avons décrites dans le chapitre 4.

6.1 Construction d'une mesure de proximité entre instances

L'objectif du calcul des proximités est double. Nous souhaitons d'une part avoir une valeur numérique permettant de quantifier la proximité entre deux objets de la base de connaissances et d'autre part disposer d'une justification compréhensible de cette proximité. Nous devons donc répondre à deux questions : comment quantifier (de manière numérique) la proximité entre deux instances de la base de connaissances ? Comment expliciter de façon compréhensible cette proximité à l'utilisateur, étant entendu que l'explicitation ne doit pas faire intervenir de données numériques, mais des informations qualitatives sur les objets comparés (par ex. l'auteur *auteur1* est proche de l'auteur *auteur2* car leurs styles sont similaires) ?

Nous suivons pour cela l'approche générale des proximités par propriétés proposée par [Pirro & Euzenat, 2010] que nous modifions cependant pour l'adapter à notre situation.

Il s'agit de déterminer la proximité de deux objets par la proximité de leurs propriétés. Plus les objets possèdent de propriétés communes et plus ils peuvent être considérés comme proches. La proximité présentée par [Pirro & Euzenat, 2010] est cependant appliquée aux concepts d'une ontologie. Nous nous intéressons, pour notre part, aux instances de concepts. En effet déterminer la proximité entre le concept de *Peinture* et le concept de *Sculpture* n'aurait que peu d'intérêt pour l'utilisateur. Nous comparons donc plutôt une peinture donnée et une sculpture donnée en cherchant à déterminer ce qu'elles ont de commun ou de différent. Plutôt que de comparer les propriétés des concepts, nous comparons donc les valeurs que prennent les propriétés des instances. Nous complétons également l'approche proposée par Pirro et Euzenat par la prise en compte des liens directs associant deux instances (par exemple le lien spécifiant qu'un artiste a influencé un autre artiste). Ces liens sont des indicateurs forts de proximités qui ne sont pas pris en compte dans la similarité par propriété.

Il s'agit donc, pour disposer d'une mesure de proximité sémantique, de comparer les instances de la base de connaissances suivant les valeurs que prennent leurs propriétés. Plus les valeurs prises par les propriétés sont proches, plus les instances sont proches. La difficulté dans la construction de cette mesure de proximité sémantique réside dans la variété des types de valeurs que peuvent avoir les propriétés. Les propriétés peuvent en effet avoir pour codomaine des instances organisées hiérarchiquement (ex. *calm:hasStyle*), des instances non organisées hiérarchiquement (ex. *calm:hasParent*) ou encore des *datatype* (ex. *calm:hasBirthDate*). Les méthodes de calculs de proximités seront différentes suivant le type du codomaine ; par exemple comparer deux artistes suivant leur style ne se fera pas de la même manière que comparer deux artistes suivant leurs dates de naissances. Notre calcul de proximités doit donc permettre d'intégrer ces informations hétérogènes pour en tirer une valeur numérique. Par ailleurs, ce calcul numérique doit avoir son équivalent sur le plan sémantique pour « expliciter », de manière intelligible à l'utilisateur, que deux objets dont la valeur de proximité est proche de 1 sont effectivement proches.

Pour pouvoir comparer deux instances suivant les valeurs que prennent leurs propriétés, il est évidemment nécessaire que ces instances possèdent des propriétés communes. Ce sera toujours le cas dans nos calculs de proximités, car nous définissons un ensemble de concepts pouvant être comparés et comparerons toujours des instances du même concept. Ainsi nous ne nous intéresserons pas à comparer un tableau du *Caravage* avec la personne *Léonard de Vinci*, mais le *Caravage* avec *De Vinci*. Cette approche est justifiée car les suggestions que nous proposons à l'utilisateur sont de même nature que l'entité qu'il considère. Par exemple, si l'utilisateur consulte la biographie d'un artiste, il lui sera suggéré des artistes « proches » de celui qu'il considère. La justification de cette approche pour les auto-évaluations apparaîtra en 6.3.5.

Les objets que nous comparons sont des instances de l'ontologie CIDOC-CRM modifiée que nous avons présentée au chapitre précédent. Nous présentons dans la suite de cette section la construction d'un tableau permettant de déterminer de manière systématique

les propriétés d'intérêt pour comparer deux instances entre elles. Nous construisons ensuite, à l'aide de ce tableau un vecteur de proximités sémantiques, permettant de mesurer la proximité entre deux instances de la base de connaissances vis-à-vis de chacune de leurs propriétés d'intérêt. Enfin, nous présentons comment, à partir de la base de connaissances, générer une « explication » de la proximité entre deux instances.

6.1.1 Propriétés et concepts d'intérêt

D'un point de vue ontologique, les caractéristiques d'un objet correspondent aux propriétés de l'instance représentant cet objet et aux valeurs que prennent ces propriétés. Cependant, toutes les propriétés ne sont pas intéressantes dans le calcul de similarité sémantique. Certaines propriétés sont en effet utilisées dans l'ontologie mais ne présentent pas d'intérêt pour le visiteur. Ainsi, dans CIDOC-CRM, une œuvre peut être liée à un musée *via* un prêt ou un legs, le fait que deux œuvres aient fait l'objet d'un même prêt n'intéressera *a priori* pas le visiteur et ne devra donc pas être pris en compte dans le calcul.

Afin de déterminer l'ensemble des propriétés présentant un intérêt pour les visiteurs, nous définissons la notion de concept d'intérêt. Un concept d'intérêt est un concept de l'ontologie dont les instances apparaîtront dans les suggestions et les activités d'auto-évaluation et pour lesquelles l'utilisateur pourra demander des informations complémentaires. Ainsi le concept de *Legs* ne fait pas partie de la liste des concepts d'intérêt mais le concept de *Personne* en fait partie. Nous avons sélectionné un ensemble de six concepts d'intérêt dans l'ontologie CIDOC-CRM : personne, style, œuvre, événement historique, lieu et thème. On définit alors une propriété d'intérêt comme une propriété ayant pour domaine et codomaine des concepts d'intérêt.

Le choix de ces concepts d'intérêt nous permet de construire, de manière automatique, un tableau de propriétés d'intérêt (table 6.1). Il s'agit d'un tableau rectangulaire T , dont la première ligne représente les concepts et les *datatype* d'intérêt et dont la première colonne représentent les concepts d'intérêts. L'élément $T[i, j]$ représente l'ensemble des propriétés d'intérêt existant entre le concept $T[i, 0]$ et l'élément $T[0, j]$. Nous notons dans la suite $P_T(A, B)$ l'ensemble des propriétés d'intérêt entre les concept d'intérêts A et B .

Ce tableau nous permet de déterminer comment comparer deux instances de concepts d'intérêt. En lisant la première ligne, on constate par exemple que deux personnes peuvent être comparées suivant leurs parents, les artistes qu'ils ont influencés ou par lesquels ils ont été influencés, leur style, les œuvres qu'ils ont produites, ainsi que leur date de naissance et de mort. Les artistes peuvent également être en lien direct, un artiste pouvant être le maître ou l'élève d'un autre artiste.

Formellement, A et B étant deux concepts d'intérêt et $P \in P_T(A, B)$ une propriété d'intérêt, deux instances i_1 et i_2 du concept A sont proches vis-à-vis de la propriété d'intérêt P si les valeurs $P(i_1)$ et $P(i_2)$ sont proches. On constate que cela nous amène à une

	cidoc:Person	getty:Style	cidoc:Artwork	xsd:date
cidoc:Person	calm:hasMaster calm:hasStudent calm:hasParent	calm:hasStyle	calm:created calm:owns	calm:birthDate calm:deathDate
getty:Style	calm:isStyleOf	skos:broader skos:narrower		calm:period
cidoc:Artwork	calm:createdBy calm:ownedBy		calm:partOf	

TABLE 6.1: Tableau des propriétés d'intérêt pour quelques concepts

définition récursive de la proximité. Nous avons donc besoin d'un calcul initial permettant de comparer $P(i_1)$ et $P(i_2)$. C'est à dire de comparer deux ensembles d'instances ou de valeurs littérales. Nous définissons dans la suite, pour les différents types de valeurs possibles, ces modes de calculs initiaux qui nous amènent à la construction d'un vecteur de proximités.

Par ailleurs, deux instances i_1 et i_2 du concept d'intérêt A sont en lien direct, et donc proches, s'il existe $P \in P_I(A, A)$ telle que $(i_1 P i_2)$ ou bien $(i_2 P i_1)$. C'est par exemple le cas de deux instances de *Person* liées entre elles par la propriété *calm:hasMaster*. Nous définirons pour ce type de liens directs, non capturés par la proximité entre propriétés, un mode de calcul spécifique s'intégrant au vecteur de proximités.

Enfin, deux instances peuvent être proches de manière indirecte, c'est à dire que les valeurs $P(i_1)$ et $P(i_2)$ ne sont pas proches, mais qu'il existe une propriété d'intérêt P_j telle que $P_j(P(i_1))$ et $P_j(P(i_2))$ soient proches. Un exemple concret serait deux artistes dont le maître est différent mais dont le style des maîtres est le même. Nous prendrons également cette proximité indirecte en compte en nous limitant à un seul « pas » pour la proximité indirecte.

6.1.2 Vecteur de proximités entre instances

Le vecteur de proximités entre instances, donne pour un couple d'instances (a, b) du concept d'intérêt X les valeurs de proximités suivant chaque propriété $P_k \in P_I(X, Y), Y$ prenant ses valeurs dans l'ensemble des concepts d'intérêt. La proximité entre deux instances est toujours comprise entre 0 et 1 (c.f. 2.2.1). Chaque composante du vecteur est donc comprise entre 0 et 1, 0 étant la proximité minimale et 1 la proximité maximale.

Le vecteur de proximités est constitué de plusieurs « parties », correspondant aux différentes catégories de valeurs que peuvent prendre les propriétés. Ces types de valeurs sont les instances non hiérarchisées (ex. personnes), les instances hiérarchisées (ex. style, voir 5.1.3) et les littéraux. Nous représentons dans la figure 6.1.1 la forme du vecteur

de proximités, il s'agit du vecteur représentant l'ensemble des propriétés d'intérêt d'un concept à un réagencement des composantes près. Pour deux instances *instance1* et *instance2* d'un même concept *C* qui possède *n* propriétés d'intérêt, le vecteur de proximités entre ces deux instances comporte *n + 1* composantes, les composantes de 1 à *n* correspondent aux proximités vis-à-vis des propriétés d'intérêt du concept *C* et la dernière composante correspond aux liens directs entre les instances *instance1* et *instance2*.

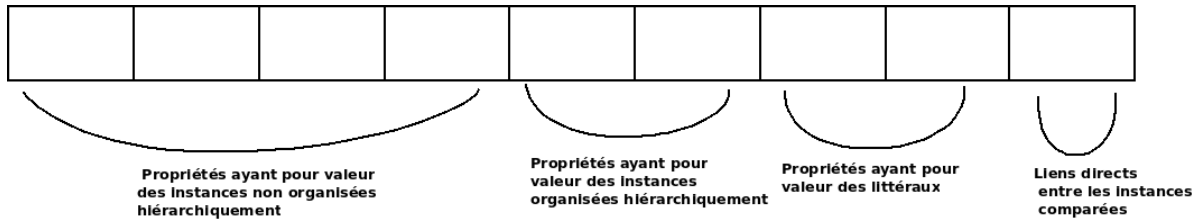


FIGURE 6.1.1: Forme du vecteur de proximités

Afin de calculer les valeurs du vecteur de proximités, nous commençons par proposer un mode de calcul pour les différentes catégories présentées figure 6.1.1. Nous nous intéressons également aux proximités indirectes. Par exemple, si on considère la proximité entre deux tableaux, dans le cas où les artistes sont différents, nous chercherons à déterminer la proximité entre les deux artistes (proximité indirecte). Nous commençons par décrire un vecteur initial de proximités et appliquons ensuite une récursion sur ce vecteur initial, afin de prendre en compte les proximités indirectes.

6.1.2.1 Cas où les valeurs de la propriété sont des instances organisées hiérarchiquement

Dans le cas où les valeurs d'une propriété d'intérêt P_i sont des instances hiérarchisées, par exemple des instances issues d'ICONCLASS ou de Getty-AAT (cf. section 5.1.3), la valeur de la composante i du vecteur de proximités doit dépendre de la similarité des instances dans ces hiérarchies.

Supposons tout d'abord qu'il n'y ait que deux instances à comparer, c'est-à-dire que les instances a et b n'aient chacune qu'une valeur, respectivement a_1 et b_1 pour la propriété P_i . Dans ce cas, nous utilisons la similarité définie par [Wu & Palmer, 1994] (cf. chapitre 2) pour comparer ces deux instances vis-à-vis de la propriété P_i :

$$\begin{aligned} Prox(a, b)_{P_i} &= Sim_{Wu}(a_1, b_1) \\ Prox(a, b)_{P_i} &= \frac{2 \cdot depth(lcs(a_1, b_1))}{depth(a_1) + depth(b_1)} \end{aligned}$$

Cette similarité est en effet bien adaptée au cas des hiérarchies. Elle peut être calculée sans autre information que la structure de la hiérarchie et prend en compte la spécificité des éléments pour la comparaison (c'est-à-dire que deux éléments situés « profondément » dans la hiérarchie seront plus proches que des éléments situés en haut de la hiérarchie, pour le même nombre d'arcs les séparant). De plus cette similarité est normalisée (elle se situe toujours dans $[0, 1]$).

Si a et b ont plusieurs valeurs pour la propriété P_i , nous calculons la proximité moyenne entre les instances de a et de b vis-à-vis de la propriété P_i . Pour chaque concept lié à a via la propriété P_i , nous déterminons le concept le plus proche dans les valeurs de b pour la propriété P_i selon la similarité de Wu et Palmer. Nous effectuons la moyenne de ces différentes similarité. La formule suivante présente le cas général de calcul de proximité dans le cas où P_i prend ses valeurs parmi un ensemble d'instances hiérarchisées.

$$\begin{aligned}
 Prox(a, b)_{P_i} = & \frac{\sum_{a_i \in P_i(a)} Max_{b_j \in P_i(b)}(Sim_{Wu}(a_i, b_j))}{2 \cdot |P_i(a)|} \\
 & + \frac{\sum_{b_i \in P_i(b)} Max_{a_j \in P_i(a)}(Sim_{Wu}(b_i, a_j))}{2 \cdot |P_i(b)|}
 \end{aligned}$$

$P_i(a)$ et $P_i(b)$ étant respectivement l'ensemble des valeurs des propriétés associées à a et b pour la propriété P_i . On note que cette similarité est bien symétrique et comprise entre 0 et 1. Elle croît si les groupes de concepts associés aux instances a et b sont plus proches pour atteindre son maximum si ces groupes sont les mêmes.

6.1.2.2 Cas où les valeurs de la propriété sont des instances non hiérarchisées

Nous décrivons maintenant le calcul pour les propriétés ayant pour valeur des instances de la base de connaissances qui ne font pas partie d'une hiérarchie. C'est par exemple le cas pour la propriété *cidoc:hasStudent* qui lie des instances de types *cidoc:Person* à des instances de type *cidoc:Person*.

Supposons que *personne1* et *personne2* soient deux instances du concept *cidoc:Person* et que l'on cherche à déterminer la proximité entre *personne1* et *personne2* vis-à-vis d'une telle propriété d'intérêt P_i . Dans ce cas, plus le nombre d'instances communes à *personne1* et *personne2* vis-à-vis de la propriété P_i est important, plus la proximité sera élevée. Dans le cas de la propriété *cidoc:participatedIn*, par exemple, cela revient à dire que deux personnes sont proches si elles ont toutes deux été impliquées dans les mêmes événements.

Nous utilisons donc l'indice de Jaccard entre les valeurs des propriétés pour le calcul de la proximité. Cet indice est en effet bien adapté pour déterminer la proximité entre deux ensembles, l'ensemble des valeurs de l'instance a pour la propriété P_i et l'ensemble des valeurs de l'instance b pour la propriété P_i . Formellement, si a et b sont deux instances qui ont pour valeur respectivement $\{a_1, \dots, a_n\}$ et $\{b_1 \dots b_m\}$ pour la propriété P_i , la valeur de la composante i du vecteur de proximités est :

$$\begin{aligned} Prox(a, b)_{P_i} &= \frac{|P_i(a) \cap P_i(b)|}{|P_i(a) \cup P_i(b)|} \\ &= \frac{|\{a_1 \dots a_n\} \cap \{b_1 \dots b_m\}|}{|\{a_1 \dots a_n\} \cup \{b_1 \dots b_m\}|} \end{aligned}$$

On constate aisément que la proximité entre deux instances est symétrique et toujours comprise entre 0 et 1, avec un maximum à 1 lorsque les ensembles comparés sont les mêmes.

6.1.2.3 Cas où les valeurs de la propriété sont de type littéral

Une propriété peut également prendre comme valeur des littéraux. Ce cas est délicat à analyser en matière de proximité, car il est nécessaire de disposer d'une fonction de comparaison entre littéraux, une telle fonction ne pouvant pas, par nature, être générique.

Cependant, dans notre modèle, le seul cas possible de comparaison de deux littéraux est le cas où ces littéraux sont de type *xsd:date*, c'est à dire que les littéraux sont des dates. Ces dates correspondent aux dates de création des œuvres, de naissance et de mort des artistes et à des événements. Nous avons défini, pour ces différents cas, une fonction $Prox_{lit}(d_1, d_2)$. Par exemple, pour la comparaison de la date de naissance des artistes, la fonction donnant la valeur de la proximité dépend de la durée, exprimée en années, $d = |d_1 - d_2|$. On a :

$$Prox_{lit}(d_1, d_2) = \frac{100 - d}{100} \text{ si } 100 - d > 0$$

$$Prox_{lit}(d_1, d_2) = 0 \text{ sinon}$$

L'idée étant que si deux artistes ont des dates de naissance séparées de plus de 100 ans, la proximité vis-à-vis de la date de naissance est nulle. Cette fonction de comparaison est présentée à titre illustratif et nous aurons besoin d'experts en musée pour fixer de manière définitive les comparaisons entre littéraux.

6.1.2.4 Cas des liens directs entre instances

Il nous reste à étudier le dernier cas, celui des relations directes entre instances. Si les instances comparées sont a et b et sont des instances du concept d'intérêt C , ces liens directs prennent la forme de triplets du type $(a P_i b)$ ou bien $(b P_i a)$, où P_i est une propriété d'intérêt appartenant à $P_I(C, C)$. Dans ce cas, en s'inspirant de la méthode de proximité proposée par [Aimé *et al.*, 2011], nous définissons la proximité entre a et b , instances de C , telles que $a \neq b$, comme le rapport entre le nombre de propriétés d'intérêt existant dans la base de connaissance entre a et b et l'ensemble des propriétés d'intérêt liant C à lui-même ($P_I(C, C)$).

$$Prox_{Link}(a, b)_C = \frac{|(a P_i b) \cup (a P_j b)|_{P_i, P_j \in P_I(C, C)}}{|P_I(C, C)|}$$

$P_I(C, C)$ étant l'ensemble des propriétés existantes entre instances de C . On notera que cette expression est nécessairement inférieure ou égale à 1. Par ailleurs, afin d'obtenir une similarité maximale lors de la comparaison d'instances identiques, nous posons quelque soit a instance d'un concept d'intérêt D de la base de connaissance :

$$Prox_{Link}(a, a)_D = 1$$

6.1.2.5 Agrégation du vecteur de proximités

Nous avons vu précédemment comment obtenir un vecteur de proximités numérique entre deux instances a et b . Ce vecteur seul ne nous permet cependant pas de répondre à la question : qui de b ou de c est le plus proche de a . Pour répondre à cette question, nous devons agréger ce vecteur en une valeur unique qui permet effectivement de classer les instances par leur proximités vis-à-vis d'une instance donnée. Par ailleurs, nous devons disposer d'une valeur d'agrégation afin d'appliquer une récursion au calcul du vecteur de proximités qui nous permettra de prendre en compte les proximités indirectes (section suivante).

L'agrégation d'un ensemble de valeur est une opération mathématique par laquelle un n-uplet de valeurs est associé à une valeur unique. On représente de manière générale l'opération d'agrégation de la manière suivante, où on supposera que tous les x_i sont compris entre 0 et 1 :

$$m = Agreg(x_1, x_2, \dots, x_n)$$

L'opérateur d'agrégation doit vérifier certaines propriétés fondamentales afin d'être effectivement un opérateur d'agrégation. Les principales propriétés sont les comportements

aux bornes c'est-à-dire $Agreg(0, \dots, 0) = 0$ et $Agreg(1, \dots, 1) = 1$, ainsi que la stricte monotonie par rapport à toutes les variables c'est à dire :

$$x_k \leq y_k \rightarrow Agreg(x_1, \dots, x_k, \dots, x_n) \leq Agreg(x_1, \dots, y_k, \dots, x_n)$$

Pour la compréhension de ce chapitre, nous considérerons dans la suite que la fonction d'agrégation est la moyenne arithmétique des composantes du vecteur de proximités. Cette fonction d'agrégation respecte bien les propriétés décrites, la moyenne étant effectivement un opérateur d'agrégation. Nous présentons dans le chapitre 7 des moyens de construction et de contrôle plus fin pour l'agrégation du vecteur de similarité.

6.1.2.6 Utilisation de la récursion pour la forme définitive du vecteur numérique de proximités

Nous disposons à présent des éléments suffisants pour construire un premier vecteur de proximités entre deux instances a et b , $Prox_{num}(a, b)$ et déterminer une valeur numérique pour ce vecteur de proximités *via* son agrégation. Cependant, ce vecteur n'est pas complètement satisfaisant car il ne prend pas en compte les proximités indirectes. Ceci tient au mode de calcul de la première partie du vecteur, qui correspond au cas où les propriétés ont pour valeur des instances. Par exemple, pour deux tableaux d'artistes différents, nous obtenons, avec le calcul actuel, une proximité de 0 pour la propriété *calm:hasArtist* dans le cas où les artistes sont différents et une proximité de 1 dans le cas où les artistes sont les mêmes. Il apparaît clairement qu'une méthode plus fine serait souhaitable, donnant une valeur entre 0 et 1 suivant la proximité entre les deux artistes. Nous proposons d'employer une forme de récursion afin de résoudre ce problème.

Cette récursion s'applique aux propriétés ayant pour valeur des instances (première partie du vecteur de la figure 6.1.1). Afin de conserver la cohérence du calcul, nous souhaitons que la proximité pour une propriété P_i entre deux instances ayant les mêmes valeurs pour cette propriété soit de 1.

Le principe du calcul, si l'on souhaite intégrer la proximité indirecte entre deux artistes dans le calcul de la proximité entre deux tableaux par exemple, est de calculer le vecteur de proximités entre les deux artistes et d'intégrer ce résultat à la composante *calm:hasArtist* du vecteur de proximités entre les deux tableaux. Cependant, le résultat de ce calcul est un vecteur et non une valeur. Nous utilisons la fonction d'agrégation définie précédemment et notons alors $\overline{Prox_{num}(a, b)}$ la moyenne arithmétique des composantes du vecteur.

Supposons que nous devons comparer deux instances a et b de la base de connaissances. La première a pour valeur x et y pour la propriété P_i la seconde a pour valeur x et z pour cette même propriété. Nous effectuons donc le calcul du vecteur initial de proximités $Prox_{num}$ pour les couples (x, x) , (x, z) , (y, x) et (y, z) , c'est-à-dire que nous comparons

systématiquement les valeurs prises par les deux vecteurs. La valeur de la composante P_i du vecteur numérique de proximités indirecte est alors :

$$Prox_{ind}(a, b)_{P_i} = \frac{Max(\overline{Prox_{num}(x, x)}, \overline{Prox_{num}(x, y)})}{2} + \frac{Max(\overline{Prox_{num}(y, x)}, \overline{Prox_{num}(y, z)})}{2}$$

On a donc toujours $Prox_{ind}(a, b)_{P_i} \leq 1$ les deux termes de la somme étant, par construction, inférieurs à $\frac{1}{2}$. L'utilisation de la fonction Max pour le calcul des proximités indirectes garantit que, dans le cas où les instances a et b ont les mêmes valeurs pour la propriété P_i , la valeur de la proximité indirecte est bien de 1.

La valeur finale de la proximité entre les instances a et b pour la propriété prend alors une forme relativement simple, il s'agit de la somme pondérée des proximités directes et des proximités indirectes :

$$Prox(a, b)_{P_i} = \alpha \cdot Prox_{num}(a, b) + \beta \cdot Prox_{ind}(a, b)$$

Avec $\alpha + \beta = 1$. Les paramètres α et β quantifient l'importance des proximités directes et indirectes dans la proximité finale. Une valeur importante pour α va « favoriser » les proximités directes tandis qu'une valeur importante pour β mettra en avant les proximités indirectes.

La forme générale de la proximité entre a et b ayant pour valeur de propriétés respectives $P_i(a) = \{x_1, \dots, x_n\}$ et $P_i(b) = \{y_1, \dots, y_k\}$ se déduit aisément de la formule précédente.

$$Prox(a, b)_{P_i} = \alpha \cdot \frac{|P_i(a) \cap P_i(b)|}{|P_i(a) \cup P_i(b)|} + \beta \cdot \frac{\sum_{x_i \in P_i(a)} Max_{y_j \in P_i(b)}(\overline{Prox_{num}(x_i, y_j)})}{|P_i(a)| + |P_i(b)|} + \beta \cdot \frac{\sum_{y_i \in P_i(b)} Max_{x_j \in P_i(a)}(\overline{Prox_{num}(x_i, y_j)})}{|P_i(a)| + |P_i(b)|}$$

Cette formule atteint son maximum lorsque $P_i(a) = P_i(b)$ et décroît lorsque $P_i(a)$ ou $P_i(b)$ possèdent des instances distinctes, la décroissance étant cependant moindre si ces instances distinctes sont proches. La proximité finale entre les instances a et b est alors la moyenne arithmétique des différentes proximités associées aux différentes propriétés.

6.1.3 Explicitation des proximités

Nous avons décrit jusqu'à présent comment calculer la proximité entre deux instances de la base de connaissances. Cependant, le résultat obtenu est uniquement numérique. Nous souhaitons maintenant déterminer les explicitations que nous pourrions présenter à l'utilisateur lorsque le système lui suggérera de s'intéresser à un auteur ou une peinture proche de celui ou celle qu'il considère. Ces explicitations sont construites à partir des assertions de la base de connaissances, qui seront transformées en phrases en langage naturel à l'aide d'un générateur de langage naturel.

Le déterminisme de l'explicitation de la proximité entre deux instances est semblable au calcul numérique des proximités. Pour déterminer les justifications associées à la comparaison des instances a et b , nous déterminons, par propriété, un ensemble d'assertions de la base de connaissances explicitant la proximité entre a et b pour cette propriété.

De la même façon que pour le vecteur de proximités numérique, trois cas de figure, suivant la nature de la valeur de la propriété, peuvent se présenter. La valeur d'une propriété peut être une instance ou un ensemble d'instances de l'ontologie (par exemple pour la propriété *calm:influencedBy*). Cette valeur peut être une instance appartenant à une des deux hiérarchies que nous avons intégrées à notre ontologie (par exemple la propriété *calm:hasStyle*). Enfin, cette valeur peut être un littéral (par exemple pour la propriété *calm:hasBirthDate*). Le cas de figure où la propriété est un lien entre les deux instances a et b sera précisé dans la suite. Nous détaillons dans la suite les explicitations liées à ces différents cas de figure. De la même manière que nous avons employé la récursion pour déterminer l'impact des proximités indirectes dans le calcul du vecteur de proximités, nous emploierons également une forme de récursion pour déterminer les justifications indirectes des similarités (ex. un tableau peut être suggéré, car son auteur a été le maître de l'auteur du tableau que l'on regarde).

6.1.3.1 Cas où les valeurs de la propriété sont de type instance non hiérarchisée

Pour deux instances a et b du même concept ayant toute deux une propriété d'intérêt P_i , si les valeurs de la propriété P_i sont des instances de la base de connaissances, la proximité dépend des valeurs communes de la propriété P_i pour a et pour b . Par exemple, si les entités à comparer sont des artistes, ces deux artistes seront proches vis-à-vis de la propriété *calm:influencedBy* si les artistes ont des influences communes.

Les assertions correspondantes pour l'explicitation de la similarité sont de la forme ($a P_i c$ et $b P_i c$). Dans l'exemple de la figure 6.1.2 ces assertions correspondent aux justifications en langage naturel : "*Léonard de Vinci et Raphaël sont tous deux catholiques romains et ont été influencés par Lorenzo de Medici*". On notera sur la figure que les instances correspondant à *Andrea del Verrocchio* et *Melozzo da Forli* ne contribuent pas aux justifications. Leur contribution apparaîtra au moment du calcul de récursion (cf. 6.1.3.5).

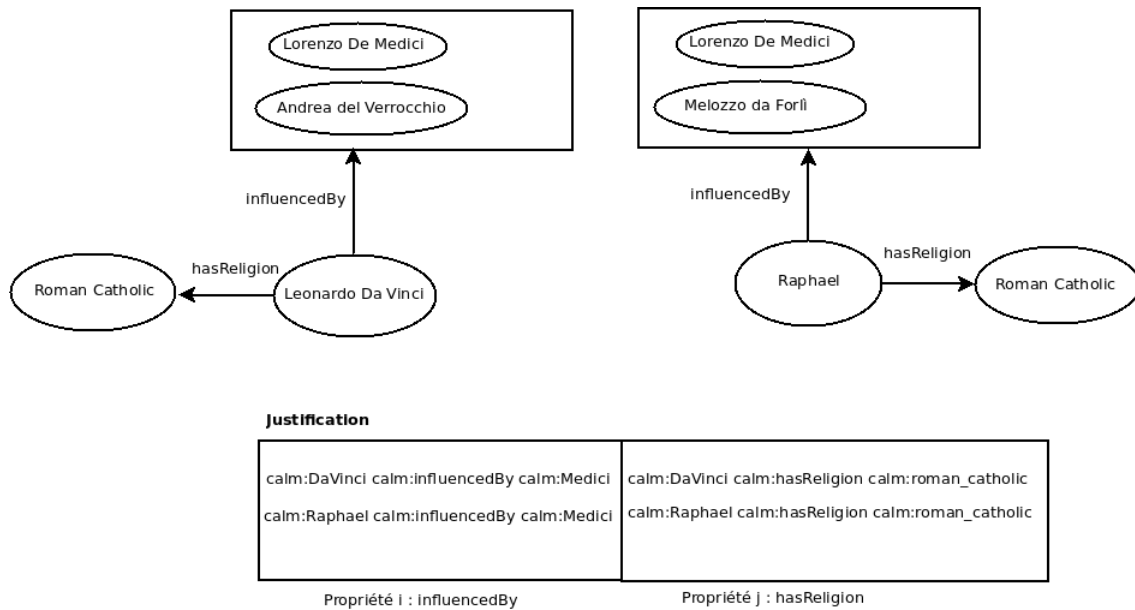


FIGURE 6.1.2: Comparaison entre instances pour la génération des justifications

6.1.3.2 Cas où les valeurs de la propriété sont de type instance hiérarchisée

Le deuxième cas de figure possible pour la valeur des instances de la base de connaissance est d'appartenir à une hiérarchie d'instances exprimée dans l'ontologie. Ceci est par exemple le cas pour le thème d'une œuvre (appartenant à la hiérarchie ICONCLASS). Supposons que P_i soit une propriété dont les valeurs appartiennent à une hiérarchie d'instances. Si a et b sont des instances de la base de connaissances, ces instances sont proches vis-à-vis de P_i si les valeurs des propriétés sont proches, c'est-à-dire si les instances de la hiérarchie sont proches.

Supposons tout d'abord que a et b n'aient chacun qu'une valeur respectivement, a_1 et b_1 pour la propriété P_i , on a donc les triplets $(a P_i a_1)$ et $(b P_i b_1)$. Dans ce cas, les assertions correspondant au calcul numérique de la proximité sont $(a P_i e)$ et $(b P_i e)$, e étant la *least common subsumer (lcs)* de a_1 et b_1 (figure 6.1.3). Ces assertions correspondent dans l'exemple à la justification en langage naturel : "La rencontre de Jacob et Esaü et Paysage boisé avec Abraham et Isaac sont toutes deux des œuvres ayant pour thème l'Ancien Testament et en particulier les patriarches".

Étudions le cas où la propriété P_i prend plusieurs valeurs dans la hiérarchie d'instances pour a et/ou b , ce qui sera souvent le cas pour les thèmes, un tableau ayant généralement plusieurs thèmes. Supposons, pour commencer, que a prend l'unique valeur a_1 pour P_i et b prend les valeurs $\{b_1, b_2, \dots, b_n\}$. Il est clair qu'il s'agit alors de trouver le meilleur *matching* possible entre a_1 et une des valeurs de P_i pour b . Pour cela, nous nous basons là encore sur le calcul de similarité correspondant, c'est-à-dire dans ce cas sur la similarité

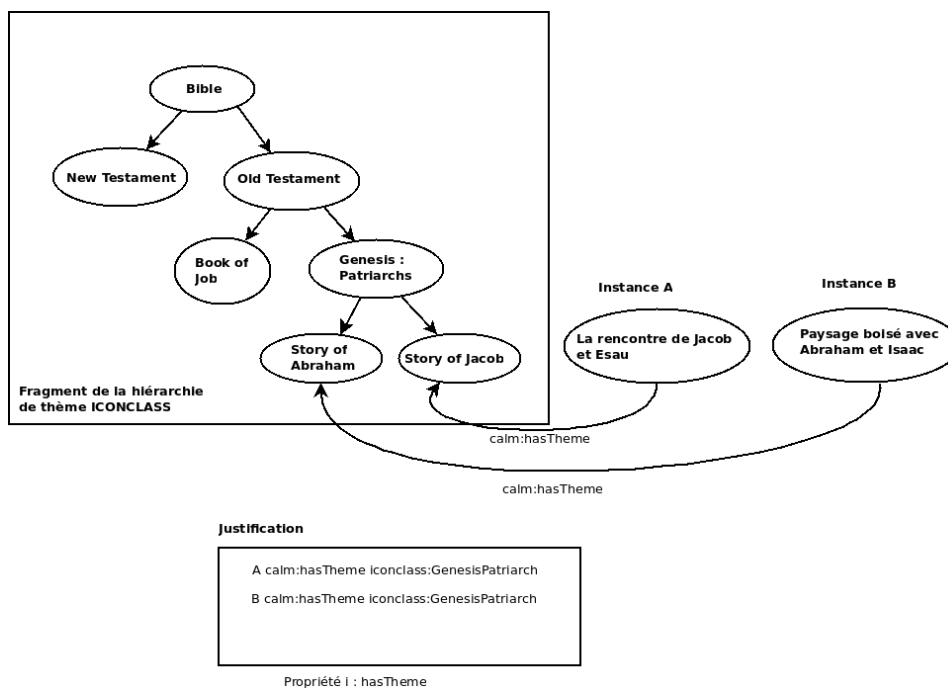


FIGURE 6.1.3: Comparaison entre concepts pour la génération des justifications

proposée par Wu et Palmer. Nous choisissons donc l'instance b_k qui maximise la similarité avec a_1 . Les assertions retenues pour la justification de la proximité sont dans ce cas les mêmes que si b avait comme valeur de propriété uniquement l'instance b_k , c'est-à-dire $(a P_i lcs(a_1, b_k))$ et $(b P_i lcs(a_1, b_k))$.

Supposons maintenant que a prenne deux valeurs $\{a_1, a_2\}$ pour la propriété P_i puis généralisons le calcul. Notons b_i et b_j les instances les plus proches respectivement de a_1 et a_2 nous arrivons à deux doublets d'assertions : $(a P_i lcs(a_1, b_i))$ et $(b P_i lcs(a_1, b_i))$ pour le premier doublet et $(a P_i lcs(a_2, b_j))$ et $(b P_i lcs(a_2, b_j))$ pour le deuxième doublet. **Deux cas de figure sont alors possibles** suivant les valeurs de $lcs_1 = lcs(a_1, b_i)$ et $lcs_2 = lcs(a_2, b_j)$.

Le premier cas de figure est qu'une des instances est en relation hiérarchique SKOS avec l'autre, c'est-à-dire qu'il existe un triplet du type $(lcs_1 skos:broader lcs_2)$ ou bien $(lcs_2 skos:broader lcs_1)$ dans ce cas, l'information portée par les deux doublets d'assertions est redondante. C'est par exemple le cas si lcs_1 correspond au concept *Nativité* et lcs_2 correspond au concept *Nouveau Testament*. En effet, si deux tableaux réfèrent à la Nativité (dans le sens classique de la naissance de Jésus-Christ), ils réfèrent également directement au Nouveau Testament. Dans ce cas, nous ne conservons que le doublet d'assertions le plus spécifique, c'est-à-dire celui pour lequel le lcs est le plus profond dans la hiérarchie.

Le deuxième cas de figure est qu'il n'existe pas de relation hiérarchique (directe ou

indirecte) entre les deux instances. Dans ce cas, les informations portées par les deux doublets ne sont pas redondantes et on conserve ces deux doublets d'assertions pour construire une justification.

La généralisation au cas où l'instance a prend un nombre quelconque de valeurs pour la propriété P_i est immédiate. Il s'agit de déterminer, pour chaque valeur de la propriété P_i de a , la valeur la plus proche parmi les valeurs de la propriété P_i de b , puis d'éliminer les éléments redondants en ne conservant que les éléments les plus spécifiques.

6.1.3.3 Cas où les valeurs de la propriété sont de type littéral

Le dernier cas possible pour le type des valeurs du codomaine des propriétés est le type littéral (correspondant aux propriétés de type *DataTypeProperty*). Il s'agit par exemple des propriétés *calm:hasBirthDate* ou *calm:hasDeathDate* qui prennent pour valeurs des littéraux du type *xsd:date*.

Dans ce cas, nous proposons une justification en fonction de la différence de valeur entre les deux littéraux. Si la différence entre les deux littéraux est inférieure à 10 ans, la justification est « Les deux artistes ont vécu dans la même période ». Si la différence entre les deux littéraux est entre 10 et 50 ans, la justification est « Les deux artistes ont vécu dans des périodes proches ». Si la différence est comprise entre 50 et 100 ans, la justification est : « Les deux artistes ont vécu dans des périodes assez proches ». Comme dans la section 6.1.2.3, nous avons besoin d'un expert pour disposer d'une expression adéquate de la proximité.

6.1.3.4 Cas des liens directs entre les instances à comparer

Enfin, il peut exister des propriétés qui lient directement les instances a et b . Dans ce cas, les assertions correspondantes sont de la forme $(a P_i b)$ ou $(b P_i a)$. Si a et b sont de type *cidoc:Person*, il peut s'agir par exemple des liens *calm:hasMaster*, *calm:hasStudent* ou encore *calm:influencedBy*.

Ces liens directs ne sont pas capturés dans les calculs précédents, car dans les cas où $(a P_i b)$ ou bien $(b P_i a)$, il n'y a pas d'instance commune à a et b pour la propriété P_i . Ces liens sont cependant directement utilisables pour justifier la proximité entre deux objets d'intérêt. Les assertions correspondantes étant naturellement $(a P_i b)$ et/ou $(b P_i a)$.

6.1.3.5 Utilisation de la récursion pour la forme définitive de l'explicitation des proximités

Nous avons à présent construit la forme initiale de la justification de la proximité entre deux instances. Notons :

$$Explicitation_{Init}(a, b)_C = \{x_1, x_2, \dots, x_n\}$$

La représentation de la proximité sémantique entre deux instances a et b du même concept C , où les $x_{i, i < n}$ correspondent aux collections d'assertions obtenues pour la propriété P_i et x_n à la collection d'assertions correspondant aux liens directs entre les instances.

Comme précisé précédemment, cette construction initiale fait l'objet d'une récursion pour obtenir la forme définitive de la justification des proximités. Cette récursion a pour objectif de traiter le cas des proximités indirectes que nous avons déjà présenté. Ce cas est illustré en figure 6.1.4. Les instances qui représentent les tableaux *Nature morte aux pommes et aux oranges* de Cézanne et *Bal du Moulin de la Galette* de Renoir n'ont évidemment pas la même valeur pour la propriété *calm:hasArtist*. Cependant, Cézanne et Renoir sont tous deux membres du mouvement impressionniste. On a donc indirectement la proximité : les deux tableaux ont été peints par des membres du mouvement impressionniste.

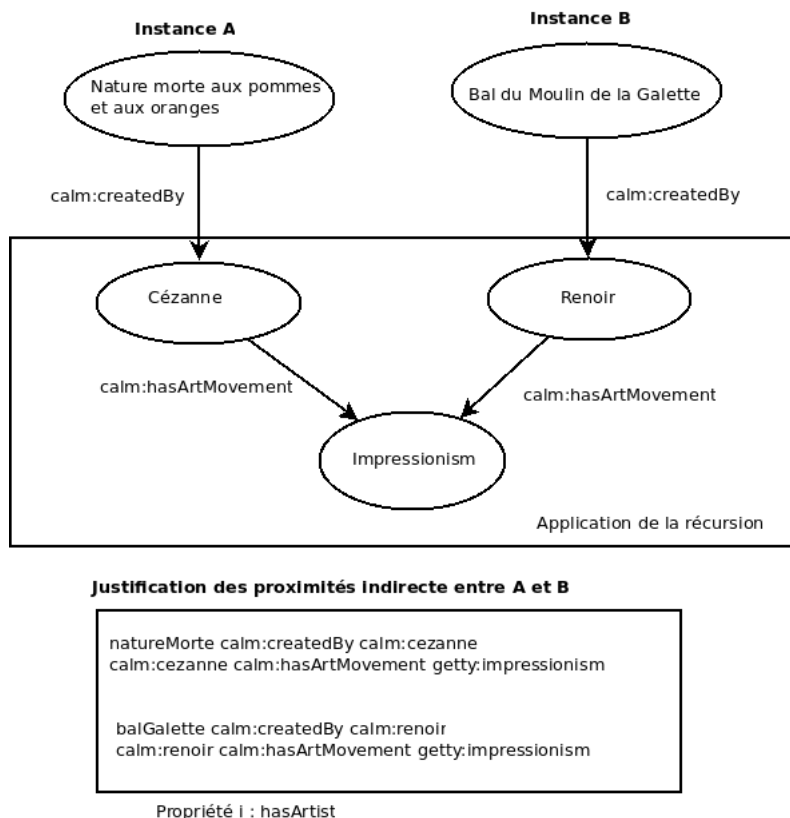


FIGURE 6.1.4: Proximités indirecte entre deux instances

Pour déterminer les justifications associées à ce type de proximités indirectes, une sim-

ple récursion sur les valeurs distinctes des propriétés suffit. Commençons par le cas où l'instance a a pour valeur c pour la propriété P_i et l'instance b a pour valeur d pour cette même propriété, avec $c \neq d$.

En reprenant le principe de la récursion pour le calcul du vecteur numérique de proximités, nous déterminons l'ensemble $Explicitation_{Init}(c, d)_{Codom(P_i)}$. Cette comparaison nous donne l'ensemble des justifications des similarités entre c et d . Les assertions à ajouter au vecteur de similarité entre a et b pour la propriété P_i sont alors des ensembles de deux triplets de la forme $((a P_i c) \text{ et } (c P_j e))$ et $((b P_i d) \text{ et } (d P_j e))$. Avec e une valeur commune de type instance ou une valeur commune de type concept pour la propriété P_j .

Dans le cas général, l'instance a a pour valeur $P_i(a) = \{a_1 \dots a_n\}$ pour la propriété P_i et b a pour valeur $P_i(b) = \{b_1 \dots b_k\}$ pour cette même propriété. Nous effectuons de manière systématique la comparaison entre chaque instance appartenant à $P_i(a)$ et chaque instance appartenant à $P_i(b)$ comme détaillé précédemment, et ajoutons les doublets d'assertions, justifiant la proximité.

La récursion nous permet donc de représenter l'ensemble des relations prises en compte dans le vecteur de proximités numérique. Nous notons :

$$Explicitation_{finale}(a, b) = \{y_1, \dots, y_n\}$$

la forme finale de l'ensemble des justifications de la proximité entre une instance a et une instance b , chaque $y_{i,i < n}$ correspondant à l'ensemble des assertions de la base de connaissances utilisées pour justifier la proximité entre a et b pour la propriété P_i et y_n correspondant aux liens directs entre instances.

On peut noter que la récursion que nous employons ici s'arrête à l'ordre 1. Il serait possible de définir une récursion d'ordre supérieur (ce qui correspondrait par exemple à comparer les propriétés des propriétés des propriétés des instances). Cependant, on constate empiriquement que l'ordre supérieur est difficile à interpréter. Nous nous limitons donc volontairement à un pas de récursion.

6.2 Construction de mesures de proximités contextuelles

Nous avons présenté dans la section précédente notre mode de calcul pour les proximités sémantiques. Ces proximités nous permettent de suggérer à l'utilisateur des entités en liens avec celles qu'il consulte tout en fournissant des explicitations précises à ces suggestions. Cependant, les proximités sémantiques sont insuffisantes pour capturer certaines relations entre un utilisateur et une œuvre, comme la simple relation de proximité physique.

Nous introduisons dans cette section des mesures de proximités contextuelles sur les trois espaces contextuels que nous avons définis : contexte de localisation, contexte historique et contexte d'intérêt. Ces mesures doivent nous permettre de donner un indice numérique quantifiant la proximité d'un ensemble d'instances, ou d'un ensemble d'assertions (ex. les justifications) avec les contextes de localisation, historique et d'intérêt de l'utilisateur. L'utilisation des mesures de proximités sémantiques et contextuelles est à la base des interactions contextualisées que nous détaillons dans la section suivante.

6.2.1 Cas du contexte de localisation et du contexte historique

Nous avons présenté dans le chapitre précédent le mode de construction du contexte de localisation et du contexte historique. Ces espaces contextuels sont essentiellement des structures permettant de représenter les connaissances muséales « autour » de l'utilisateur dans le cas du contexte de localisation et la façon dont l'utilisateur a interagi avec les connaissances muséales dans le cas du contexte historique.

Les calculs de proximités contextuelles sont très similaires dans le cas du contexte historique et du contexte de localisation. Nous détaillons le calcul pour le contexte de localisation. Deux cas sont envisageables : dans le premier cas, il s'agit de mesurer la proximité entre un ensemble d'instances et le contexte de localisation, par exemple pour déterminer l'adéquation d'un questionnaire à choix multiples au contexte de localisation. Dans le deuxième cas, il s'agit de déterminer la proximité entre un ensemble d'assertions (c'est-à-dire de triplets) et le contexte de localisation. Ce sera le cas par exemple lorsque l'on voudra évaluer la proximité entre un ensemble d'explicitations pour une proposition de suggestion et le contexte de localisation.

Afin de déterminer la proximité vis-à-vis du contexte de localisation, nous commençons par nous ramener pour ces deux cas à un cas unique : le cas d'un groupe d'instances. Dans le cas de l'ensemble d'assertions, nous considérons donc uniquement les instances composant ce groupe d'assertions.

Nous nous retrouvons donc avec le problème suivant : comment déterminer la proximité entre un groupe d'instances $E = \{i_1 \dots i_n\}$ et le contexte de localisation de l'utilisateur. Nous construisons pour cela un ensemble $E_L = \{l_1 \dots l_m\}$ de toutes les instances de la base de connaissances muséales présentes dans le contexte de localisation ainsi que les valeurs de leurs propriétés directes (par exemple, si un tableau est présent dans le contexte de localisation nous ajoutons à l'ensemble E_L son style et son auteur...).

Enfin, nous réalisons l'opération de comparaison de l'ensemble E avec l'ensemble E_L . Cette opération nous donne une valeur de proximité qui quantifie la présence d'instances de E dans le contexte de localisation :

$$Prox_{C_L}(E) = \frac{|E \cap E_L|}{|E_L|}$$

Si $Prox_{C_L}(E)$ est proche de 1, cela signifie que l'ensemble E est proche de l'ensemble E_L . Ainsi les instances présentes dans E sont présentes dans l'environnement immédiat de l'utilisateur. Par exemple, les instances associées aux peintres, styles ou thèmes présents dans E seront présentes autour de l'utilisateur. On notera que cette formulation est très proche de l'indice de Jaccard des ensembles E et E_L . Cependant nous ne faisons pas apparaître l'union des deux ensembles au dénominateur car il s'agit de quantifier la présence de E dans E_L et non pas de comparer ces deux ensembles.

De la même manière, nous définissons la proximité de l'ensemble d'instances E vis-à-vis du contexte historique :

$$Prox_{C_H}(E) = \frac{|E \cap E_H|}{|E_H|}$$

E_H étant l'ensemble des instances de la base de connaissance muséale qui appartiennent au contexte historique. On peut noter que le calcul de cette proximité pour le contexte historique n'est pas très précis au niveau temporel, toutes les instances de la base de connaissances muséales, depuis le début de la visite, étant utilisées dans ce calcul. Nous présentons dans le chapitre suivant des modes de contrôles permettant de spécifier une proximité vis-à-vis de certaines périodes particulières du contexte historique.

6.2.2 Cas du contexte d'intérêt

Dans le cas du contexte d'intérêt nous proposons un calcul quelque peu différent. L'idée étant ici que si l'utilisateur possède un intérêt pour un peintre par exemple, il possèdera certainement un intérêt pour ses œuvres, son style... c'est à dire formellement pour les valeurs des propriétés de l'instance représentant le peintre en question.

Nous construisons alors l'ensemble E_I des instances de la base de connaissances muséales présentes dans le contexte et appliquons une relaxation sémantique à cet ensemble. C'est-à-dire que, pour chaque instance de la base de connaissances muséales présente dans l'ensemble E_I , nous ajoutons à l'ensemble E_I les valeurs des propriétés d'intérêt de cette instance. En notant $E_{I_{rel}}$ ce nouvel ensemble, nous déterminons la proximité entre un groupe d'instances E et le contexte d'intérêt de la manière suivante :

$$Prox_{C_I} = \frac{|E_{I_{rel}} \cap E|}{|E_{I_{rel}}|}$$

6.3 Interactions contextualisées

6.3.1 Génération de suggestions

La génération automatique et personnalisée de suggestions est souvent une part importante, voire la seule, de l'assistance à la visite de musée en mobilité. Nous présentons ici notre méthode pour la génération de ces suggestions. L'originalité de cette méthode est triple. En premier lieu, nous ne nous limitons pas aux suggestions d'œuvres mais nous suggérons également des auteurs, styles ou thématiques susceptibles d'intéresser l'utilisateur. Ensuite, nous fournissons des justifications précises accompagnant les suggestions qui permettent d'aider le visiteur à se faire une image des liens existant entre les œuvres, auteurs et styles. Enfin, nous proposons une méthode de prise en compte du contexte de visite, afin d'offrir des suggestions pertinentes sur le plan sémantique et contextuel.

Ces suggestions offrent au visiteur une navigation alternative, par rapport aux audioguides classiques, au sein des connaissances structurées de notre base de connaissances. Ce parcours numérique est constamment en lien avec le parcours physique effectif du visiteur du musée. Les suggestions doivent aider le visiteur à porter un regard différent sur son parcours en explicitant les liens entre les artefacts muséaux. La question qui se pose est donc : comment réduire la complexité des informations présentées à l'utilisateur tout en conservant les informations les plus « utiles » suivant son parcours et ses intérêts ? Cette question en amène immédiatement deux autres : comment choisir les suggestions à offrir au visiteur et comment choisir les justifications qui lui seront présentées suivant la suggestion ? Afin répondre à ces questions, nous utilisons les calculs de proximités sémantiques et contextuelles que nous avons présentés.

6.3.2 Comparaison des instances

Supposons l'utilisateur intéressé par l'instance a du concept C de la base de connaissances. Comme précisé auparavant, a est instance d'un concept d'intérêt dans le sens où nous l'avons défini en 6.1.1. L'utilisateur peut alors consulter un document concernant l'instance a , par exemple la notice s'il s'agit d'une œuvre ou la biographie de l'artiste s'il s'agit d'une personne.

L'idée principale va alors être de suggérer au visiteur d'autres instances du même concept (ex. artistes s'il s'agit d'un artiste, œuvres s'il s'agit d'une œuvre) en lien avec l'instance qu'il considère. La figure 6.2.1 est un écran de l'application illustrant ces suggestions.

La première étape est de déterminer les instances ayant effectivement un lien sémantique « absolu », c'est à dire indépendamment du contexte, susceptibles d'être suggérées. Pour cela, nous commençons par effectuer le calcul de proximités sémantiques présenté

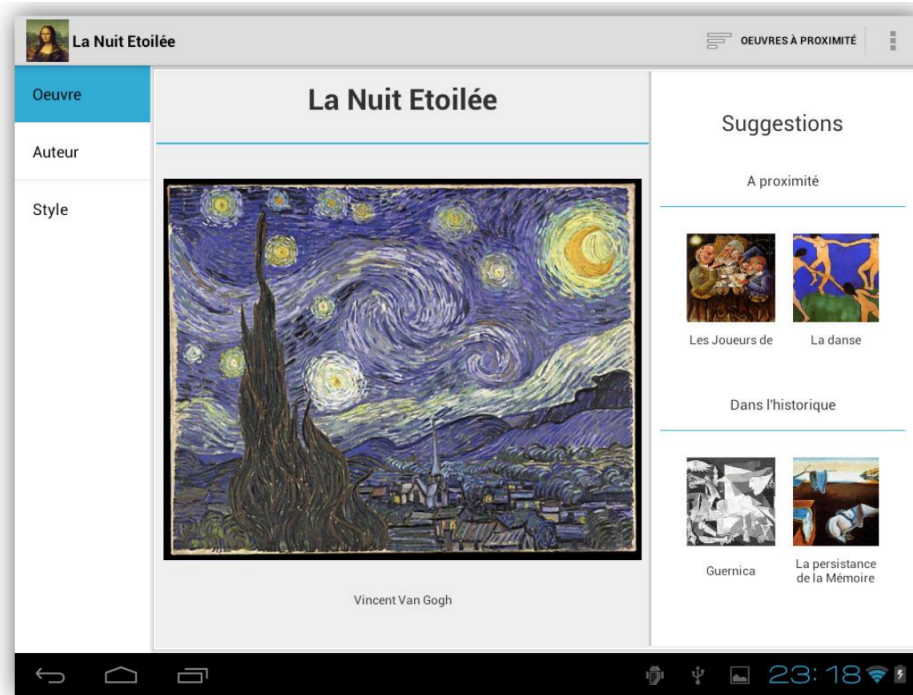


FIGURE 6.3.1: Suggestion d'œuvres lors de la consultation d'une notice

plus haut sur l'ensemble des instances de même type que celle considérée par l'utilisateur. Ce calcul nous permet d'obtenir un ensemble de valeurs correspondant aux valeurs numériques de la proximité, ainsi que l'ensemble associé des assertions de la base de connaissances, explicitant de ces proximités.

$$P_{num}(a) = \{Prox_{num}(a, b_k)\}_{b_k \in C}$$

$$P_{sem}(a) = \{Explicitation_{finale}(a, b_k)\}_{b_k \in C}$$

Avec a l'instance considérée, b_k l'instance de même type susceptible d'être suggérée et C le concept d'intérêt correspondant à a .

Ces deux ensembles pourraient déjà nous permettre de proposer des suggestions aux visiteurs. Les instances que nous proposerions seraient celles pour lesquelles la valeur de proximité (c'est-à-dire $Prox_{num}(a, b_k)$) est la plus proche de 1. Cependant, nous souhaitons intégrer les connaissances sur le contexte de visite dans la suggestion. Ces connaissances contextuelles nous permettent d'intégrer l'historique du parcours et des consultations ainsi que les intérêts du visiteur dans les suggestions, rendant le parcours parmi les connaissances structurées réellement personnalisé.

6.3.3 Calcul des taux d'adéquations au contexte, choix des instances à présenter

Considérons que l'utilisateur consulte un document (notice, bibliographie...) concernant une instance a de la base de connaissances, a étant instance d'un concept C . Nous disposons d'un ensemble d'instances $\{b_k\}_{k \in \mathbb{N}}$ candidates potentielles à la suggestion, c'est-à-dire l'ensemble des instances du même concept C . Considérons un de ces candidats : b_k , alors $Explicitation_{finale}(a, b_k)$ décrit l'ensemble des « explicitations » de la proximité entre a et b_k . Afin de déterminer les taux d'adéquation au contexte de ce candidat, nous construisons l'ensemble $E = \{j_1 \dots j_n\}$ de toutes les instances présentes dans $Explicitation_{finale}(a, b_k)$. Cet ensemble d'instances comporte, par construction de l'ensemble $Explicitation_{finale}(a, b_k)$, b_k ainsi que toutes les valeurs de propriétés communes pour a et b_k .

Le choix des instances à proposer dépend de la proximité sémantique avec l'instance considérée par l'utilisateur ainsi que des scores de proximité vis-à-vis des contextes de localisation, historique et d'intérêt. L'idée étant ici de considérer que plus le score de proximité vis-à-vis des différents contextes est important, plus les liens entre les deux instances sont connus par l'utilisateur. Par exemple, si les deux instances représentent des tableaux proches en terme de style et que l'utilisateur est familier de ce style, il lui sera plus facile de comprendre la suggestion. Pour chaque instance susceptible d'être suggérée, on attribue donc un score de la manière suivante :

$$Score(i_k) = \alpha \cdot Prox_{num}(a, i_k) + \beta \cdot Prox_{C_L}(E) + \gamma \cdot Prox_{C_H}(E) + \delta \cdot Prox_{C_I}(E)$$

α , β , γ et δ étant les pondérations respectives de la similarité numérique et des différents taux d'adéquation au contexte.

6.3.4 Génération des explicitations

Un aspect important pour tirer pleinement parti de la suggestion est de disposer d'une justification claire de cette suggestion, c'est-à-dire d'explicitier au visiteur les proximités entre l'entité qu'il considère et les entités qui lui sont suggérées. C'est cette justification qui permet au visiteur de construire une image mentale des liens existant entre œuvres, artistes, styles... tout au long de sa visite.

Pour générer ces justifications, nous disposons de la forme sémantique de la proximité entre deux instances $Explicitation_{finale}(x, y)$. Par construction, cet ensemble contient, l'ensemble des assertions directes et indirectes entre l'entité que considère l'utilisateur et l'entité à suggérer.

Deux questions se posent alors : comment exprimer ces liens de façon intelligible pour l'utilisateur et comment choisir les liens les plus pertinents pour l'utilisateur ? La première question est technique et implique l'utilisation d'un générateur de langage naturel. La deuxième question est plus fondamentale et implique de choisir parmi un ensemble potentiellement important de liens ceux qui vont le plus « parler » à l'utilisateur. Par exemple, si un des liens entre deux œuvres fait intervenir le concept de *pointillisme*, concept qui a déjà été vu par l'utilisateur, il sera peut-être intéressant de mettre en avant ce lien par rapport aux autres.

Génération des explicitations en langage naturel

Nous décrivons brièvement le processus de génération en langage naturel des justifications. Nous avons décrit précédemment les trois types de justifications que nous utilisons pour expliciter la proximité entre une instance a et une instance b . Les explicitations basées sur un lien direct entre les instances, de la forme $(a P_i b)$, les explicitations basées sur une propriété commune entre a et b (par exemple si a et b sont des œuvres du même peintre) et enfin les explicitations basées sur une propriété commune indirecte entre a et b , par exemple si a et b sont de deux peintres différents, ces peintres ayant toutefois un maître commun.

Le cas de la génération des explicitations pour les liens directs entre instances est trivial, nous disposons de l'assertion $(a P b)$. Afin de générer l'explicitation, les instances a et b possèdent une propriété renvoyant vers un littéral les représentant en langage naturel (ex. pour l'instance représentant Léonard de Vinci, le littéral est « Léonard de Vinci »). De la même manière, chaque propriété est associée à un littéral représentant le verbe décrit par la propriété. Par exemple, la propriété *calm:hasCreated* est associée au verbe « créer ». En utilisant un générateur de langage naturel, $(a P_i b)$ est alors directement transformée en une phrase en utilisant les termes (littéraux) associés a , b et P_i . Par exemple l'assertion $(\text{calm:deVinci } \text{calm:inspiredBy } \text{calm:deMessine})$ est transformé en « Léonard de Vinci a été inspiré par Antonello de Messine ».

Etudions le cas où deux instances ont une valeur de propriété commune. Supposons que l'entité considérée par l'utilisateur soit représentée par l'instance a , que l'entité candidate pour la suggestion soit représentée par l'instance b . Les explicitations directes sont des ensembles de deux triplets de la forme :

$$(a P_i c) \text{ et } (b P_i c)$$

Ce qui signifie que les instances a et b ont au moins la valeur c en commun pour la propriété P_i . De la même manière, en utilisant un générateur de langage naturel, il est aisé de générer une phrase du type « La Dame à l'Hermine et l'Homme de Vitruve ont toutes deux été créés par Léonard de Vinci ».

Considérons à présent le cas des justifications basées sur une propriété commune indirecte. On suppose comme précédemment que a est l'instance représentant l'entité considérée par l'utilisateur et b est l'entité candidate pour la suggestion. L'explicitation indirecte est alors représentée par deux ensembles de deux triplets de la forme :

$$((a P_i c) \text{ et } (c P_j d)) \text{ et } ((b P_i e) \text{ et } (e P_j d))$$

De la même manière que précédemment, par l'utilisation d'un générateur de langage naturel, on arrive aisément à une phrase du type : « Les peintures *Nature morte aux pommes et aux oranges* et *Bal du Moulin de la Galette* ont été créées par des impressionnistes ».

Choix des explicitations à présenter

La question du choix des explicitations à présenter se pose lorsque leur nombre est important. En effet, il serait contre-productif de proposer trop de justifications au visiteur, ce dernier ne prendrait certainement pas le temps de les lire toutes et les explicitations pertinentes dans son contexte seraient noyées dans l'ensemble des autres.

Nous avons choisi de nous limiter au plus à trois justifications. Il s'agit alors de choisir, parmi l'ensemble des explicitations possibles, celles qui sont les plus susceptibles de « parler » à l'utilisateur. Pour cela, nous nous basons sur un calcul de taux d'adéquation au contexte des explicitations. L'hypothèse étant qu'une explicitation faisant intervenir des éléments déjà vus ou connus par l'utilisateur sera mieux comprise et plus "parlante" que des justifications faisant intervenir des entités qu'il ne connaît pas.

Nous avons vu que les explicitations sont exprimées sous forme de triplets ou de couples de triplets dans la forme sémantique de la proximité $Explicitation_{finale}(x, y)$. Pour chaque explicitation, on construit l'ensemble $E_J = \{j_1, \dots, j_n\}$ représentant l'ensemble des instances présentes dans l'explicitation.

Le score d'une explicitation est défini en utilisant le calcul de proximité contextuelle présenté précédemment :

$$Score(E_J) = \alpha \cdot Prox_{C_L}(E) + \beta \cdot Prox_{C_H}(E) + \gamma \cdot Prox_{C_I}(E)$$

α , β , et γ étant les pondérations associées aux différentes proximités telles que $\alpha + \beta + \gamma = 1$. On remarquera que dans le score de la justification E_J , n'apparaît pas la "pertinence" de l'explicitation. Par exemple, savoir que deux artistes ont un maître commun, sera certainement plus pertinent que de savoir que deux artistes ont le même pays d'origine. Comme nous l'avons précisé en 6.1.3, nous traitons dans ce chapitre toutes les propriétés comme ayant une même pertinence, c'est pourquoi la moyenne arithmétique est utilisée pour agréger le vecteur numérique de proximités. Nous précisons dans le chapitre

suivant comment associer également un score de pertinence à l'explicitation qui sera présentée.

6.3.5 Génération d'activités d'auto-évaluation

La deuxième catégorie d'activités contextualisées que nous proposons est l'utilisation de quizz d'auto-évaluation. Ces activités sont présentées de manière ludique, sous forme de mini-jeux que le visiteur est libre d'accomplir ou non. Notre hypothèse est que ce type de mini-jeux peut être un média efficace pour l'acquisition, le renforcement et la validation de nouvelles connaissances.

Plusieurs difficultés apparaissent dans la génération automatique d'activités d'auto-évaluation. D'une part, comment générer ce type d'activité en fonction de ce dont nous disposons déjà (base de connaissances muséales et proximités sémantiques)? D'autre part, comment adapter la difficulté de ces activités en fonction du profil utilisateur? Enfin comment proposer des activités contextuellement pertinentes, par exemple des activités faisant intervenir des connaissances sur des objets (tableaux, artistes..) que l'utilisateur a déjà rencontrés?

6.3.5.1 Activités de type QCM

Plusieurs travaux se sont intéressés à la génération automatique de questions à partir d'ontologies ou de bases de connaissances. Les systèmes OntoQue [Al-Yahya, 2011] et le système proposé par [Papasalouros *et al.*, 2008] sont typiques des approches basées sur les ontologies pour la génération de QCM (Questionnaire à Choix Multiples). Le principe de ces systèmes est d'établir un ensemble de stratégies de génération, de la forme « *Choose individuals which are not members of a given class, provided that they are members of one of its superclasses. More specifically, if $A(a)$ for some a , then correct answer is: $A(a)$. For the distractors selection, we assume that B is a superclass of A . Then, if $B(b)$, $b \neq a$ and b is not an individual of A , then $A(b)$ is a distractor.* » [Papasalouros *et al.*, 2008].

Plusieurs problèmes apparaissent cependant avec ce type d'approches : d'une part, elles exigent une liste exhaustive de stratégies de formulation de questions, qui peut être difficile à construire ; d'autre part, les distracteurs, c'est-à-dire les réponses incorrectes, manquent parfois clairement de cohérence avec la question posée. Enfin, ces méthodes ne permettent ni la gestion de la difficulté, ni la gestion de l'adéquation au contexte de la question posée.

Notre proposition s'appuie sur une stratégie unique de détermination des distracteurs par des calculs de proximités sémantiques, cette stratégie nous permet de reproduire l'essentiel des questions posées par les systèmes présentés précédemment tout en ayant une gestion fine de la difficulté et de l'adéquation au contexte.

Nous groupons dans les activités de types QCM deux types de questionnaires. Les QCM classiques qui comportent une bonne réponse et trois distracteurs (mauvaises réponses), ainsi que les questions de type vrai-faux. Ces deux types d'activité sont très semblables dans leur construction. La figure 6.3.1 représente des copies d'écran de ces activités.

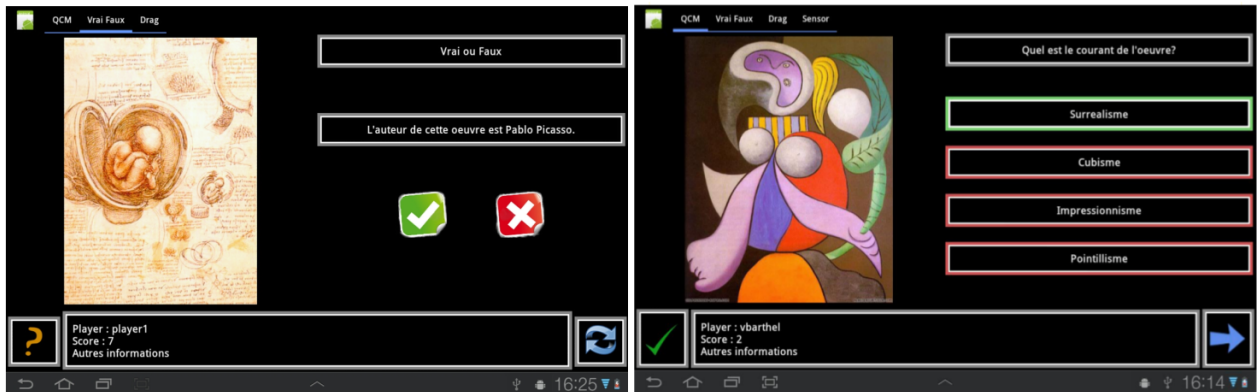


FIGURE 6.3.2: Ecrans de l'application pour les questions de type vrai/faux (gauche) et les QCM (droite)

Nous détaillons en premier lieu la méthode de génération de QCM avant de nous pencher sur les questions de type vrai/faux.

Génération automatique d'activités de type QCM

Un item d'un questionnaire à choix multiple est composé d'une question, d'une réponse correcte et de plusieurs réponses incorrectes (ou distracteurs). Le nombre de distracteurs est souvent fixé à trois, ce qui sera le cas ici. Afin de générer automatiquement des QCM nous utilisons une méthode systématique qui, à partir d'une instance de concept d'intérêt de la base de connaissances, va produire l'ensemble des items de QCM possibles, c'est à dire l'ensemble des triplets (question, bonne réponse, distracteurs) possibles à partir de cette instance.

Nous partons d'une instance a de concept d'intérêt de la base de connaissances. Cette instance a un certain nombre de propriétés d'intérêt P_i . Pour chaque propriété d'intérêt, nous sélectionnons une bonne réponse b , c'est-à-dire une instance telle que le triplet (a, P_i, b) appartienne à la base de connaissances. Nous sélectionnons ensuite l'ensemble des groupes de trois instances appartenant au codomaine de P_i et n'étant pas des bonnes réponses, c'est-à-dire que pour aucune de ces instances d_j le triplet (a, P_i, d_j) n'appartient à la base de connaissances. Cette opération est répétée pour chaque « bonne réponse » de la propriété d'intérêt P_i et pour chaque propriété d'intérêt de l'instance a . La formulation de la question à partir de l'instance a et de la propriété P_i est ensuite triviale en utilisant un générateur de langage naturel.

Dans le cas des questions de type vrai-faux, nous partons également d'une instance a de concept d'intérêt et pour chaque propriété P_i et chaque valeur v_j prise par les propriétés, nous construisons la phrase « vrai » à partir du triplet appartenant donc à la base de connaissances (a, P_i, v_j) . Ce triplet est ensuite reformulé pour former une assertion en langage naturel du type « Léonard de Vinci a peint la Dame à l'Hermine ». Pour générer les assertions fausses, nous déterminons pour chaque propriété P_i de l'instance A l'ensemble des instances f_j appartenant au codomaine de P_i et ne vérifiant pas (a, P_i, f_j) .

Ces algorithmes nous fournissent un nombre très important d'items, c'est-à-dire d'ensemble (questions, bonne réponse, distracteurs) pour les QCM et d'assertions pour les questions de type vrai-faux. En effet, les instances ont souvent plusieurs propriétés et les possibilités combinatoires sont très importantes, puisque la sélection des distracteurs revient à choisir 3 éléments parmi N , N étant le nombre d'éléments ne vérifiant pas l'assertion. L'ordre de grandeur du nombre de questions est donc de l'ordre de $\binom{N}{3}$ avec N pouvant être important suivant la base de connaissances (par exemple N peut être égal au nombre de peintres de la base de connaissances). Nous voyons dans la suite comment sélectionner les questions potentiellement intéressantes pour l'utilisateur parmi cet important corpus de questions, sur la base de la difficulté et du taux d'adéquation au contexte.

Gestion de la difficulté

La difficulté dans un questionnaire à choix multiples tient essentiellement aux distracteurs qui sont proposés. Plus les distracteurs sont crédibles, moins il sera facile de répondre au QCM. Nous avons choisi d'utiliser les proximités sémantique entre instances pour évaluer la crédibilité des distracteurs et donc la difficulté du QCM. Une instance a proche sémantiquement d'une instance b suivant la définition de la proximité que nous avons donnée est une instance possédant de nombreuses propriétés communes avec l'instance b . C'est une instance qui « ressemble » à b suivant plusieurs propriétés. Ainsi, les instances de *Léonard De Vinci* et *Raphaël* sont proches, alors que toutes deux sont éloignées de l'instance *Vincent Van Gogh*. Supposons que la question soit : "Qui a peint la Dame à l'Hermine?". La question étant accompagnée d'une image de l'œuvre, l'utilisateur pourra éliminer plus facilement le distracteur *Vincent Van Gogh* que le distracteur *Raphaël*.

Afin de formaliser cette idée, nous proposons un calcul de la difficulté d'une question basé sur la proximité moyenne des distracteurs par rapport à la réponse correcte. Ainsi, si la bonne réponse est l'instance a et les distracteurs les instances d_1 , d_2 et d_3 , la difficulté de la question est calculée de la manière suivante :

$$\text{diff} = \frac{\text{Prox}_{\text{num}}(a, d_1) + \text{Prox}_{\text{num}}(a, d_2) + \text{Prox}_{\text{num}}(a, d_3)}{3}$$

Il est à noter que d'autres opérateurs d'agrégation que la moyenne peuvent être utilisés, notamment les proximités maximale et minimale entre les distracteurs et la bonne réponse. Cependant ces opérateurs sont moins fins que l'opérateur moyenne : l'opérateur « proximité maximale » ne garantit la difficulté que pour un seul distracteur (il ne prend pas en compte les autres distracteurs) et l'opérateur proximité minimale ne permet pas d'avoir certains distracteurs faciles à éliminer. L'opérateur moyenne semble donc une alternative adéquate pour le calcul de la difficulté.

Dans le cas des questions de type Vrai/Faux, le calcul de la difficulté est différent : il n'est pas possible de déterminer une difficulté basée sur la proximité sémantique si l'assertion est vraie. Si l'assertion est fausse, la difficulté est fonction du caractère vraisemblable de l'assertion, c'est-à-dire de la proximité sémantique entre la réponse incorrecte et la réponse correcte.

Adéquation au contexte

Le dernier point à considérer pour choisir un QCM spécifique pour un utilisateur dans une situation donnée est le taux d'adéquation au contexte de la question et des réponses possibles. Nous reprenons pour ce calcul le principe de calcul de la proximité contextuelle présenté en 6.2.

Considérons Q , un QCM et $I(Q)$ l'ensemble des instances impliquées dans Q (c'est à dire les instances de concepts d'intérêt présentes dans la question, la bonne réponse et les distracteurs).

En reprenant les notations de la partie 6.2, le score de proximité contextuelle du QCM Q est calculé de la manière suivant :

$$Score(Q) = \alpha \cdot Prox_{C_L}(I(Q)) + \beta \cdot Prox_{C_H}(I(Q)) + \gamma \cdot Prox_{C_I}(I(Q))$$

Par le choix des coefficients α , β et γ , il est possible de favoriser des questions impliquant des entités appartenant au contexte d'intérêt, de localisation ou historique. Nous présentons dans le chapitre suivant une alternative plus fine pour déterminer les questions à présenter à l'apprenant en fonction du contexte.

6.3.5.2 Activités de classification

Nous présentons ici les activités de classification alternatives aux activités de type QCM. L'idée est de proposer à l'apprenant de classer un ensemble d'entités (ex. peintres, tableaux) suivant la valeur prise par certaines propriétés. L'activité de classement la plus évidente est le classement des œuvres par rapport à leur date de création, ce qui correspond en fait à un ordonnancement (figure 6.3.2).

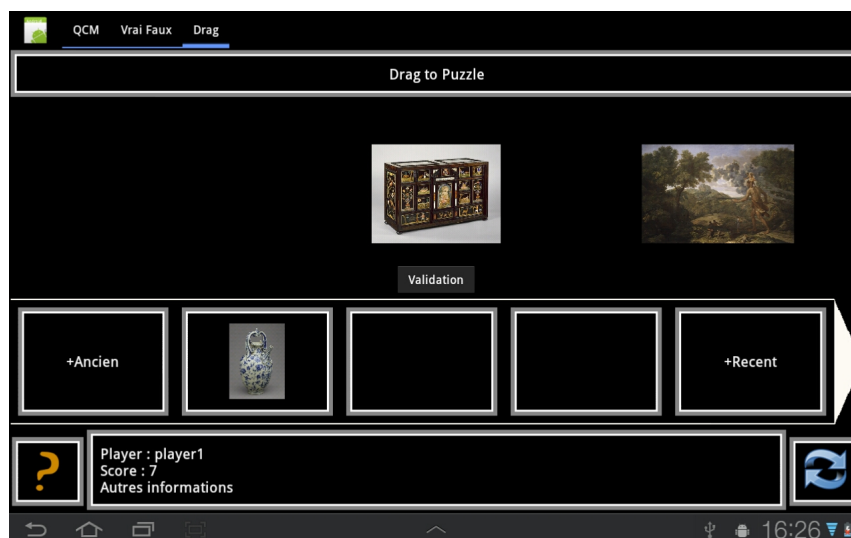


FIGURE 6.3.3: Activité d'ordonnement chronologique des œuvres

Le calcul de la difficulté pour cette activité de classification est très similaire au calcul de la difficulté pour les QCM. L'idée étant de connaître la proximité sémantique « moyenne » des objets entre eux, nous calculons la proximité sémantique entre chaque paire d'objets et prenons la moyenne de ces proximités. Le calcul du taux d'adéquation au contexte est similaire à celui présenté précédemment.

L'autre type d'activité de classification que nous proposons fait appel aux hiérarchies d'instances issues de Getty-AAT et d'ICONCLASS. L'idée est de proposer à l'utilisateur de classer un ensemble d'œuvres ou d'artistes suivant leur style ou leur thème. La difficulté de la question est ici évaluée par la proximité des styles ou thèmes proposés pour la classification. Il est évident qu'il sera plus facile de classer des œuvres parmi les styles "Impressionniste", "Cubiste" et "Surréaliste" que parmi les styles "Baroque", "Haute-Renaissance", "Louis XIV". Pour déterminer la difficulté d'un jeu de classification nous déterminons donc la proximité moyenne entre les catégories de classement proposées, le calcul de proximité entre deux catégories étant celui de [Wu & Palmer, 1994] que nous avons déjà présenté.

6.3.6 Annotations contextualisées

Il est généralement admis, dans le cadre de situations d'apprentissage, que les communications entre apprenants favorisent largement l'acquisition et l'appropriation de nouvelles connaissances. En particulier, dans le cas où les apprenants sont en situation authentique, c'est-à-dire lorsqu'ils ont l'usage de ces connaissances au travers d'activités du monde réel, les échanges favorisent l'appropriation et l'usage à bon escient d'un vocabulaire commun. Dans le cadre d'un apprentissage informel en musée, nous instrumentons

ces communications *via* l'annotation libre d'œuvres d'art. Ce type de communication se révèle en effet particulièrement utile pour permettre aux visiteurs « de décrire, partager et expliciter leur ressenti par rapport aux œuvres » [Stevens & Martell, 2003].

La conception d'un système d'annotation pose notamment deux problèmes : tout d'abord, s'il existe un grand nombre d'annotations, comment permettre au visiteur de retrouver les annotations qui l'intéressent ; ensuite, comme pour toute communication, directe ou différée, le contexte de la communication est important et donne du sens à cette communication, ce qui pose la question des éléments contextuels à intégrer dans l'annotation et de leur mode de visualisation.

Nous présentons un modèle d'annotation contextualisée, prenant en compte la sémantique et le contexte de l'annotation. Afin d'offrir des possibilités de navigation riches dans l'ensemble des annotations, nous définissons également des proximités sémantiques et contextuelles entre annotations permettant de naviguer dans le corpus d'annotations.

6.3.6.1 Modèle de l'annotation contextualisée

Notre proposition s'inspire du modèle proposé par [Aubry, 2007]. Nous distinguons quatre parties dans l'annotation : le contenu, la cible, la caractérisation sémantique et le contexte.

Le contenu désigne le titre et le corps de l'annotation. Le corps de l'annotation représente les connaissances inscrites par l'auteur, c'est à dire un texte en langage naturel. Nous nous limitons volontairement au texte pour le contenu de l'annotation. En effet, bien que les périphériques mobiles actuels permettent l'enregistrement audio ou vidéo, il est beaucoup plus délicat d'extraire des informations sémantiques de séquences audio ou vidéo que de documents textuels.

La cible désigne l'œuvre à laquelle se rapporte l'annotation. À la cible est éventuellement associée une ancre afin de préciser la partie de l'œuvre à laquelle se rapporte l'annotation. Cette zone apparaîtra alors sous forme d'une ellipse superposée à l'image de l'objet lors de la consultation de l'annotation sur le dispositif.

Le contexte de l'annotation comporte deux catégories d'informations : les informations épisodiques précisent le nom de l'auteur et la date de l'annotation et les informations circonstanciées précisent le contexte perçu localement au moment du dépôt de l'annotation. Les informations contextuelles qui seront retenues dépendent de la sémantique de l'annotation ainsi que de l'œuvre annotée.

6.3.6.2 Proximités sémantiques et contextuelles entre annotations

Nous présentons ici comment sont caractérisées sémantiquement et contextuellement les annotations. Ces caractérisations nous permettent de définir des proximités sémantiques

et contextuelles entre annotations qui seront utilisées pour instrumenter une navigation alternative dans le corpus des annotations.

Caractérisation sémantique et contextuelle des annotations

Le processus de caractérisation sémantique et contextuelle que nous proposons se déroule en deux étapes. La première étape est la caractérisation sémantique de l'annotation à partir du titre, du corps de l'annotation et de la description sémantique de l'œuvre annotée. Cette étape nous permet de déterminer un ensemble d'instances $Sem_{annotation} = \{a_1 \dots a_N\}$ qui formeront la caractérisation sémantique de l'annotation. A l'issue de cette opération, nous appliquons une règle de caractérisation contextuelle qui définira le contexte de l'annotation. Ce contexte sera visualisable lors de la consultation de l'annotation afin d'aider à sa compréhension.

Caractérisation sémantique

Nous détaillons tout d'abord la construction de l'ensemble $Sem_{annotation}$, c'est-à-dire la récupération d'un ensemble d'instances à partir du texte déposé par un annotateur et de l'œuvre annotée. Le lien principal entre un texte et une base de connaissances est la terminologie. Les éléments de la base de connaissances sont associés à des termes *via* par exemple l'utilisation de propriétés telles que *skos:prefLabel* ou *skos:altLabel*. Le principe d'extraction des instances à partir du texte repose sur un *matching* entre les caractérisations lexicales des termes et les mots ou groupes de mots présents dans l'annotation. Ainsi, si le texte contient le terme « Napoléon », l'instance ayant pour représentant lexical « Napoléon » sera ajoutée à l'ensemble $Sem_{annotation}$.

Un problème se pose cependant en raison de l'ambiguïté de certains termes ou groupes de termes. En effet certaines instances peuvent avoir les mêmes représentants lexicaux. Un exemple assez évident est le groupe de termes « Vierge à l'enfant » qui désigne une multitude de peintures. Pour discriminer ce genre de cas, nous nous basons sur la caractérisation sémantique de l'entité annotée. Si l'entité annotée ou une de ses propriétés possède le représentant lexical « Vierge à l'enfant », l'instance correspondante sera ajoutée à $Sem_{annotation}$. Dans le cas où l'entité annotée ne permet pas de discriminer entre plusieurs instances, nous nous basons sur le contexte de la personne posant l'annotation, en recherchant dans le contexte historique, de localisation ou d'intérêt une instance ayant le bon représentant lexical. Si la discrimination reste impossible, les instances possédant le représentant lexical sont toutes ajoutées à $Sem_{annotation}$.

La deuxième étape de la caractérisation sémantique consiste à inclure dans $Sem_{annotation}$ les éléments sémantiques propres à l'œuvre annotée. Cette étape est relativement simple : nous ajoutons à l'ensemble $Sem_{annotation}$ l'instance représentant l'œuvre annotée ainsi que l'ensemble des instances constituant les propriétés de cette instance (auteur, style, thème, personnage...).

Caractérisation contextuelle

La caractérisation contextuelle de l'annotation est, dans sa forme, relativement similaire à la caractérisation sémantique. Il s'agit de construire trois ensembles $Hist_{annotation} = \{h_1 \dots h_i\}$, $Loc_{annotation} = \{l_1 \dots l_j\}$ et $Int_{annotation} = \{i_1 \dots i_k\}$ représentant des éléments pertinents pour resituer l'annotation dans le contexte dans lequel elle a été déposée. L'hypothèse que nous faisons est que pour aider à la compréhension d'une annotation, la visualisation d'éléments du contexte de l'annotateur aidera la personne visualisant l'annotation à comprendre le cheminement mental de l'annotateur et donc à mieux comprendre l'annotation. Par ailleurs cette caractérisation contextuelle nous permettra de définir une proximité contextuelle entre annotations offrant un mode de navigation alternatif entre annotations.

La construction des ensembles $Hist_{annotation}$, $Loc_{annotation}$ et $Int_{annotation}$ est basée sur l'application d'une règle portant sur la sémantique de l'annotation et le contexte de l'annotateur. Le principe est de retenir, parmi les éléments du contexte historique, de localisation et d'intérêt de l'annotateur, ceux ayant un rapport direct avec la sémantique de l'annotation. Ainsi, si l'annotation porte sur une œuvre impressionniste, il sera sans doute intéressant de connaître les autres œuvres impressionnistes qu'a vues précédemment l'annotateur mais il sera sans doute moins intéressant de connaître les œuvres modernes qu'a vues l'annotateur (à moins évidemment qu'il ne fasse référence à ces œuvres modernes dans l'annotation).

La règle de construction des ensembles $Hist_{annotation}$, $Loc_{annotation}$ et $Int_{annotation}$ est la même pour les trois ensembles. Nous la décrivons donc pour $Hist_{annotation}$ à titre d'exemple. Le principe de construction de l'ensemble est de trouver les éléments du contexte historique « proches », au sens de la proximité sémantique, de la caractérisation sémantique de l'annotation.

Pour chaque instance de la base de connaissances muséales impliquée dans le contexte historique du visiteur, une comparaison de la proximité sémantique avec chaque instance de la caractérisation sémantique de l'annotation est effectuée. Si pour une instance h du contexte historique, h est suffisamment proche d'une instance a de la caractérisation sémantique de l'annotation, alors h appartient au contexte historique de l'annotation.

$$\alpha \in [0, 1] h \in C_H, \exists a \in Sem_{annotation}, Sim_{num}(h, a) \geq \alpha \Rightarrow h \in Hist_{annotation}$$

La formule précédente formalise cette idée, le paramètre $\alpha \in [0, 1]$ définit le seuil de proximité minimale pour permettre à l'instance du contexte historique de faire partie du contexte d'annotation. Si $\alpha = 1$ seules les instances appartenant au contexte historique **et** au contexte d'annotation seront présentes, si $\alpha = 0$ toutes les instances du contexte historique seront présentes.

Proximités sémantiques et contextuelles entre annotations

Plusieurs méthodes lexicales ont été proposées pour évaluer la similarité entre documents. La méthode la plus usitée est certainement la représentation vectorielle de Salton [Salton & McGill, 1986], où les documents sont représentés par des vecteurs et leur proximité estimée par le produit scalaire de ces vecteurs. Ces approches ne permettent cependant pas toujours d'associer deux documents évoquant des sujets proches (par exemple « Napoléon » et « le Premier Empire »). Nous proposons une mesure de proximité sémantique qui permet de lier des annotations n'ayant éventuellement aucun terme commun mais évoquant des sujets proches. Nous utilisons pour cela la caractérisation sémantique de l'annotation.

Les calculs de proximités sémantique et contextuelle sont relativement similaires. Il s'agit à chaque fois de comparer deux ensembles d'instances de la base de connaissances pour en extraire une valeur de proximité dans l'intervalle $[0, 1]$. Nous avons donc choisi de nous inspirer de la méthode de calcul de proximité entre textes proposée par [Mihalcea *et al.*, 2006]. Cette méthode a été développée pour permettre la comparaison entre des documents textuels issus du web (textes courts). Dans la formulation originale, la similarité entre deux textes T_1 et T_2 est basée sur la formule suivante.

$$sim(T_1, T_2) = \frac{1}{2} \left(\frac{\sum_{\omega \in T_1} maxSim(\omega, T_2) \cdot idf(\omega)}{\sum_{\omega \in T_1} idf(\omega)} + \frac{\sum_{\omega \in T_2} maxSim(\omega, T_1) \cdot idf(\omega)}{\sum_{\omega \in T_2} idf(\omega)} \right)$$

L'interprétation de cette formule est assez naturelle : pour chaque mot ω du texte T_1 on cherche à déterminer le mot ayant le score de similarité le plus proche dans le texte T_2 (c'est la fonction *maxSim*) en pondérant le résultat obtenu par la spécificité du mot. Cette spécificité est calculée suivant l'indice classique *idf* (inverse document frequency) qui est simplement le logarithme du nombre total de documents dans le corpus divisé par le nombre total de documents impliquant ce mot. Moins le mot est employé, plus il est spécifique et plus *idf* est important.

Nous nous inspirons donc de cette mesure, en procédant à quelques modifications. Tout d'abord les textes T_1 et T_2 sont remplacés par les ensembles $Sem_{annotation1}$ et $Sem_{annotation2}$ construits à partir des annotations (voir **6.3.6.2 Caractérisation sémantique**), ensuite, la fonction de calcul de similarité *maxSim* est remplacée par la fonction de calcul numérique de similarité entre instances décrite précédemment. Reste la question de la fonction *idf*(ω). Dans le calcul de Mihalcea cette fonction représente la spécificité d'un terme par rapport à un corpus de document D :

$$idf(\omega) = \log \frac{|D|}{|d \in D, \omega \in D|}$$

$|D|$ étant le nombre total de documents du corpus et $|d \in D, \omega \in D|$ le nombre de documents contenant le terme ω . Afin de rester cohérents avec cet objectif, nous proposons la fonction $idf_{sem}(i)$ qui est le logarithme du rapport du nombre total d'instances d'œuvres sur le nombre d'instances d'œuvres associées à l'instance i . Ainsi, si une annotation traite d'un sujet représenté dans la plupart des œuvres, la contribution de l'instance représentant ce sujet à la similarité sera faible. Dans le cas contraire, si l'annotation traite d'un sujet peu présent dans les œuvres, la contribution de l'instance représentant ce sujet sera importante.

La similarité contextuelle entre deux annotations se calcule de la même manière que la similarité sémantique. La différence étant que les ensembles $Sem_{annotation1}$ et $Sem_{annotation2}$ sont remplacés par les caractérisations contextuelles respectives des annotations :

$\{Hist_{annotation1}, Loc_{annotation1}, Int_{annotation1}\}$.

Ces caractérisations sémantiques et contextuelles ainsi que les similarités entre annotations nous permettent de définir des modes de navigation et de visualisation originaux, prenant en compte la sémantique et le contexte des annotations.

6.3.6.3 Navigation et visualisation des annotations

Nous reprenons ici les questions posées au début de la section : comment sélectionner les annotations susceptibles d'intéresser l'utilisateur parmi un corpus potentiellement important d'annotations ? Comment donner à la personne qui visualise l'annotation une idée du contexte dans laquelle elle a été déposée ?

Navigation parmi les annotations

Pour répondre à la première question, nous définissons deux modes de navigation parmi les annotations : un mode contrôlé par le système, où les annotations sont sélectionnées et présentées de manière automatique, et un mode contrôlé par l'utilisateur offrant une fonction de recherche parmi les annotations.

Dans le premier mode de navigation, l'utilisateur visualise les annotations associées à une œuvre donnée, ces annotations étant présentées sous forme d'une liste en fonction de leur proximité contextuelle avec le contexte de l'utilisateur, l'hypothèse étant qu'un utilisateur sera plus intéressé par des annotations de personnes ayant eu un parcours similaire et des intérêts similaires aux siens.

Le deuxième mode de navigation est plus classique et repose sur une recherche par mots clés. L'utilisateur entre un certain nombre de termes dans le champ de recherche, ces termes sont associés, quand c'est possible, à des instances de la base de connaissances et les annotations sont présentées en fonction de leur proximité sémantique avec l'ensemble des instances représentant les termes suivant la formule modifiée de [Mihalcea *et al.*, 2006]

que nous avons proposée dans la section précédente. Il est ainsi possible d'accéder à des annotations éventuellement lointaines sur le plan géographique.

Visualisation des annotations

Lors de la visualisation d'une annotation contextualisée (figure 6.3.4), sont présentés l'œuvre, le texte et le titre de l'annotation, ainsi que deux barres latérales. La barre latérale de gauche présente le contexte historique de l'annotateur en relation avec l'annotation, c'est-à-dire les œuvres consultées par l'annotateur ayant un rapport avec la caractérisation sémantique de l'annotation. En permettant ainsi la visualisation de ce contexte historique, l'objectif est que la personne consultant l'annotation puisse se rendre compte des parcours géographique et intellectuel qui ont pu conduire à cette annotation.



FIGURE 6.3.4: Visualisation d'une annotation contextualisée

La barre latérale de droite présente des annotations sémantiquement et contextuellement proches, au sens de la proximité entre annotations que nous avons définie auparavant. L'objectif est ici d'offrir un parcours alternatif dans le corpus d'annotations en permettant, de proche en proche, de se faire une représentation plus générale des annotations qui ont été posées. La proximité sémantique et contextuelle garantit une certaine homogénéité dans ce parcours et doit aider la personne consultant les annotations à comparer les points de vue des différents annotateurs.

6.4 Conclusion

Nous avons présenté dans ce chapitre une méthode pour déterminer la proximité sémantique entre des instances de la base de connaissances, ainsi qu'une méthode pour déterminer l'adéquation au contexte de ces instances. Nous avons détaillé l'utilisation de ces proximités sémantiques et contextuelles dans le cadre de trois types d'interactions situées : la recommandation automatique d'œuvres, des activités de type auto-évaluation ainsi que le dépôt et la consultation d'annotations contextualisées.

Plusieurs remarques peuvent être faites quand aux interactions contextualisées que nous proposons. Tout d'abord, nous ne nous basons pas sur des méthodes spécifiques à la muséologie pour déterminer les proximités. Les méthodes que nous proposons sont génériques. Ces méthodes pourraient *a priori* être employées dans d'autres cadres applicatifs.

Ensuite, concernant les calculs de proximités en eux-mêmes, si l'obtention d'une valeur numérique de proximité est indispensable pour instrumenter l'interaction, la justification de cette proximité l'est tout autant. C'est ici qu'on peut voir un apport assez net des méthodes sémantiques par rapport aux autres méthodes de recommandation numériques, qui ne permettent pas d'obtenir des justifications aussi claires des relations entre les entités décrites.

Enfin, une limite apparaît quant à l'organisation des différentes activités. Nous n'avons pas décrit de formalisme permettant d'agencer de manière précise les différents types d'activité. Cette limite est liée au caractère informel de l'apprentissage en musée que nous proposons. Cependant, il pourrait être utile pour des enseignants de spécifier quand et comment s'organisent les activités de consultation, de jeu et d'annotation. Le chapitre suivant propose plusieurs pistes pour mettre en place une telle organisation.

Chapitre 7

Modes de contrôle du système

Une difficulté particulière des systèmes à base de connaissances pour l'apprentissage est de proposer des modes de contrôle explicites et simples d'utilisation permettant de modifier le fonctionnement des systèmes. Il est aujourd'hui admis que pour qu'une solution pédagogique soit adoptée, il faut laisser à l'enseignant une marge de manœuvre dans le « bricolage pédagogique » [Caron, 2007]. Nous détaillons ici des modes de contrôle spécifiques à l'enseignant. De la même manière, l'expert métier (ex. conservateur, guide conférencier) doit pouvoir comprendre le système que nous proposons et l'adapter en fonction de ses besoins ou de sa vision de ce que doit être le parcours muséal.

Plusieurs difficultés apparaissent dans la définition de modes de contrôle pour des non-informaticiens. La première est due à la complexité du système, basé sur des représentations sémantiques et des calculs complexes. Il est difficile de demander à un expert ou à un enseignant de comprendre le fonctionnement du système pour modifier ou influencer sur ce fonctionnement. Nous devons donc proposer des outils de contrôle ayant du sens pour l'expert en muséologie non-informaticien ou l'enseignant. La deuxième difficulté est liée aux calculs de proximités sémantiques et contextuelles. Les modifications apportées au système peuvent avoir des résultats non nécessairement évidents : par exemple une modification dans le mode de calcul des proximités aura un impact sur les suggestions proposées, mais cet impact peut être difficile à décrire de manière qualitative. D'autre part, l'utilisation en autonomie par des apprenants en sortie scolaire va imposer certaines contraintes dans les modes de contrôle. Par exemple, il ne s'agira pas pour l'enseignant d'associer une activité prévue à une œuvre mais de décrire les conditions qui mèneront à la présentation de tel type d'activité dans tel type de situation.

Pour répondre à ces problèmes, nous détaillons dans la suite deux modes de contrôle correspondant à deux catégories d'utilisateur du système ainsi qu'aux connaissances qu'on peut raisonnablement leur attribuer : connaissances muséologiques pour l'expert métier et connaissances pédagogiques pour l'enseignant. Ces modes de contrôle ne sont pas exhaustifs, mais ils permettent de couvrir un ensemble assez large de situations où

notre système pourrait être utilisé.

7.1 Contrôle du système par l'enseignant : intégration de connaissances pédagogiques

La question du contrôle pédagogique dans les systèmes d'apprentissage ubiquitaire fait apparaître une contradiction. En effet, un des principaux intérêts de ces environnements est de conserver le caractère authentique des situations d'apprentissage en accordant à l'apprenant une liberté importante lors du déroulement des sessions d'apprentissage. Cependant, en particulier dans le cadre d'un apprentissage scolaire, il est nécessaire d'offrir à l'enseignant un certain contrôle pédagogique sur le déroulement de la situation d'apprentissage. En l'absence de contrôle, l'apprenant risque en effet de se disperser à travers les interactions et informations disponibles. Il s'agit alors de déterminer comment offrir ce contrôle à l'enseignant tout en permettant à l'apprenant de conserver la liberté propre à l'apprentissage ubiquitaire.

Notre hypothèse concernant l'utilisation de CALM en classe est que la sortie scolaire s'inscrit dans une thématique liée au programme scolaire (ex. Moyen Âge pour des élèves de CM1). Cette thématique définit explicitement l'axe suivant lesquels la visite doit s'organiser.

Par ailleurs, nous supposons que même si l'enseignant ne dispose pas de connaissances approfondies en muséologie, il dispose de connaissances pédagogiques sur la manière d'organiser et de structurer les activités proposées en fonction du contexte de visite (ex. Proposer un QCM portant sur les personnages historiques des œuvres en fin de visite).

Nous présentons dans la suite deux modes de contrôle pédagogique : le contrôle thématique et le contrôle contextuel. Le premier mode de contrôle vise à inscrire l'ensemble des activités de visite dans une même thématique (ex. la thématique définie par le programme scolaire). Le deuxième mode de contrôle est plus spécifique. Il vise à proposer certains types d'activité en fonction d'un état du contexte, par exemple proposer à l'apprenant de répondre à une question ouverte si l'œuvre qu'il considère réunit un certain nombre de critères. Ces mode de contrôle doivent permettre une certaine liberté à l'apprenant tout en maintenant des axes de visite bien définis.

7.1.1 Contrôle thématique

Le premier type de contrôle que nous proposons porte sur l'ensemble de la visite et a pour objectif d'assurer une cohérence thématique tout au long du parcours. Ce contrôle s'applique durant les phases de navigation parmi les connaissances muséales et pour la

génération des jeux d'auto-évaluation. Il permet d'attirer l'attention des apprenants sur les éléments pertinents vis-à-vis de la thématique choisie par l'enseignant.

Pour cela, on propose à l'enseignant de choisir un certain nombre de ressources dans la base de connaissances muséales. Par exemple si la thématique choisie par l'enseignant porte sur « Le Premier Empire », ce dernier pourra sélectionner les personnages, lieux, événements, styles et œuvres en lien avec cette thématique. Pour l'assister dans ce choix, un outil de sélection des instances, basé sur la proximité sémantique définie plus haut, lui est proposé. Ainsi, en choisissant la ressource « Napoléon Bonaparte », il lui sera par exemple suggéré d'ajouter à la sélection « Marie-Louise », « Waterloo »... Cet ensemble d'instances, qui définit la thématique de visite, est noté T dans la suite.

Le choix de l'ensemble T va modifier le comportement du système pour les différentes activités proposées. Lorsque l'apprenant consulte la liste des œuvres à proximité, les œuvres les plus proches sémantiquement de l'ensemble T seront mises en évidence, de manière à l'inciter à considérer ces œuvres. De la même manière, lors de la consultation des informations relatives à une œuvre, si certains éléments (ex. styles, personnages) appartiennent ou sont proches de l'ensemble T , ces éléments seront mis en avant afin de favoriser leur consultation par l'apprenant.

Par ailleurs, le mode de calcul des suggestions contextualisées est modifié pour intégrer le choix de la thématique de visite. L'idée est ici de suggérer des éléments sémantiquement proches de l'ensemble T , tout en restant cohérent avec l'élément que consulte l'apprenant. Cette modification prend la forme d'un nouvel ordonnancement des suggestions, visant à favoriser les suggestions proches de T . Initialement les suggestions sont ordonnées simplement suivant leur proximité sémantique et contextuelle avec l'entité considérée : si l'apprenant consulte les informations relatives à une entité i (c'est-à-dire une instance de la base de connaissances correspondant par exemple à une œuvre ou un artiste), le score d'une suggestion s est donné par (6.3.3) :

$$Score(s) = \alpha \cdot Prox_{num}(i, S) + \beta \cdot Prox_{context}(i)$$

Avec $\alpha + \beta = 1$. $Prox_{context}(i)$ étant la somme des différentes proximités contextuelles relative à i . En prenant en compte la thématique de visite, le calcul du score d'une suggestion s par rapport à l'entité i devient :

$$Score(s) = \alpha \cdot Prox_{num}(i, s) + \beta \cdot Prox_{context}(i) + \gamma \cdot Prox_{moy}(i, T)$$

Avec $\alpha + \beta + \gamma = 1$. $Prox_{moy}(i, T)$ étant la moyenne de la proximité entre i et chaque instance de T .

Enfin, le mode de génération des jeux d'auto-évaluation est également modifié pour prendre en compte la thématique de visite. Initialement, les questions et les distracteurs sont choisis dans le contexte historique et de localisation, le score de pertinence d'un jeu vis-à-vis du contexte est calculé de la manière suivante :

$$Score(QCM, C_L, C_H) = \alpha \frac{|I(QCM) \cap I(C_L)|}{|I(QCM) \cup I(C_L)|} + \beta \frac{|I(QCM) \cap I(C_H)|}{|I(QCM) \cup I(C_H)|}$$

$I(C_L)$ et $I(C_H)$ désignant respectivement l'ensemble des instances du contexte de localisation et du contexte historique. Avec la prise en compte de l'ensemble T , le calcul du score devient :

$$Score(QCM, C_L, C_H, T) = \alpha \frac{|I(QCM) \cap I(C_L)|}{|I(QCM) \cup I(C_L)|} + \beta \frac{|I(QCM) \cap I(C_H)|}{|I(QCM) \cup I(C_H)|} + \gamma \frac{|I(QCM) \cap T|}{|I(QCM) \cup T|}$$

7.1.2 Contrôle contextuel

7.1.2.1 Règles de contrôle contextuel

L'objectif du contrôle contextuel est de permettre à l'enseignant de proposer des activités spécifiques à certains moments. On peut par exemple supposer que l'enseignant souhaite proposer un questionnaire en fin de visite sur les œuvres qui ont été vues en début de visite, ou bien que l'enseignant souhaite proposer une question ouverte relative à une œuvre à l'issue d'un certain nombre de consultations.

Le principe des règles de contrôle contextuel est de proposer une interaction donnée en fonction de l'état du contexte à un moment donné. La figure 7.1.1 présente l'état du contexte historique d'un visiteur après quelque temps passé dans le musée.

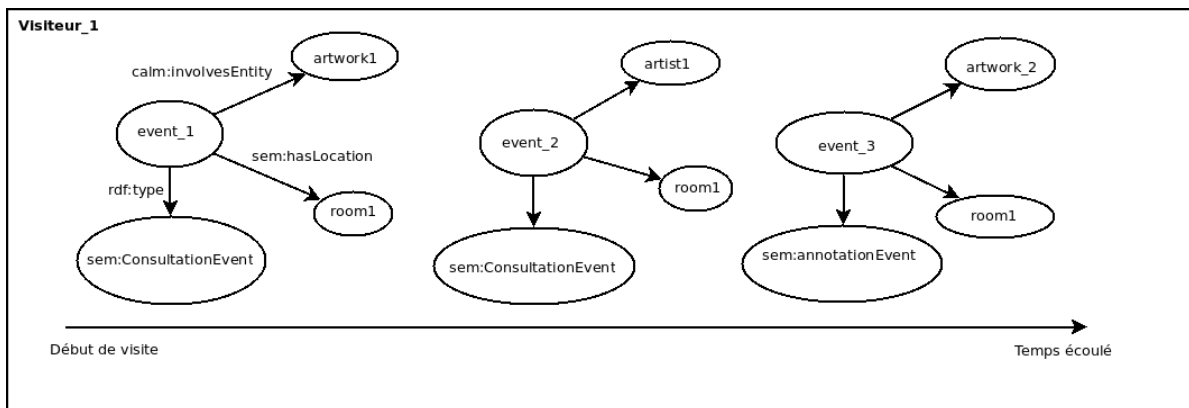


FIGURE 7.1.1: Contexte historique d'un visiteur

On constate que le contexte historique est effectivement un reflet de l'activité effective de l'apprenant dans le musée. Nous utilisons cette représentation contextuelle pour proposer

un formalisme de règles permettant à l'enseignant d'assigner des activités spécifiques à certains contextes.

Une règle de contrôle contextuel est définie par trois paramètres : l'intervalle de temps où cette règle est applicable, le contexte d'application de la règle (préconditions) et enfin les informations sur l'activité proposée lors du déclenchement de la règle.

La figure 7.2.2 montre deux exemples de règle de contrôle contextuel. La première spécifie qu'un QCM doit être proposé à l'apprenant en fin de visite (ex. à partir de la centième minute) sur les œuvres vues en début de visite. La seconde indique qu'une question ouverte doit être proposée à l'apprenant sur un tableau de type portrait impliquant un personnage appartenant à la thématique de visite.

```
Règle 1
Intervalle de temps : à partir de 100 min
Précondition : aucune
Activité : QCM
Sujet : SELECT ?artwork WHERE
      {
        ?eventrdf:type sem:ConsultationEvent.
        ?event sem:hasDateTime DATETIME.
        ?event sem:hasActor VISITOR.
        ?event sem:hasObject ?artwork.
        setDate(DATETIME,0,30).
      }

Règle 2
Intervalle de temps : Aucun
Précondition : (?activity rdf:type sem:ConsultEvent.)
              (?activity sem:hasActor VISITOR.)
              (?activity calm:involvesEntity ?artwork.)
              (?artwork calm:hasType getty:Portrait.)
              (?artwork cidoc:depict ?subject.)
              &&
              ?subject in T
Activité : Question ouverte
          : Quels sentiments exprime le personnage ?
```

FIGURE 7.1.2: Exemple de règles de contrôle contextuel

7.1.2.2 Règles de génération contextuelle

Les règles de contrôle contextuel permettent de proposer des jeux d'auto-évaluation aux apprenants à certains moments. Cependant l'apprenant peut souhaiter, de lui-même, effectuer un jeu d'évaluation à n'importe quel moment. Afin que ces jeux soient pertinents, nous proposons une méthode de génération automatique de jeux en fonction du contexte et de contraintes spécifiées par l'enseignant. Cette méthode est plus souple que les méthodes de contrôle contextuel car elle permet aux apprenants d'effectuer des jeux quand ils le souhaitent.

La méthode de génération automatique de quizz se base sur trois paramètres : une fonction de focalisation contextuelle, les taux d'adéquation aux contextes historique, d'intérêt et de localisation et une amplitude temporelle exprimée en minute.

Focalisation contextuelle

La fonction de focalisation contextuelle définit les éléments qui apparaîtront dans le quizz (sujet, distracteurs...), elle est déterminée par l'enseignant. L'objectif de la focalisation contextuelle est de traduire des contraintes pédagogiques sur les entités à prendre en

compte pour la génération du quizz en fonction de l'historique de visite de l'apprenant. Par exemple, poser une question sur les œuvres vues il y a 10 minutes.

La focalisation contextuelle permet d'obtenir un ensemble d'instances avec lesquelles l'utilisateur a été en contact (*via* les œuvres) à un moment donné. Le focus contextuel est l'ensemble des entités (instances de la base de connaissances) issues de la focalisation contextuelle.

Nous présentons maintenant notre formalisme pour les règles de focalisation contextuelle. Une règle de focalisation est constituée de deux paramètres : une fonction de sélection des événements et une amplitude. La fonction de sélection des événements F_{select} est une fonction du contexte historique dans la base de connaissance muséale, l'amplitude est exprimée en minutes.

Les deux règles suivantes illustrent la focalisation contextuelle à partir de contraintes pédagogiques. Dans ces règles VISITOR désigne l'URI du visiteur auquel s'applique la règle.

Règle 3

- *Contrainte pédagogique* : prendre en compte les œuvres vues il y a 10 minutes plus ou moins 5 minutes
- F_{Select} : SELECT ?arwork WHERE
{?event rdf:type sem:ConsultationEvent.
?event sem:hasDateTime DATETIME
?event sem:hasActor VISITOR.
?event sem:hasObject ?artwork.
setDateTime(DATETIME,currentDateTime(),10).}
- *Amplitude* : 5 minutes

Règle 4

- *Contrainte pédagogique* : prendre en compte tous les personnages vus
- F_{Select} : SELECT ?character WHERE
{?event rdf:type sem:ConsultationEvent.
?event sem:hasActor VISITOR.
?event sem:hasObject ?artwork.
?artwork cidoc:depicts ?character}
- *Amplitude* : 0 minute

Génération de quizz à partir du contexte

L'objectif est ici de traduire les connaissances de l'enseignant sur les questions « intéressantes » à poser, en dehors des questions explicitement spécifiées en 7.1.2.1. Notre

hypothèse est que les questions « intéressantes » concernent des éléments qui intéressent le visiteur qui sont autour de lui, ou qui ont déjà été vus.

Nous définissons alors une règle de génération de quizz comme un quadruplet $T_L, T_I, T_H, F_{select}$. Où T_L, T_I, T_H sont les taux d'adéquation au contexte de localisation, historique et d'intérêt et F_{select} la règle de focalisation contextuelle employée.

On rappelle que pour calculer ces taux d'adéquation, nous adaptons le calcul de l'indice de Jaccard entre deux ensembles, on a donc :

$$T_X(Q) = \frac{|Q \cap D_X|}{|Q|}$$

où T_X est le taux d'adéquation à l'espace contextuel X , D_X l'ensemble des instances présentes dans cet espace contextuel et Q l'ensemble des entités présentes dans le quizz.

L'enseignant fixe alors les valeurs souhaitées de T_L, T_H et T_I . Le choix d'un quizz respectant les contraintes consiste alors à énumérer l'ensemble des quizz possibles pour en choisir un qui corresponde aux spécifications de l'enseignant.

Cette méthode présente cependant une faiblesse due au caractère « tout ou rien » de l'indice de Jaccard. En effet, soit un élément est présent dans les deux ensembles (Q et D_X), soit il n'est pas pris en compte. La conséquence directe de ce caractère tout ou rien est que, si les taux d'adéquation au contexte sont élevés et qu'il y a peu d'entités communes aux différents contextes, aucun quizz ne respectera les taux d'adéquation. Nous présentons dans la suite une méthode, basée sur la relaxation contextuelle, permettant de résoudre cette question.

Relaxation sémantique et contextuelle

Le processus de relaxation est classiquement employé dans les systèmes de requêtes basés sur des technologies sémantiques. Il s'agit, en l'absence de résultats de la requête, de relâcher de manière appropriée certaines contraintes afin de retourner des résultats [Hurtado *et al.*, 2006].

Dans le problème qui nous occupe, nous souhaitons relaxer les contraintes sur les entités peuplant les espaces contextuels afin de permettre la génération de quiz ayant les taux d'adéquation définis par l'expert ou le pédagogue. Il s'agit donc, pour chaque espace, de définir une méthode de relaxation R permettant de passer d'un ensemble E d'instances à un ensemble E_R tel que $E \subset E_R$. Nous proposons deux types de relaxation suivant les espaces contextuels à relaxer : la **relaxation sémantique** et la **relaxation contextuelle**.

La relaxation sémantique concerne l'espace contextuel "intérêts du visiteur". Cet espace est en effet peuplé d'instances de la base de connaissances qui ne sont pas liées à un

contexte physique particulier. Notre hypothèse est alors que si un utilisateur a un intérêt pour une instance de la base de connaissances (ex. Napoléon Bonaparte), il aura également un intérêt pour les propriétés d'intérêt de cette instance.

Ainsi, si un utilisateur a un intérêt pour l'ensemble E constitué de l'instance *napoleon-Bonaparte*, l'ensemble E_R est constitué de E et de l'ensemble des valeurs des propriétés d'intérêt de cette instance. Ici on aura donc :

$$E_R = \{napoleonBonaparte, josephine, batailleWagram\}$$

Cette relaxation nous permet donc bien d'obtenir un ensemble E_R tel que $E \subset E_R$, le caractère sémantique de la relaxation étant dû au fait que cette relaxation est basée sur la matrice des propriétés d'intérêt de l'ontologie que nous avons définie plus tôt (cf chapitre 6).

La **relaxation contextuelle** concerne les espaces de localisation et historique. Dans le cadre de la relaxation basée sur le contexte de localisation, nous avons donc choisi de définir une relaxation élémentaire comme l'ajout à ce contexte de l'ensemble des entités *daisy:Location* immédiatement adjacentes à celles du contexte de localisation.

La relaxation du contexte historique dépend essentiellement des règles de focalisation utilisées. Ces règles peuvent être relaxées de deux façons : en relaxant la contrainte sur la fonction de choix des événements ou en relaxant l'amplitude. Nous avons choisi de retenir cette dernière méthode ; en effet une relaxation sur la fonction de choix des événements risque de ne plus faire correspondre la règle aux contraintes pédagogiques qu'elle exprime. La limite de la relaxation portant sur l'amplitude est que dans le cas où la règle pédagogique ne porte pas sur les aspects temporels (ex. règle 4), l'augmentation de l'amplitude ne permettra pas d'augmenter le nombre d'instances.

7.1.3 Conclusion sur l'intégration de connaissances pédagogiques

Comme nous l'avons vu, il est possible pour l'enseignant de spécifier de manière simple et sans connaissance particulière en muséologie, des axes qui vont orienter la visite ainsi que des règles contextuelles, propres à une activité donnée. Un aspect intéressant de l'utilisation de ces règles est qu'elle ne vont pas « enfermer » l'apprenant dans un scénario fixé, mais plutôt, orienter en partie son activité.

D'autres types de règles, de conditions de déclenchement ou même d'activités peuvent facilement être envisagés. Cela amène l'idée de scénarisation de la visite et plus largement de scénarisation dans le cadre d'un apprentissage ubiquitaire. Bien que cela sorte du cadre de cette thèse, nous présenterons en perspective une discussion sur les modalités que pourrait prendre une telle scénarisation.

7.2 Contrôle du système par l'expert : expression des proximités

Nous avons présenté au chapitre précédent une méthode de calcul très simple pour l'agrégation du vecteur numérique de proximités entre instances : la moyenne arithmétique. Cette méthode n'est pas satisfaisante dans la mesure où elle donne à chaque propriété la même importance dans le calcul des proximités. Par exemple, les propriétés *hasTheme* et *hasArtist* d'une œuvre ont la même importance pour déterminer sa proximité avec une autre œuvre. Il semble cependant évident que pour un expert en muséologie, toutes les propriétés n'auront pas la même importance.

Une solution immédiate à ce problème serait de permettre à l'expert en muséologie d'associer des poids aux propriétés (c'est à dire de pondérer les critères d'agrégation). Nous obtiendrions ainsi une moyenne pondérée comme opérateur d'agrégation. Mais nous allons le voir, cette réponse ne permet pas de prendre en compte un aspect important : la dépendance entre critères d'agrégation.

Nous décrivons dans la suite l'utilisation de l'intégrale de Choquet comme opérateur d'agrégation. Cet opérateur permet de prendre en compte la dépendance entre critères d'agrégation. La définition d'une intégrale de Choquet demande de fixer de nombreux paramètres. Nous proposons un certain nombre de critères permettant de fixer les paramètres de l'intégrale de Choquet à partir de l'ontologie.

7.2.1 Présentation informelle du problème : insuffisance de la moyenne pondérée

Nous illustrons l'insuffisance de la moyenne pondérée à partir d'un exemple adapté de [Grabisch, 2006]. Supposons qu'il faille choisir entre 3 œuvres à suggérer à l'utilisateur. Pour simplifier le problème, nous considérons que les œuvres sont comparées suivant deux critères, le critère *Thème* et le critère *Personnage*. Plus une œuvre a un score important sur un critère et plus il est pertinent de la présenter à l'utilisateur. Le tableau suivant présente trois œuvres (x , y et z) candidates pour la suggestion, et leur score de pertinence suivant les deux critères. Ces scores de pertinence correspondent à la proximité entre l'œuvre considérée par l'utilisateur et l'œuvre candidate.

Critère\œuvre	x	y	z
<i>Thème</i>	0.4	0	1
<i>Personnage</i>	0.4	1	0

TABLE 7.1: Critères de comparaison d'une œuvre

On a donc la valeur 0.4 sur le critère *Thème* dans le cas de l'œuvre x , la valeur 0 pour ce même critère pour l'œuvre y et la valeur 1 pour ce critère pour l'œuvre z . Supposons que, dans cette situation, l'expert recommande de présenter d'abord l'œuvre x , et ensuite indifféremment l'œuvre y ou l'œuvre z . Le raisonnement de l'expert pour cette présentation peut être : si une œuvre présente des proximités aussi bien sur les thèmes que sur personnages, il est pertinent de présenter cette œuvre, même si ces proximités sont faibles, deux œuvres similaires uniquement sur les thèmes ou sur les personnages pouvant être présentées ensuite de manière indifférente.

De manière assez surprenante, si on utilise la moyenne pondérée, il n'existe aucune solution permettant de conserver l'ordre privilégié par l'expert. Pour s'en convaincre, notons w_1 le poids associé à *Thème* et w_2 le poids associé à *Personnage*. Les contraintes définies par l'expert s'expriment de la manière suivante (en utilisant les notations classiques \prec et \succ pour les ordres partiels et \sim pour l'équivalence) :

$$\begin{aligned} Y &\sim Z \iff w_1 = w_2 \\ X &\succ Y \iff 0.4(w_1 + w_2) > w_2 \end{aligned}$$

Ce qui aboutit à $0.8w_2 > w_2$ ce qui est manifestement impossible, w_2 devant rester strictement positif.

Pour surmonter ce problème, on peut considérer des poids non seulement sur les critères uniques mais également sur les groupes de critères. Ainsi le groupe de critères *Thème* et *Personnage* peut avoir un poids plus important que la somme des poids des critères *Thème* et *Personnage* pris individuellement. Ceci nous conduit naturellement à l'utilisation de mesures non additives, en particulier à l'intégrale de Choquet dans laquelle les poids ne sont plus attribués uniquement à des critères uniques mais aussi à des groupes de critères.

7.2.2 Formalisation : utilisation de l'intégrale de Choquet

On considère qu'un objet X , par exemple une peinture, est représenté par un ensemble $N = \{1 \dots n\}$ de critères. Ces critères prennent leurs valeurs dans $[0, 1]$. Notons $\{X_1 \dots X_n\}$ les valeurs que prennent ces critères pour l'objet X . Plus un objet est satisfaisant sur le critère i , plus X_i sera important. Dans notre cas, les critères correspondent aux proximités établies entre X et une autre instance du même concept d'intérêt.

L'intégrale de Choquet est basée sur la notion de capacité. Une capacité est définie de la manière suivante. Notons $N = \{1 \dots n\}$ l'ensemble des critères. Une capacité sur N classiquement notée ν est une fonction de l'ensemble des parties des critères $\mathcal{P}(N)$ vers $[0, 1]$ avec les conditions suivantes :

$$\begin{aligned}\nu(N) &= 1 \\ \nu(\emptyset) &= 0 \\ \forall i \in N, \forall T \subset P(N), \nu(T \cup i) &\geq \nu(T)\end{aligned}$$

L'opérateur d'agrégation de Choquet, donnant le « score » de l'objet X suivant les valeurs des critères X_i , est alors calculé de la manière suivante [Detyniecki, 2001]:

$$C_\nu(X) = \sum_{i=1}^n X_{(i)}[\nu(A_i) - \nu(A_{i+1})]$$

Avec $(.)$ une permutation de N telle que $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)}$, $A_i = \{(i) \dots (n)\}$ et $A_{n+1} = \emptyset$.

Reprenons l'exemple proposé en 7.1.1. Pour déterminer le score d'un objet suivant l'intégrale de Choquet nous devons donc déterminer : $\nu(\text{Thème}, \text{Personnage})$, $\nu(\text{Thème})$ et $\nu(\text{Personnage})$.

Nous souhaitons par ailleurs que les préférences de l'expert soient modélisées, c'est-à-dire que les scores des œuvres x , y et z vérifient $C_\nu(X) > C_\nu(Y)$ et $C_\nu(Y) = C_\nu(Z)$.

Par définition d'une capacité, on a directement $\nu(\text{Thème}, \text{Personnage}) = 1$, il reste donc à déterminer le score des deux autres critères. L'expert attribue le même score total aux objets y et z ce qui se traduit par $\nu(\text{Thème}) = \nu(\text{Personnage})$. Par ailleurs ce score doit être suffisamment faible pour conserver l'inégalité $C_\nu(X) > C_\nu(Y)$. On peut alors choisir $\nu(\text{Thème}) = 0.3$.

On vérifie alors facilement que l'ordre proposé par l'expert est modélisé par une intégrale de Choquet sur ν :

$$\begin{aligned}C_\nu(X) &= C_\nu(0.4, 0.4) \\ &= 0.4(\nu(\text{Thème}, \text{Personnage}) - \nu(\text{Thème})) + 0.4 \cdot \nu(\text{Thème}) \\ &= 0.4(1 - 0.3) + 0.3 \times 0.4 \\ &= 0.4 \\ C_\nu(Y) &= C_\nu(0, 1) \\ &= 0 \cdot (\nu(\text{Thème}, \text{Personnage}) - \nu(\text{Personnage})) + 1 \cdot (\nu(\text{Personnage})) \\ &= 0.3\end{aligned}$$

On retrouve bien à travers les scores l'ordre des préférences telles que définies par le décideur (on a directement $C_\nu(Y) = C_\nu(Z)$ par symétrie).

L'utilisation de l'intégrale de Choquet est relativement aisée dans le cas de deux critères. Supposons que nous soyons dans le cas d'une agrégation sur les trois critères $\{1, 2, 3\}$, pour calculer le « score » de l'objet $X = \{x_1, x_2, x_3\}$ en supposant $x_3 \leq x_1 \leq x_2$ on a :

$$\begin{aligned} C\nu(x_1, x_2, x_3) = & x_3(\nu(\{1, 2, 3\}) - \nu(\{1, 2\})) \\ & + x_1(\nu(\{1, 2\}) - \nu(\{2\})) \\ & + x_2(\nu(\{2\})) \end{aligned}$$

7.2.2.1 Importance et interaction des critères

Plusieurs caractérisations des capacités ont donc été proposées qui permettent d'analyser le comportement de l'intégrale de Choquet, les caractérisations les plus couramment utilisées sont les indices d'importance et d'interaction des critères.

L'indice d'importance d'un critère quantifie l'importance globale de ce critère. Dans le cas de la moyenne pondérée, cet indice correspond au poids du critère. Cet indice est noté $\phi(I)$ pour un critère I et peut se calculer à partir de la capacité (voir Annexe **A**). On l'utilise de deux façons : pour vérifier que la capacité que l'on a définie a bien le comportement souhaité (les éléments jugés importants ont bien un indice d'importance élevé) ou bien l'indice peut être considéré comme une contrainte fixée par un expert pour la construction d'une capacité, la capacité solution devant alors vérifier cette contrainte.

Le deuxième indice est l'indice d'interaction. Cet indice quantifie, la quantité de synergie entre les membres d'un groupe de critères. Il peut se calculer à partir d'une capacité. Si un groupe de critères A à un indice d'interaction positif, il y a complémentarité entre les critères : si chaque élément du groupe de critères est satisfait, le score final de l'objet sera important. Le cas inverse peut se produire : l'indice d'interaction peut être négatif. Il y a alors redondance sur cet ensemble de critères et le score global d'un objet satisfaisant un ensemble de critères redondant sera moins important. De la même manière que pour l'indice d'importance, l'indice d'interaction peut être utilisé soit pour vérifier qu'une capacité a le comportement souhaité, soit être fixé par l'expert pour construire une capacité.

Nous considérons dans la suite que les capacités que nous utilisons sont 2-additives. C'est à dire que la valeur des indices d'interaction est nulle au delà de trois indices. Cette hypothèse simplificatrice est couramment utilisée [Grabisch, 2006].

7.2.3 Ajouts de contraintes liées à l'ontologie pour l'identification des capacités

7.2.3.1 Type de contraintes envisageables

Nous cherchons ici à simplifier le travail de l'expert, en déterminant des contraintes sur la capacité en fonction de l'ontologie et de la base de connaissances utilisées. Ces contraintes permettent par la suite de simplifier la construction de la capacité en réduisant l'espace des possibles.

Formellement, en supposant qu'on dispose d'un ensemble N de critères, d'un ensemble X d'alternatives sur ces critères, et des scores d'utilité individuelle x_i de chaque alternative, les informations données par l'expert peuvent prendre les formes suivantes [Marichal & Roubens, 2000]:

- Un pré-ordre partiel \preceq_X sur X (classement d'un ensemble d'alternatives)
 - Un pré-ordre partiel \preceq_N sur N (classement des critères suivant leur importance)
 - Un pré-ordre partiel \preceq_P sur l'ensemble des paires de critères (classement des paires de critères suivant les indices d'interaction)
 - Le signe des indices d'interaction entre certaines paires de critères ($I_{ij} < 0, = 0, > 0$)
- Dans notre cas, les critères sont basés sur les proximités vis-à-vis de certaines propriétés d'une ontologie. Des liens sémantiques, définis dans l'ontologie, pré-existent donc entre les critères. Nous proposons d'utiliser ces liens pour définir des contraintes sur ces critères et réduire l'effort demandé à l'expert. Nous distinguerons les contraintes nécessaires, c'est-à-dire les contraintes qui doivent impérativement être vérifiées sous peine de ne pas être cohérent avec l'ontologie, des contraintes *a priori*, c'est-à-dire des contraintes raisonnables, mais que l'expert peut modifier sans être en contradiction avec l'ontologie.

7.2.3.2 Classement des critères suivant leur importance

Le premier type de contrainte, le classement des différentes alternatives est difficilement envisageable sans l'expert. Etudions le cas du deuxième type de contrainte, le classement des critères suivant leur importance. Supposons que l'on doive donner l'ordre d'importance de deux critères correspondant à deux propriétés P_1 et P_2 telles que P_1 *rdfs:subProperty* P_2 alors, nécessairement $\phi(P_1) > \phi(P_2)$ puisque d'après la définition de *rdfs:subProperty*, $P_1 \Rightarrow P_2$, c'est-à-dire qu'on a forcément plus d'information en sachant que deux objets sont proches vis-à-vis de P_1 que de P_2 . La contrainte suivante est donc nécessaire :

$$P_1 \text{ rdfs:subProperty } P_2 \Rightarrow \phi(P_1) > \phi(P_2)$$

Nous pouvons généraliser ce raisonnement pour construire une contrainte *a priori*. L'idée étant que plus une propriété est spécifique, plus le critère qui est exprimé par cette propriété est important. Par exemple, la propriété *hasCountry* qui a pour domaine le concept *cidoc:Person* est moins spécifique que la propriété *hasStyle* qui a pour domaine le concept *calm:Artist*, il est alors raisonnable de supposer que $\phi(\text{hasCountry}) < \phi(\text{hasStyle})$.

Pour comparer les propriétés suivant leur spécificité nous devons disposer d'une mesure de la spécificité des propriétés. Ce type de mesure a été proposé par [Anyanwu *et al.*, 2005] qui proposent de mesurer la spécificité d'une propriété en fonction de la probabilité de "rencontrer" cette propriété en faisant un tirage aléatoire dans l'ensemble des propriétés de la base de connaissances. La spécificité d'une propriété P_i , vis-à-vis d'une base de connaissances B est alors donnée par :

$$\text{Spec}(P_i) = \frac{[P_i]_B}{[P]_B}$$

$[P_i]_B$ étant le nombre d'occurrences de la propriété P_i dans la base de connaissances et $[P]_B$ le nombre total d'occurrences de propriétés dans la base de connaissances. Nous proposons d'utiliser cette définition de la spécificité des propriétés pour établir un classement *a priori* sur l'importance des critères en présupposant que plus une propriété est spécifique au sens de [Anyanwu *et al.*, 2005] plus le critère auquel correspond cette propriété à un indice d'importance élevé.

On constate aisément que cette hypothèse est compatible avec l'hypothèse nécessaire précédente, c'est-à-dire que $P_1 \text{ rdfs:subProperty } P_2 \Rightarrow \phi(P_1) > \phi(P_2)$, puisque l'occurrence d'une sous-propriété P_1 dans la base de connaissances implique l'occurrence de la sur-propriété P_2 . La sous-propriété est donc forcément plus "rare" et donc plus spécifique.

7.2.3.3 Signes des indices d'interaction

Intéressons-nous maintenant au dernier type de contrainte, les signes des indices d'interaction. Commençons comme précédemment par considérer le cas de propriétés liées entre elles par des relations de type *rdfs:subProperty*. En reprenant le raisonnement sur l'importance des critères, nous pouvons dégager la contrainte nécessaire suivante :

$$P_1 \text{ rdfs:subProperty } P_2 \Rightarrow I_{P_1 P_2} < 0$$

$I_{P_1 P_2}$ étant l'indice d'interaction de la paire de critères qui représente respectivement P_1 et P_2 . Cette contrainte traduit simplement le fait que deux objets similaires vis-à-vis de P_1 seront nécessairement similaires vis-à-vis de P_2 .

Intéressons-nous maintenant aux domaines et codomaines des propriétés et essayons de dégager des règles raisonnables *a priori* sur l'interaction entre propriétés en fonction

des rapports entre leurs domaines et codomains. Un exemple de règle *a priori* est de considérer que deux propriétés ayant même domaine et même codomaine vont exprimer des connaissances proches. Par exemple, les propriétés *hasBirthDate* et *hasDeathDate* auront certainement un indice d'interaction inférieur à 0, dans la mesure où tant la date de naissance que la date de mort d'un artiste permettent de le situer sur le plan historique.

Nous proposons donc de lister les relations pouvant exister entre domaines et codomains des propriétés pour en tirer des règles raisonnables (mais non nécessaires) sur les indices d'interaction de ces propriétés. Nous utilisons pour exprimer les relations entre domaines et codomains les notations ensemblistes issues des logiques de description. Si A et B sont deux concepts d'une ontologie, les relations suivantes peuvent exister entre A et B :

1. $A \sqsubseteq B$
2. $B \sqsubseteq A$
3. $A \cap B \equiv \emptyset$
4. $A \equiv B$

A partir de ces cas de figure, nous pouvons construire une matrice donnant une valeur *a priori* raisonnable pour le signe de l'interaction en fonction des relations entre domaines et codomains des propriétés. Le tableau suivant présente les signes des critères d'interactions *a priori* entre deux propriétés P_1 et P_2 suivant les relations entre leurs domaines (D_1 et D_2) et leurs codomains (C_1 et C_2) respectifs.

Codomaine \ Domaine	$D_1 \sqsubseteq D_2$	$D_2 \sqsubseteq D_1$	$D_1 \equiv D_2$	$D_1 \cap D_2 \equiv \emptyset$
$C_1 \sqsubseteq C_2$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \geq 0$
$C_2 \sqsubseteq C_1$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \geq 0$
$C_1 \equiv C_2$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \leq 0$	$I_{P_1 P_2} \geq 0$
$C_1 \cap C_2 \equiv \emptyset$	$I_{P_1 P_2} \geq 0$	$I_{P_1 P_2} \geq 0$	$I_{P_1 P_2} \geq 0$	$I_{P_1 P_2} \geq 0$

TABLE 7.2: Signes des indices d'interaction suivant les domaines et codomains

Nous détaillons le raisonnement pour quelques paires de critères. Si $C_1 \equiv C_2$ et $D_1 \equiv D_2$ les propriétés P_1 et P_2 ont même domaine et même codomaine. Dans ce cas on peut supposer que ces deux propriétés expriment une relation semblable, c'est-à-dire qu'il existe une dépendance entre les deux propriétés. C'est le cas par exemple des propriétés *hasStudent* et *influencedBy*, ces deux propriétés ont le concept *Artist* comme domaine et codomaine et il est raisonnable de supposer que deux artistes proches vis-à-vis de la relation *hasStudent* le seront vis-à-vis de la relation *influencedBy*, c'est-à-dire que l'interaction entre les propriétés est négative.

A contrario, si les domaines des propriétés sont les mêmes et que les codomains des deux propriétés ont une intersection vide, c'est-à-dire si $D_1 \equiv D_2$ et $C_1 \cap C_2 \equiv \emptyset$ on peut supposer qu'il existe une dépendance positive ou nulle entre les deux propriétés. Ce sera le cas par exemple pour une peinture entre les propriétés *hasStyle* et *hasTheme*.

7.2.4 Intégration des préférences de l'expert et construction de la capacité

Nous disposons à présent d'un ensemble de contraintes raisonnables ou nécessaires, déterminées de manière automatique en fonction de l'ontologie ou de la base de connaissances. En utilisant l'algorithme de construction de capacité proposée par [Kojadinovic, 2006] (Annexe A) il est possible de trouver une capacité solution respectant l'ensemble de ces contraintes.

Afin de valider les contraintes raisonnables, calculées automatiquement, nous proposons à l'expert de répondre à un ensemble de questions d'ordonnement. Chaque question proposant un choix entre deux œuvres à présenter vis-à-vis de leur proximité avec une troisième œuvre. Lorsque l'expert est d'accord avec le classement fourni par la capacité identifiée, il n'y a pas de changement à faire.

Si l'expert n'est pas d'accord avec le classement proposé, on tente de construire une nouvelle capacité en accord avec les dépendances calculées et le classement de l'expert en utilisant l'algorithme de [Kojadinovic, 2006]. S'il existe une capacité solution, cette capacité remplace la capacité calculée *a priori*. S'il n'existe pas de solution, on cherche quelles contraintes calculées *a priori* sont en contradiction avec le classement donné, et on propose à l'expert de modifier ces contraintes *a priori*.

7.2.5 Illustration

Nous illustrons le mécanisme de calcul proposé sur un exemple relativement simple. On souhaite déterminer une capacité permettant d'ordonner des artistes suivant 4 critères. Ces critères correspondent aux proximités suivant les propriétés suivantes :

1. *Cidoc:tookEndofExistence* : P_1
2. *Cidoc:dieIn* : P_2
3. *Cidoc:performed* : P_3
4. *Cidoc:hasStyle* : P_4

Avec les relations entre propriétés :

1. *Cidoc:dieIn* *rdfs:subproperty* *Cidoc:tookEndOfExistence*

De part la relation de type *rdfs:subProperty*, on peut déduire les contraintes nécessaires suivantes

$$\begin{aligned}\phi(P_2) &> \phi(P_1) \\ I_{P_1 P_2} &< 0\end{aligned}$$

Par ailleurs, en considérant les domaines et codomaines des indices d'interaction, on peut dégager les contraintes *a priori* suivantes :

$$\begin{aligned}
 I_{P_1 P_3} &\geq 0 \\
 I_{P_2 P_4} &\geq 0 \\
 I_{P_1 P_4} &\geq 0 \\
 I_{P_2 P_3} &\geq 0
 \end{aligned}$$

Ces contraintes nous permettent de construire une capacité candidate, c'est-à-dire vérifiant l'ensemble des contraintes fixées. Pour cela nous utilisons le logiciel de calcul scientifique R¹ ainsi que la librairie Kappalab.

Par application directe de l'approche proposée par [Kojadinovic, 2006] (cette approche est implémentée dans Kappalab), nous obtenons les valeurs suivantes pour les indices d'importances:

	P_1	P_2	P_3	P_4
$\phi(P_i)$	0.06	0.56	0.189	0.189

TABLE 7.3: Valeur des indices d'importances

Ces valeurs ne sont pas surprenantes : on a bien $\phi(P_1) > \phi(P_2)$, et en l'absence de contraintes sur les autres propriétés, les même valeurs pour $\phi(P_3)$ et $\phi(P_4)$.

Concernant les indices d'interaction, on peut le représenter dans un tableau à double entrée ; en effet la capacité étant 2-additive, la valeur des indices d'interactions est nulle au-delà de deux indices :

Propriété	P_1	P_2	P_3	P_4
P_1	\emptyset	-0.1	0.01	0.01
P_2	-0.1	\emptyset	0.13	0.13
P_3	0	0.13	\emptyset	0
P_4	0	0.13	0	\emptyset

TABLE 7.4: Valeur des indices d'interactions

La valeur des indices d'importance et d'interaction définit entièrement la capacité solution. Par construction, cette capacité vérifie les contraintes spécifiées à partir de l'ontologie. Afin d'affiner le modèle il faudrait maintenant proposer plusieurs classifications à l'expert et intégrer ces résultats dans le calcul de la capacité.

1. <http://www.r-project.org/>

7.2.6 Conclusion sur l'intégration des connaissances de l'expert en muséologie

Nous avons présenté une méthode d'intégration des connaissances expertes assez fine ; cette méthode est basée sur une utilisation des intégrales de Choquet qui n'a, à notre connaissance, pas encore été utilisée dans les calculs de proximités sémantiques. Cette méthode offre l'avantage de prendre en compte les interactions entre propriétés, la contrepartie étant une difficulté plus importante pour la construction de la fonction d'agrégation que pour des agrégations plus classiques (moyenne pondérée, géométrique...). Nous avons cependant proposé un certain nombre de contraintes, basées sur la structure de l'ontologie et de la base de connaissances, qui permettent de simplifier le travail de l'expert dans la construction de cette fonction.

Notons que la prise en compte des connaissances expertes porte uniquement sur l'importance des propriétés et sur les interactions entre ces propriétés. Ceci garantit une certaine généralité de cette méthode. Les propriétés utilisées pour décrire les œuvres ne sont pas susceptibles de changer dans les musées de même type (ex. musées historiques, musées d'histoire naturelle). Cependant un expert peut posséder des connaissances locales sur l'importance des œuvres, des artistes... c'est-à-dire des connaissances sur les instances d'un musée particulier. Nous discuterons dans les perspectives de notre travail la manière dont pourraient être intégrées ces connaissances "locales".

7.3 Conclusion

Nous avons présenté, dans le chapitre précédent, plusieurs activités réalisables à l'aide d'un dispositif mobile dans un musée. Ces activités sont générées automatiquement en fonction de proximités sémantiques et contextuelles. Nous avons montré dans ce chapitre que ces activités peuvent être adaptées en fonction de deux types d'utilisateurs du système : l'expert en muséologie et l'enseignant accompagnant un groupe d'élèves dans le musée.

En terme de contrôle du fonctionnement du système, on peut constater que l'utilisation de méthodes de calcul numérique (intégrales de Choquet, proximité entre une instance et le contexte utilisateur...) couplées à une modélisation sémantique du domaine et du contexte permettent une certaine finesse dans le contrôle des interactions proposées. L'approche sémantique permet de rendre compte de relations logiques entre les différentes ressources et donne un caractère intelligible à ces relations logiques. L'approche numérique permet de déterminer de manière fine les relations potentiellement les plus intéressantes à présenter à l'utilisateur.

Implémentation et évaluation

Chapitre 8

La plateforme CALM

Nous décrivons dans ce chapitre les travaux applicatifs réalisés dans la cadre de cette thèse : la conception et le développement de la plateforme CALM. Cette plateforme est constituée de deux parties principales, une application client, *CALM-Mobile*, fonctionnant sous Android et destinée aux utilisateurs finaux, et une partie serveur, *CALM-RCP*, chargée notamment des calculs de proximités sémantiques et contextuelles. Ces travaux nous permettent dans la suite de valider les modèles employés ainsi que de confirmer par l'expérimentation l'intérêt d'utiliser des proximités sémantiques et contextuelles pour l'apprentissage en musée.

Afin de donner au lecteur le point de vue de l'utilisateur final sur l'application, nous commençons par présenter deux scénarios d'utilisation. L'utilisation par un visiteur seul et l'utilisation lors d'une sortie pédagogique de classe en musée. Nous présentons ensuite en détail nos réalisations techniques et les modalités de communication entre *CALM-Mobile* et *CALM-RCP*. Nous montrons la généralité de notre approche et examinons les conditions d'adaptation de cette approche à d'autres cadres applicatifs.

Les sources de connaissances utilisées ne seront que brièvement décrites dans ce chapitre, la conception de l'application étant indépendante de ces sources de connaissances. Nous présentons dans le chapitre 9 une revue détaillée de ces sources de connaissances.

8.1 Scénarios d'utilisation

Le premier scénario concerne une visite informelle, le deuxième scénario concerne l'utilisation de CALM dans le cadre d'une sortie scolaire. Dans les deux cas, le dispositif avec lequel interagissent effectivement les utilisateurs dans le musée est une tablette Android. Cependant, nous verrons dans le deuxième scénario que les phases d'avant-visite et d'après-visite, propres aux sorties pédagogiques, peuvent également être instrumentées

via l'utilisation d'un espace Web reprenant en partie les fonctionnalités de l'application mobile.

8.1.1 Utilisation par un visiteur seul

On peut séparer les activités proposées au visiteur en deux catégories : les activités de consultation et de navigation parmi les connaissances muséales (activités où le visiteur est plutôt récepteur) et les activités d'annotation et de jeux d'auto-évaluation (où le visiteur est plus actif). Le visiteur est libre de passer quand il le souhaite d'une activité à l'autre. Pour des raisons de simplicité, nous présentons tout d'abord les activités de consultation et navigation avant de présenter les activités d'auto-évaluation et d'annotation.

Afin d'illustrer le peuplement automatique des espaces contextuels durant l'utilisation de l'application, nous présentons, pour les différents écrans, l'état du contexte de l'utilisateur. Pour des raisons de lisibilité, nous ne représentons pas toutes les instances constituant les différents espaces contextuels, mais seulement celles qui sont pertinentes dans le cadre de notre scénario d'utilisation.

8.1.1.1 Navigation parmi les connaissances associées aux œuvres

Lorsque la visite débute, les différents espaces contextuels (de localisation, historique et d'intérêt) sont vides. Le visiteur se déplace dans l'espace muséal et lorsqu'il rencontre une œuvre qui l'intéresse, il peut consulter le dispositif qui lui proposera un choix parmi les différentes œuvres physiquement proches de lui (figure 8.1.1). Cette étape de choix explicite de l'œuvre est rendue nécessaire par la précision relativement faible des outils de localisation *indoor*.

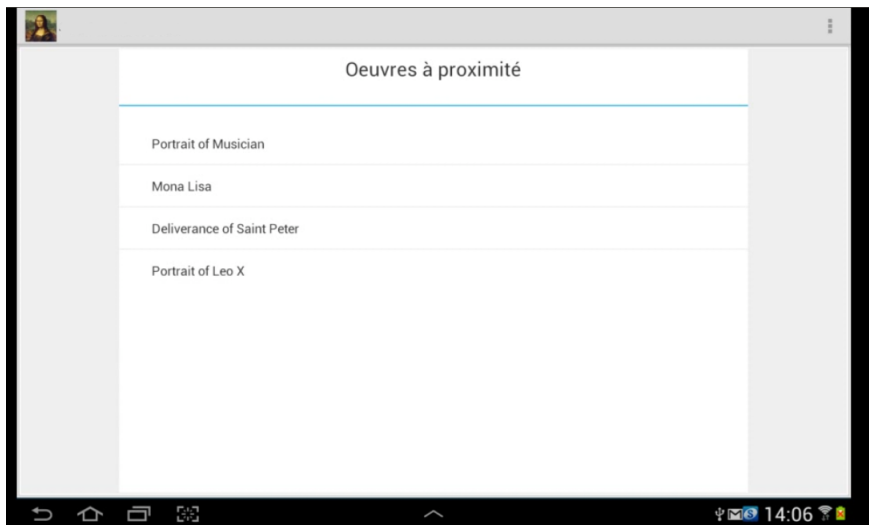
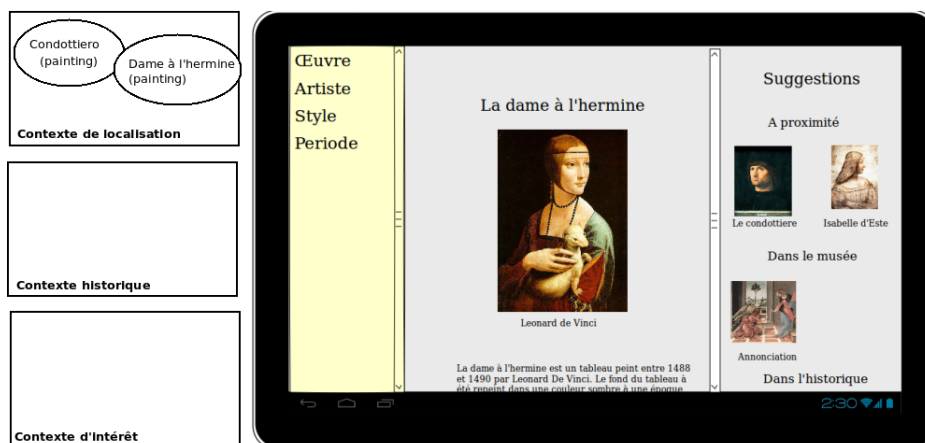


FIGURE 8.1.1: Choix de l'œuvre à considérer

L'utilisateur, ayant choisi l'œuvre sur laquelle il souhaite des informations, est dirigé vers un écran présentant la notice de l'œuvre (les informations correspondant au cartel, titre, auteur et date d'exécution). La page présente également une description plus ou moins approfondie de l'œuvre. Cette description n'est pas générée automatiquement (c'est donc la même pour tous les visiteurs). Elle correspond à ce qu'on pourra trouver dans le catalogue de l'exposition. La figure 8.1.2 présente la consultation de l'œuvre *La dame à l'hermine* de Léonard de Vinci ainsi que les informations contextuelles sur le visiteur effectuant cette consultation. Le visiteur est en début de visite, seul le contexte de localisation comporte des éléments.

FIGURE 8.1.2: Consultation de l'œuvre *La dame à l'hermine*

Cet écran de l'application comporte trois parties. La partie de gauche, sur laquelle nous

reviendrons, permet d'obtenir des renseignements complémentaires liés à l'œuvre en question, par exemple des renseignements sur l'artiste, le style où les événements décrits dans l'œuvre.

La partie centrale de l'écran comprend une image de l'œuvre permettant de confirmer immédiatement à l'utilisateur qu'il est bien en train de consulter la notice de l'œuvre choisie. Cette partie comporte une barre de défilement verticale permettant de consulter l'intégralité de la description.

Enfin, la partie droite de l'écran comporte les suggestions proposées à l'utilisateur. Ces suggestions d'œuvres sont classées suivant trois critères : les œuvres présentes dans le musée que l'utilisateur n'a pas encore vues, les œuvres à proximité immédiates de l'utilisateur et enfin les œuvres précédemment consultées par l'utilisateur.

À partir de cet écran, plusieurs possibilités d'interaction existent. Tout d'abord, une interaction implicite : si le visiteur reste suffisamment longtemps sur la notice de l'œuvre, cette œuvre est automatiquement ajoutée comme faisant partie de son contexte d'intérêt. Le visiteur peut également signifier que cette œuvre l'intéresse en passant par le menu de l'application (figure 8.1.3).

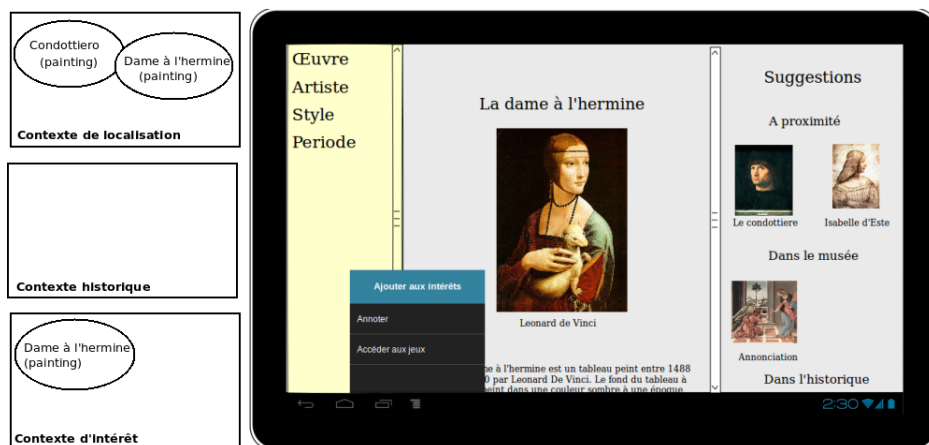


FIGURE 8.1.3: Ajout explicite d'une œuvre au contexte d'intérêt du visiteur

Il peut également consulter les œuvres qui lui sont suggérées à la droite de l'écran. La figure 8.1.4 représente la consultation de la suggestion « Le condottiere ». Lors de la consultation d'une suggestion, les justifications de cette suggestion sont exposées en langage naturel au visiteur. Ces justifications visent à aider le visiteur à établir les liens entre les différentes œuvres en utilisant de préférence des concepts ou des entités qu'il a déjà vues ou qui l'intéressent dans le choix des justifications présentées (cf. chapitre 6).

Enfin, l'utilisateur peut également consulter des informations sur l'auteur, le style, les personnages, la période ou les événements liés à l'œuvre. Pour ce faire, il utilise le menu de gauche. Si plusieurs entités du même type sont associées à l'œuvre (par exemple

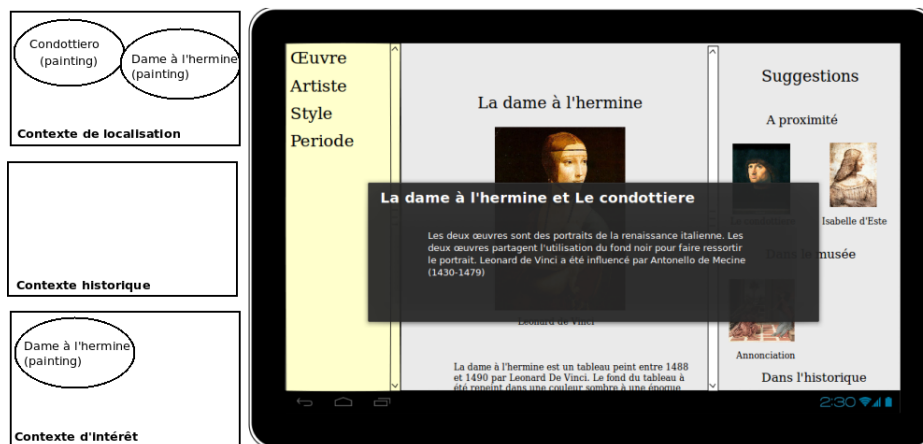


FIGURE 8.1.4: Justification d'une suggestion contextualisée

plusieurs personnages), il peut choisir l'entité pour laquelle il souhaite des informations complémentaires. La consultation des informations relatives à une entité liée à une œuvre prend exactement la même forme que la consultation d'une œuvre. Une notice décrivant l'entité lui est présentée (par exemple la biographie de l'artiste), l'utilisateur peut exprimer son intérêt pour l'entité en question et le contexte historique du visiteur est mis à jour pour refléter les entités qu'il a consultées.

La consultation des informations relatives à l'auteur de *La dame à l'hermine* est présentée dans la figure 8.1.5. L'utilisateur se voit proposer comme suggestions des artistes proches de Léonard de Vinci, ayant des peintures dans son contexte de localisation. Il se voit également suggérer des artistes dont il a précédemment consulté les œuvres ou la biographie, ainsi que d'autres artistes proches de Léonard de Vinci dont les œuvres sont présentes dans le musée. Dans tous les cas, l'utilisateur a accès à une justification de la suggestion, qui lui permet de faire le lien entre l'artiste dont il consulte la biographie et l'artiste suggéré.

8.1.1.2 Activités d'auto-évaluation et d'annotation

À tout moment, le visiteur peut accéder aux activités d'auto-évaluation ainsi qu'aux activités d'annotation et de parcours au sein de l'ensemble des annotations. Le fait de pouvoir consulter des annotations d'autres visiteurs, qui ne sont pas nécessairement experts, vise à engager le visiteur dans une relation plus personnelle avec l'œuvre, en démontrant par l'exemple que l'analyse ou la critique ne sont pas uniquement réservées aux experts. L'accès à ces activités d'auto-évaluation et d'annotation se fait *via* le menu de l'application (figure 8.1.3).

Si l'utilisateur choisit d'accéder aux annotations associées à une œuvre, il est dirigé vers

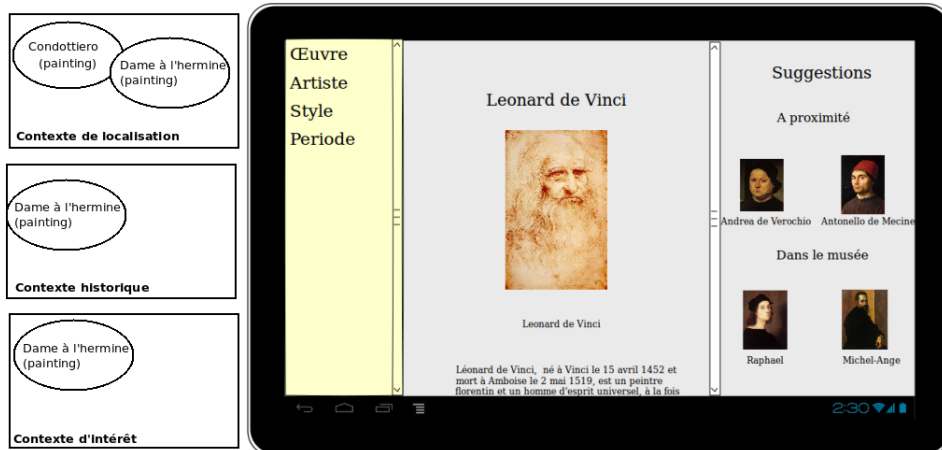


FIGURE 8.1.5: Consultation de la biographie de l'artiste associé à l'œuvre

un l'écran présenté en figure 8.1.6. Cet écran est divisé en trois parties. La partie centrale présente la liste des titres et des premiers mots des annotations liées à l'œuvre. La partie de gauche liste l'ensemble des œuvres à proximité de l'utilisateur, permettant à ce dernier de consulter les annotations associées aux autres œuvres autour de lui. Enfin, la partie droite de l'écran comprend un champ de recherche parmi l'ensemble des annotations.

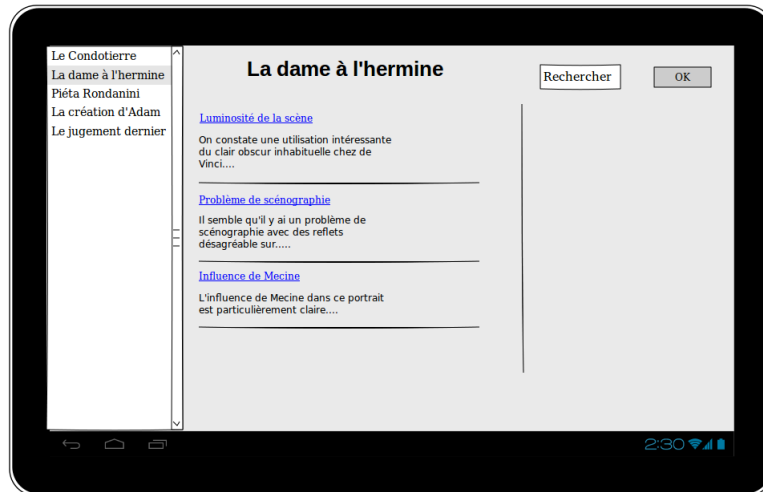


FIGURE 8.1.6: Choix des annotations à consulter

A l'inverse, l'utilisateur peut évidemment déposer une annotation sur une œuvre. Son contexte de visite est alors associé à l'annotation suivant les modalités décrites en 6.3.6. Lorsque l'utilisateur choisit de consulter une annotation particulière, il est dirigé vers l'écran de consultation présenté en figure 8.1.7. Ici encore l'écran est divisé en trois par-

ties : la partie centrale présente le texte complet de l'annotation, la partie de gauche permet de visualiser les éléments contextuels associés à l'annotation (parcours et intérêts de l'annotateur), enfin la partie de droite présente les titres d'annotations sémantiquement proches de celle qu'il est en train de visualiser (éventuellement associées à d'autres œuvres). Le visiteur dispose donc de plusieurs modes de parcours au sein des annotations, un mode classique permettant de visualiser l'ensemble des annotations associées à une œuvre et un mode plus guidé permettant de naviguer parmi le corpus d'annotations suivant des proximités sémantiques. L'objectif est ici d'inciter le visiteur à consulter des annotations qu'il n'aurait peut-être pas considérées de lui-même, mais qui peuvent cependant présenter un intérêt pour lui.



FIGURE 8.1.7: Visualisation d'une annotation

Le dernier type d'activité contextualisée que nous proposons sont les jeux d'auto-évaluation. Dans le cadre d'une visite par un visiteur libre, ces activités ont pour objectifs d'offrir une détente ludique durant la visite et de permettre au visiteur de valider ses connaissances à travers des exercices de difficulté croissante.

L'utilisateur peut accéder à tout moment à ces jeux via le menu de l'application. Quatre types de jeux lui sont alors proposés (c.f. chapitre 6) : les jeux de type QCM, les jeux de type vrai/faux, les jeux d'ordonnancement et les jeux de classification. Lorsque l'utilisateur sélectionne l'item **Accéder aux jeux** dans le menu (figure 8.1.3), il est dirigé vers l'écran correspondant aux jeux de type QCM. Une question, en adéquation avec son contexte comme décrit en 6.3.5, lui est alors proposée.

Ici encore, plusieurs choix sont offerts à l'utilisateur : il peut bien entendu répondre à la question. Dans ce cas un message l'informerait si sa réponse est correcte ou erronée, en

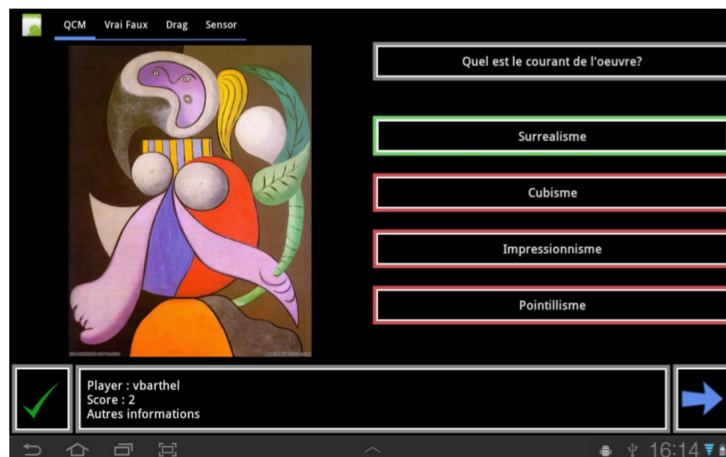


FIGURE 8.1.8: Jeu de type QCM

donnant la bonne réponse le cas échéant. L'utilisateur peut également, si la question ne lui plaît pas, charger une question différente à l'aide du bouton **Recharger** situé en bas à droite de l'écran.

Les autres jeux sont accessibles via un scrolling horizontal, assez classique dans ce type d'application. Lorsque l'utilisateur passe d'un jeu à un autre, l'état courant du jeu, c'est-à-dire la dernière question posée, est enregistré. Lors du retour sur l'écran du jeu, le visiteur retrouve donc la question qui lui avait été posée. Le visiteur peut passer d'un jeu à l'autre *via* un scrolling horizontal classique dans ce type d'application.

8.1.2 Utilisation par un groupe scolaire accompagné d'un enseignant

L'utilisation de CALM-Museum par un groupe scolaire accompagné d'un enseignant présente deux différences importantes par rapport à l'utilisation pour une visite informelle. Tout d'abord, **sur le plan organisationnel**, trois étapes sont impliquées dans une visite scolaire : la préparation de la visite, la visite en elle-même et le retour d'expérience en classe. La deuxième différence porte sur les **objectifs de visite** : la visite scolaire s'inscrit dans un programme, avec des objectifs pédagogiques clairement définis, contrairement à la visite informelle.

Concernant l'organisation de la visite, il est nécessaire d'offrir à l'enseignant des ressources pédagogiques pour l'assister durant les phases de préparation et d'après visite. Ceci dépasse donc le cadre de l'application mobile *CALM-Mobile*. Pour ce faire, nous proposons l'utilisation d'un espace Web offrant des services centrés sur la préparation de la visite et l'après-visite. Cet espace Web, reprenant pour l'essentiel les fonctionnalités de CALM, est en cours de réalisation.

Concernant la question des objectifs pédagogique de visite, nous avons présenté dans le chapitre 7 des modes de contrôle permettant à l'enseignant de spécifier une thématique de visite et des règles de contrôle contextuel. Nous détaillons dans la suite l'utilisation par l'enseignant de ces modes de contrôle.

8.1.2.1 Préparation de la visite

La préparation de la visite est une phase indispensable lors des sorties impliquant des publics scolaires. Cette préparation va permettre d'introduire les élèves aux champs lexicaux rencontrés au musée, de mettre en avant les termes, concepts et techniques pertinents vis-à-vis des objectifs pédagogiques de visite. Cette phase de préparation rend les apprenants plus réceptifs au discours muséal [Du Sablon & Racette, 1991].

Nous distinguons deux parties dans la préparation de la visite : la première partie concerne la sensibilisation des élèves aux concepts muséaux et la deuxième partie concerne la scénarisation de la visite en elle-même, par l'enseignant.

Phase de sensibilisation des élèves

Durant cette phase, l'enseignant peut accéder au site web présentant des fonctionnalités similaires à l'application mobile. Le site permet de consulter les œuvres présentes dans le musée et leurs notices ainsi que les entités liées aux œuvres, suivant les liens établis dans la base de connaissances. L'enseignant peut alors utiliser ces ressources pour proposer des informations pertinentes aux élèves avant la visite, suivant les objectifs pédagogiques qu'il a fixés. Le site permet également la génération automatique de QCM concernant certaines œuvres choisies par l'enseignant. L'enseignant peut générer ces QCM directement sous une forme imprimable, à destination des apprenants.

Phase de scénarisation

La phase de scénarisation est destinée à l'enseignant seul. L'objectif est ici de lui permettre d'intégrer des connaissances et objectifs pédagogiques qui vont modifier le comportement du système durant la visite en elle-même. Nous avons décrit pour cette scénarisation deux modes de contrôle dans le chapitre 7 : le contrôle de la thématique de visite, permettant d'inscrire l'ensemble des activités dans une thématique cohérente et le contrôle contextuel, visant à spécifier de manière précise certaines activités à effectuer durant la visite en fonction du contexte.

Concernant le contrôle thématique, le principe est de permettre à l'enseignant de sélectionner un certain nombre de ressources (ex. peintres, tableaux, événements historiques...) qui correspondent à des instances de la base de connaissances muséale. Ces ressources

seront alors mises en avant durant les phases de navigation parmi les connaissances muséales ainsi que dans la génération des jeux d'auto-évaluation (chapitre 7).

Le contrôle contextuel est basé sur un formalisme de règles contextuelles (chapitre 7). Ces règles permettent d'associer une activité spécifique à une certaine valeur du contexte du visiteur. Pour le moment, ces règles sont exprimées en SPARQL suivant le format décrit dans le chapitre 7. La création des règles requiert donc une certaine expertise, en particulier la connaissance de SPARQL, rendant nécessaire la présence d'un informaticien durant la phase de construction des règles. Nous présentons en perspectives des éléments pour une simplification de cette étape *via* un outil automatisant la génération des fichiers de règles contextuelles.

8.1.2.2 Visite du musée

L'utilisation de l'application *CALM-Museum* dans un cadre scolaire est assez similaire à son utilisation par un visiteur seul. Les principales différences sont dues aux objectifs pédagogiques définis par l'enseignant durant la phase de pré-visite. Ces objectifs permettent d'orienter, dans une certaine mesure, la visite et de conformer les activités de visite aux objectifs pédagogiques de l'enseignant.

À cette fin, les éléments suffisamment proches ou appartenant à la thématique de visite définie par l'enseignant seront mis en valeur. Par ailleurs, durant la visite, les modalités de sélection des suggestions et de génération des QCM sont également modifiées pour prendre en compte les objectifs de visite de l'enseignant (chapitre 7).

8.1.2.3 Après la visite

À l'issue de la visite, une phase de retour d'expérience en classe est nécessaire pour revenir sur les connaissances acquises au musée et les pérenniser. Ce retour d'expérience peut concerner différents aspects de la visite : retour sur le musée comme une expérience esthétique, sur les œuvres majeures ou importantes vis-à-vis du programme scolaire. De même ce retour peut prendre des formes diverses, de la discussion libre aux QCM en passant par des exposés oraux ou des rédactions.

Dans ce cadre, la capitalisation de l'expérience de visite (œuvres vues, réponses aux QCM, réponses aux questions ouvertes) peut être particulièrement intéressante tant pour l'enseignant que pour les apprenants. La mise en place d'un site offrant des espaces personnels aux apprenants ainsi qu'un accès à l'enseignant pourrait être un média efficace pour la capitalisation de l'expérience de visite. Les espaces personnels des apprenants leur permettent de revenir sur l'ensemble des œuvres et des informations qu'ils ont consultées dans le musée, ainsi qu'à leurs réponses aux questions ouvertes durant la visite. L'espace réservé à l'enseignant permet de vérifier quelles œuvres ont le plus intéressé les élèves, de consulter leurs réponses aux QCM ainsi qu'aux questions libres.

8.2 Réalisation de l'application CALM-Mobile

L'application mobile a été réalisée pour le système d'exploitation Android. Nous avons fait ce choix en nous basant sur la large majorité des dispositifs fonctionnant sur Android par rapport à IOS. Par ailleurs, d'un point de vue technique, le développement Android, entièrement en Java est plus accessible que le développement IOS, basé sur Objective-C, un langage de programmation nettement moins commun que Java. Nous n'avons pas choisi l'option d'un site web conçu pour un usage mobile car ces derniers ne permettent pas d'accéder aux fonctionnalités avancées d'Android, comme la localisation GPS.

8.2.1 L'application CALM-Mobile

L'application CALM-Mobile est l'application Android utilisée dans la visite de musée. Pour le développement de l'application CALM-Mobile, nous avons utilisé trois catégories de composants : les activités, les services et les fragments (c.f. Annexe B, Architecture d'une application Android). Les fragments correspondent à des parties d'activités, c'est à dire des parties d'interface utilisateur réutilisables d'une activité à l'autre. L'intérêt de l'utilisation des fragments est d'une part cette réutilisabilité et d'autre part la possibilité de séparer une activité en deux sous-activités (deux fragments) lorsque les propriétés d'affichage du dispositif ne permettent pas d'afficher l'écran de l'activité dans son ensemble. Ainsi, l'application peut s'adapter automatiquement à la taille de l'écran de périphérique mobile utilisé (ex. smartphone ou tablette).

Le diagramme suivant présente les différents modules de l'application Android et les liens entre ces modules.

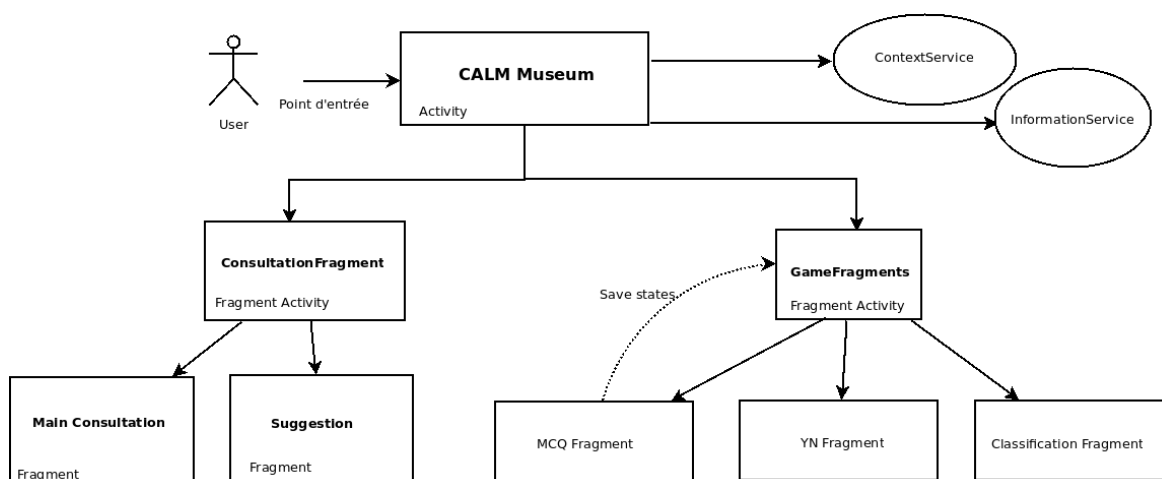


FIGURE 8.2.1: Architecture de l'application mobile

Le point d'entrée de l'application est l'activité *CALM Museum* qui va initialiser les services de communication avec le serveur : *ContexteService* pour la transmission des données contextuelles et *InformationService* pour la récupération des données affichées sur les différents écrans. Depuis l'activité *CALM Museum*, deux *FragmentActivity* sont accessibles, *ConsultationFragment* et *GameFragment*. *ConsultationFragment* activité se décompose lui même en deux *fragments*, l'un affichant le menu de droite et la partie centrale de l'application, l'autre se chargeant de la partie suggestion de l'application. De la même manière, *GameFragment* se décompose en trois fragments correspondant aux trois types de jeux que nous avons présentés.

Depuis *ConsultationFragment*, l'utilisateur peut passer *via* le menu à l'activité *GameFragment*. Lors de ce passage, l'état courant de l'activité *ConsultationFragment* est sauvegardé pour être restauré si l'utilisateur revient sur une activité de consultation. De la même manière, le jeu en cours est sauvegardé lors du passage d'un *GameFragment* à un *ConsultationFragment*.

8.2.2 Communication avec le serveur via l'API RESTLet

Nous avons choisi d'utiliser le protocole REST [Fielding, 2000] pour les communications entre notre application mobile et les applications serveur. Ce choix s'explique par la simplicité technique dans la mise en place de services REST par rapport, par exemple, aux services SOAP [Goncalves, 2010]. De plus REST permet une meilleure interopérabilité et facilite le passage à l'échelle par le couplage très faible entre les applications clients serveurs basées sur cette architecture.

Le principe de l'utilisation de l'architecture REST repose sur les notions de *ressources* et de *representations*. Une *ressource* est un objet informatique dont on peut accéder à la *representation* via une requête http. Une *ressource* peut désigner des entités très différentes ; en fait, suivant la définition proposée par Roy Fielding, « *any information that can be named can be a resource* ». La mise en place d'une application basée sur l'architecture REST impose donc tout d'abord de lister les différentes ressources utiles à l'application et de leur associer des URL. Ensuite, pour chaque type de ressource, une sémantique est associée aux opérations CRUD permises par le protocole http (GET, PUT, DELETE, POST). Nous présentons ici la méthode employée dans le cadre des activités de type jeux, l'utilisation du protocole REST étant très similaire dans les autres échanges d'information entre le client Android et le serveur.

Nous avons défini, dans l'application serveur, un ensemble d'URL correspondant aux différents types de jeux, à la consultation des suggestions ainsi qu'à la consultation des annotations :

- `http://calm.utc.fr/mcq?user`
- `http://calm.utc.fr/truefalse?user`
- `http://calm.utc.fr/classif?user`

- `http://calm.utc.fr/annotation?user?artwork`
- `http://calm.utc.fr/suggestion?user?artwork`

Respectivement pour les QCM, les jeux de types vrai-faux, les jeux de classifications, les annotations et les suggestions. Le paramètre *user* représente un identificateur unique associé à l'utilisateur au début de la visite. Le paramètre *artwork* représente l'identificateur unique d'une œuvre. La sémantique des opérations CRUD est alors directe, l'opération GET permettant de récupérer un nouveau jeu et l'opération POST d'envoyer les résultats d'un jeu donné. La figure 8.2.3 illustre la simplicité du processus de récupération d'informations *via* l'utilisation de l'API RESTlet dans l'environnement Android.

```

@Override
protected Document doInBackground(String... urls) {
    Document ret = null;
    //Récupération de la ressource via RESTLET
    try {
        ClientResource m_resource = new ClientResource(urls[0]);
        DomRepresentation m_xml = new DomRepresentation(m_resource.get());
        ret = (Document) m_xml.getDocument();
    } catch (IOException e1) {
        e1.printStackTrace();
    }
    return ret;
}

```

FIGURE 8.2.2: Récupération des informations liées à une activité

8.2.3 Problèmes rencontrés

Nous avons rencontré deux difficultés principales dans la réalisation de l'application Android : la **gestion de la géolocalisation** et la **maintenance du code applicatif** durant l'évolution très rapide du framework Android.

Contrairement à ce qu'on pourrait penser, la question de la géolocalisation n'est pas triviale. En effet si la plupart des dispositifs actuels permettent une géolocalisation en extérieur à l'aide d'un GPS, ce système ne permet pas de localiser les utilisateurs en intérieur. Deux catégories de solutions sont actuellement disponibles, les solutions de localisation **indirectes** qui requièrent une interaction de la part de l'utilisateur et les solutions **directes**, moins contraignantes, mais plus coûteuses.

Parmi les solutions indirectes, la plus courante est certainement l'utilisation de QR-Codes (figure 8.2.4) associés aux différentes œuvres du musée.



FIGURE 8.2.3: Exemple de QR-Code

Lorsque l'utilisateur scanne un QR-Code, sa position est déduite de la position du QR-Code. Cependant, cette solution nécessite d'exposer les QR-Code à côté des œuvres, ce

qui modifie de fait la mise en scène muséale et peut donc être refusé par le musée. Une deuxième solution indirecte est basée sur la reconnaissance d'image, l'idée étant de demander à l'utilisateur de photographier une œuvre, puis par des techniques de traitement d'image de reconnaître cette œuvre et donc la position de l'utilisateur. Cette solution ne demande pas de modification de la mise en scène muséale mais requiert cependant un important travail en amont de photographie des œuvres sous différents angles et différentes luminosités ainsi que des techniques complexes de traitement d'image.

Les solutions directes sont essentiellement basées sur une utilisation alternative des réseaux sans fil afin d'identifier la position de l'utilisateur. Par exemple, [Bruns *et al.*, 2007] proposent l'utilisation d'un maillage des salles du musée avec des dispositifs Bluetooth, la localisation de l'utilisateur étant déduite des dispositifs détectables depuis sa position. Une autre solution est l'utilisation de réseaux WiFi. En effet des réseaux WiFi existent dans la plupart des structures et peuvent être utilisés pour la localisation comme pour le transport de données. Dans le cadre de CALM, c'est cette dernière solution que nous avons choisie, en collaboration avec une entreprise de localisation WiFi.

Le deuxième problème que nous avons rencontré est dû à l'évolution très rapide du framework Android. Entre 2009 et 2012, pas moins de douze versions du système d'exploitation, et donc de l'environnement de développement, se sont succédées. Plusieurs changements majeurs dans les API ainsi que dans les principes de développement ont accompagné ces mises à jour. Ainsi, l'utilisation des *fragments*, particulièrement importants pour proposer une application s'adaptant à la taille de l'écran, a été introduite en 2011. De même, en terme de *design*, les recommandations ont évolué, par exemple avec la suppression du bouton physique *Menu* pour les dispositifs utilisant nativement la version 11 de l'API. Ces changements, s'ils permettent souvent de réaliser de manière plus simple et plus élégante certaines tâches, impliquent également des mises à jour fréquentes du code applicatif et un travail permanent de veille et d'apprentissage, afin d'être informé des nouvelles possibilités propres à chaque version.

8.3 Présentation des applications serveur

De la même manière que nous avons présenté l'application client, en nous plaçant tout d'abord du point de vue de l'utilisateur, nous présentons l'utilisation des applications serveur avant de fournir des éléments sur leur implémentation.

Trois applications serveur distinctes constituent CALM-*RCP*. Ces trois applications correspondent à trois besoins différents. La première application CALM-*Mapping* permet d'extraire de manière semi-automatisée des connaissances issues du web de données¹ et d'assurer un alignement entre ces connaissances et l'ontologie CIDOC-CRM que nous utilisons. La deuxième application CALM-*Gen* permet, à partir d'une ontologie et d'une

1. <http://linkeddata.org/>

base de connaissances, de définir des concepts d'intérêt et de générer l'ensemble des suggestions possibles, leur justification, ainsi que les différents jeux d'auto-évaluation que nous proposons. Enfin, l'application *CALM-Context* est chargée de la communication avec le dispositif Android et de la gestion des connaissances contextuelles en particulier du peuplement de l'historique de visite et des calculs de proximités contextuelles. La figure 8.3.1 illustre le fonctionnement global de la plateforme CALM.

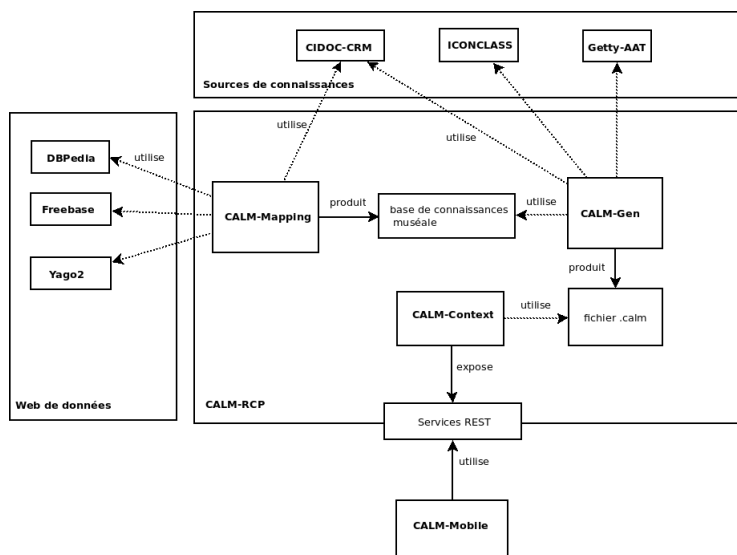


FIGURE 8.3.1: Architecture globale de la plateforme

Nous présentons ces différentes applications en nous centrant sur un objectif applicatif précis : la génération et le contrôle d'activités pédagogiques en musée. Cependant, le porté de *CALM-RCP* est plus générale, les applications étant définies indépendamment des sources de connaissances utilisées.

8.3.1 Construction de la base de connaissances muséales : l'application *CALM-Mapping*

8.3.1.1 Objectif de l'application

Afin de procéder aux expérimentations, il est nécessaire de disposer d'une base de connaissances muséales. En effet, si l'ontologie CIDOC-CRM apporte des informations structurales sur l'organisation de ces connaissances, c'est la base de connaissances qui contient les informations sur les artistes, œuvres, styles... présents dans le musée et sur les relations entre ces différents éléments.

La création de la base de connaissances muséales est donc une question importante. Pour des raisons pratiques évidentes, nous avons renoncé à créer cette base de connaissances

« à la main », en utilisant un outil comme Protégé². En effet, le temps nécessaire pour mettre en place une telle base de connaissances serait très important, rendant quasiment impossible le test de l'application sur différents jeux de données. Nous avons donc choisi d'utiliser les connaissances accessibles sur le web de données en concevant une application permettant de requêter différentes sources de données contenant des informations d'intérêt sur le patrimoine culturel. Cette application permet également d'assurer la transformation de ses connaissances sous une forme compatible avec notre version de CIDOC-CRM.

On peut cependant noter que dans le cadre d'une application industrielle, les connaissances issues du web de données ne seraient certainement pas retenues comme fiables par les experts du monde muséal (conservateur, guide-conférencier). Il faudrait alors proposer une application permettant le peuplement manuel de la base de connaissances. Une telle application est cependant hors du cadre de ce travail.

8.3.1.2 Description de l'application

La figure 8.3.1 représente l'utilisation de l'application *CALM-Mapping*.

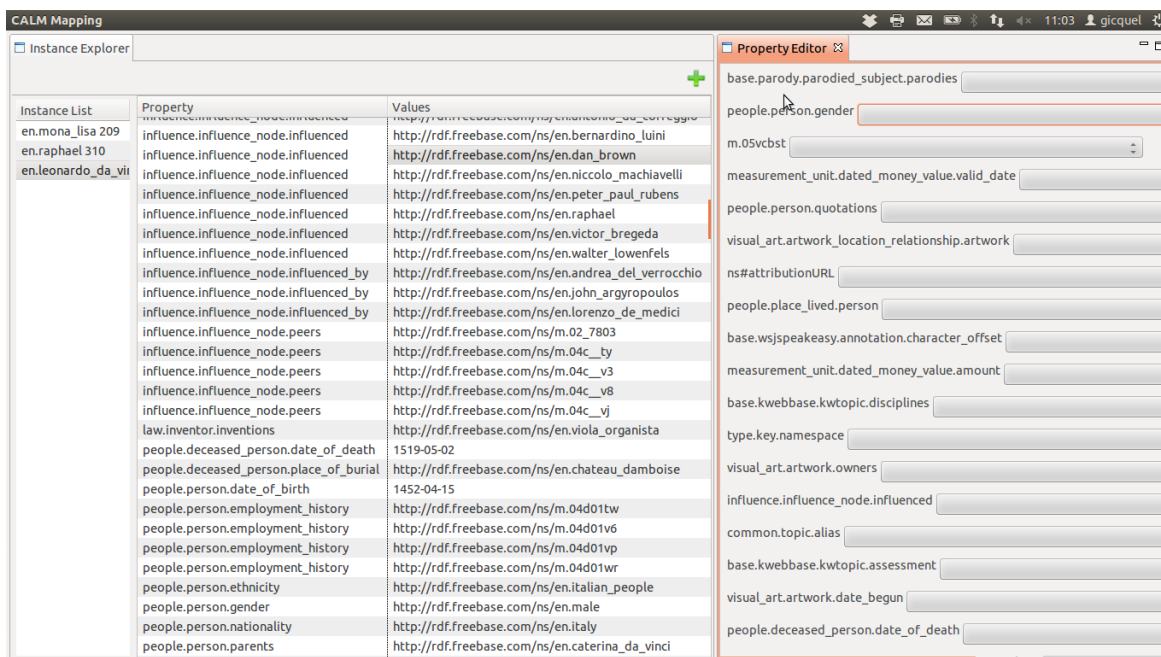


FIGURE 8.3.2: Utilisation de l'application CALM-Mapping

La partie de gauche de l'application est la liste des instances à utiliser dans la base de connaissances muséales cible (c'est-à-dire la base de connaissances qui sera utilisée par

2. <http://protege.stanford.edu/>

CALM-GEN). Ici cette liste est limitée à trois instances, représentant respectivement la Joconde, Léonard de Vinci et Raphaël. Lorsqu'une instance est sélectionnée, la partie centrale présente l'ensemble des propriétés de cette instance dans la base de connaissances source (ex. DBPedia). Enfin la partie de droite permet de réaliser un *mapping* entre les propriétés de la base de connaissances source et les propriétés de la version de CIDOC-CRM que nous utilisons. L'utilisation typique de l'application comporte trois phases : la construction de la base de connaissances initiale, la réalisation du *mapping* permettant de rendre compatible cette base de connaissances avec CIDOC-CRM et enfin l'export de la base de connaissances transformée.

La première étape de l'utilisation est basée sur l'ajout successif d'instances d'artistes, d'œuvres... qui constitueront la base de connaissances muséales. Les instances peuvent être ajoutées de **trois manières** : soit directement via le bouton ajouter, soit en se basant sur les propriétés d'une instance, soit via une requête SPARQL.

L'ajout direct est effectué via le bouton en haut à droite (croix) de la vue *Instance Explorer*. L'utilisateur peut alors saisir directement l'URI de l'instance. L'instance est alors ajoutée à la base de connaissances muséales et ses propriétés sont récupérées.

La deuxième manière d'ajouter une instance est de se baser sur les propriétés d'une autre instance. Il est en effet possible d'ajouter directement la valeur d'une propriété dans la base de connaissances (à condition que cette valeur soit une instance et pas un littéral). Par exemple, si l'utilisateur a ajouté l'instance correspondant à Léonard de Vinci, il peut ajouter les valeurs de propriétés correspondant aux œuvres de De Vinci. Ce système est assez pratique et permet de construire rapidement des bases de connaissances volumineuses, par exemple, pour Léonard de Vinci, il est très simple d'ajouter l'ensemble de ces œuvres ainsi que l'ensemble des artistes qu'il a influencés ou par lesquels il a été influencé.

La troisième façon d'ajouter une instance est classiquement de construire une requête SPARQL de type SELECT, qui va rapatrier un ensemble d'instances vérifiant les conditions de la requête ainsi que l'ensemble de leurs propriétés.

La deuxième étape consiste à déterminer un *mapping* permettant de rendre la base de connaissances compatible avec notre version de CIDOC-CRM. Par exemple, remplacer la propriété `http://rdf.freebase.com/visual_art.visual_artist.artworks` qui associe un artiste à une de ses œuvres dans la base de connaissances Freebase par la propriété `calm:created` qui associe un artiste à une œuvre dans notre version de CIDOC-CRM. La partie droite de la figure 8.3.1 illustre la construction de ce *mapping* : l'ensemble des propriétés de la base de connaissances source sont listées et, pour chaque propriété, il est possible de spécifier la propriété correspondante dans CIDOC-CRM. Le type d'une instance dans la base de connaissances cible (c'est-à-dire le concept de l'ontologie à laquelle une instance est associée) est alors déterminé par les contraintes sur les domaines et codomains des propriétés de CIDOC-CRM.

Enfin, **la troisième étape** correspond simplement à l'export de la base de connaissances

compatible avec notre version de CIDOC-CRM. Cet export est réalisé simplement *via* le menu principal de l'application.

Il est à noter que l'application *CALM-Mapping* est très générique. Elle peut utiliser comme sources de données n'importe quelle source de connaissances du web de données pourvu que la source dispose d'un *endpoint* SPARQL. De la même manière n'importe quelle ontologie peut être choisie pour l'opération de *mapping*.

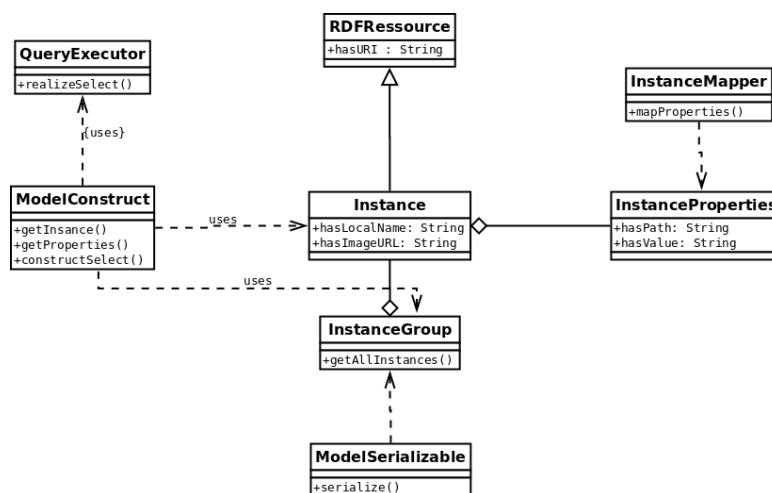
8.3.1.3 Réalisation

Nous présentons les éléments relatifs à l'interface de l'application dans l'annexe C. Afin de réaliser la partie fonctionnelle de l'application, nous nous sommes basées sur l'API Jena³. Cette API permet entre autres de créer, manipuler et sérialiser des ontologies et bases de connaissances de types RDF(S) ou OWL. L'API permet également les requêtes SPARQL sur des serveurs locaux ou distants.

L'API Jena permet la manipulation directe des instances, concepts et propriétés. Nous avons cependant choisi de construire nos propres classes pour manipuler ces représentations. En effet, nous avons des besoins spécifiques comme l'association d'une image et d'un label (ex. nom de l'artiste) aux différentes instances. Par ailleurs, l'utilisation directe des classes proposées par Jena se révèle complexe dans le cadre d'une application interactive, en particulier en raison des modes d'accès aux informations concernant les ressources d'une base de connaissances. En effet, Jena utilise de manière intensive le *design pattern* Itérateur pour l'accès aux concepts, instances et propriétés. Cependant, il n'est pas possible de réaliser des opérations sur une ressource durant une itération. Par exemple il n'est pas possible de construire un itérateur sur l'ensemble des instances du concept *Person* qui va associer à chaque instance une valeur pour la propriété *hasImage*, les opérations sur les ressources devant être réalisées hors de l'itération.

Afin de résoudre ces problèmes, nous avons choisi d'implémenter nos propres classes, représentant les instances et propriétés d'une base de connaissances : *CalmInstance* et *CalmProperty*. Le principe est qu'un objet de type *CalmInstance* est associé, via une structure de mapping, à un ensemble d'instances de *CalmProperty*, chacune d'elles comportant l'URI de la propriété et la valeur de cette propriété pour l'instance en question. Le mapping consistera alors à modifier la valeur de *CalmProperty* pour correspondre à la propriété choisie par l'utilisateur (par exemple transformer la propriété *http://dbPedia/hasAuthor* en la propriété *http://calm/artworkHasAuthor*). Par ailleurs, nous avons construit une classe de sérialisation permettant l'enregistrer les instances et propriétés sous forme d'un document RDF valide. Le diagramme de classe de la figure 8.3.2 illustre l'organisation des classes principales de *CALM-Mapping*.

3. <http://jena.apache.org/>

FIGURE 8.3.3: Principales classe de l'application *CALM-Mapping*

8.3.2 Génération des suggestions et des jeux d'auto-évaluation : CALM-Gen

8.3.2.1 Objectif de l'application

Les calculs de proximités sur des bases de connaissances sont particulièrement lents. Ceci peut entraîner un temps de réponse inacceptable si ces calculs sont effectués durant l'utilisation de l'application mobile. Nous avons donc décidé de générer en amont l'ensemble des suggestions, des justifications et des jeux d'auto-évaluation. L'application CALM-Gen a pour objectif de générer ces éléments.

Afin de fournir l'ensemble des outils pour cette génération, l'application CALM-Gen permet de définir quels sont les concepts d'intérêt de l'ontologie et de déterminer de manière automatique la matrice des concepts d'intérêt. Par ailleurs, cette application permet d'associer des labels aux propriétés ou aux compositions de propriétés pour la génération des questions en langage naturel et des justifications. Par ailleurs, CALM-Gen permet d'associer des poids aux différentes propriétés pour les calculs de proximités, la gestion des indices d'importance et d'interaction (chapitre 7) étant toujours en développement.

8.3.2.2 Description de l'application

La figure 8.3.3 présente l'interface utilisateur de CALM-Gen.

La partie gauche de l'écran représente la hiérarchie de concepts de l'ontologie utilisée (une version simplifiée de CIDOC-CRM de cette hiérarchie est présente dans la figure 8.3.3).

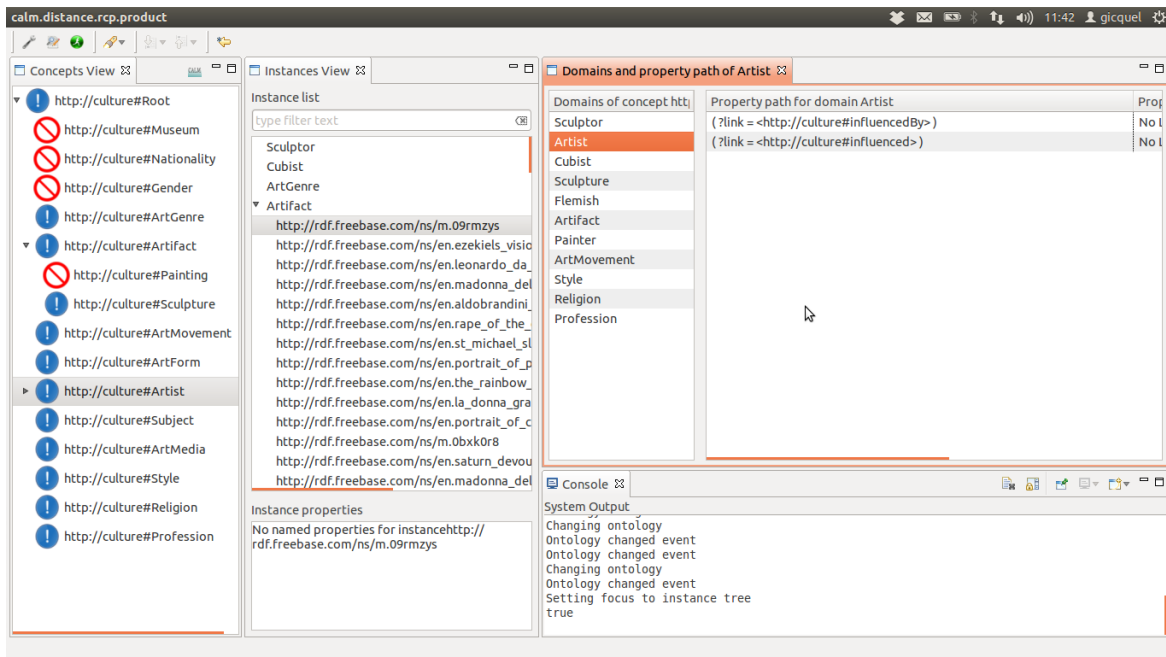


FIGURE 8.3.4: L'application CALM-Gen

On peut noter que deux types d'icône sont associés aux concepts : l'icône bleue représente un concept d'intérêt et l'icône rouge un concept qui n'est pas d'intérêt. L'utilisateur peut changer directement le statut d'un concept par un clic sur le concept de son choix.

Une fois l'ensemble des concepts d'intérêt choisis, l'utilisateur peut lancer le calcul de la matrice des concepts d'intérêt. Il est ensuite possible d'éditer, par concept, l'ensemble des propriétés d'intérêt. Cette édition se fait via l'éditeur en partie droite de la figure 8.3.3. Pour un concept donné (dans la figure le concept d'artiste), il est possible de visualiser l'ensemble des autres concepts liés directement ou indirectement (par composition de propriétés) à ce concept. Ici on constate que 7 concepts d'intérêts sont associés à artiste (de sculpteur à profession), ce qui signifie qu'une instance d'artiste peut être en relation avec 7 types d'instances différents. L'utilisateur peut alors associer un label sous forme d'un verbe à l'infinitif, ainsi qu'un score de pertinence à la propriété. C'est cette forme verbale qui permettra la génération des questions, des jeux d'auto-évaluation et des suggestions, le score de la propriété étant utilisé dans le calcul des proximités.

Une fois que les différents paramètres ont été définis, l'utilisateur peut lancer la génération des suggestions et des jeux d'auto-évaluation. Cette génération produit un fichier d'extension *.calm* qui sera chargé par *CALM-Context* lors de l'utilisation de l'application mobile.

L'application *CALM-Gen* est en grande partie générique, la génération de jeux et de suggestion pouvant se faire à partir d'une ontologie et d'une base de connaissances quel-

conque. Un aspect de l'application est cependant spécifique au domaine muséal : le calcul des proximités suivant les hiérarchies de concepts SKOS pour les propriétés *hasTheme* et *hasStyle*. D'une part, le calcul de proximités suivant une hiérarchie de concepts SKOS impose des contraintes sur la forme de la hiérarchie, d'autre part on présuppose que ces propriétés existent et ont bien pour codomaine des éléments d'une hiérarchie de concepts SKOS. Bien que cet aspect soit spécifique au monde muséal, les modifications pour adapter CALM-Gen à d'autres types de hiérarchie sont relativement simples.

8.3.2.3 Réalisation

Nous avons utilisé deux principales API pour la réalisation du coeur de l'application : l'API *Jena* que nous avons déjà évoqué et l'API *simpleNLG* permettant la génération de phrases en langage naturel (pour la génération des justifications et des questionnaires). Les sources de l'application se composent de 14 packages, pour un total de 98 classes Java. La partie fonctionnelle principale de l'application se limite à 3 packages, le package *calm.distance.compare* chargé du calcul de proximités et les packages *calm.distance.sentenceGenerator* et *calm.distance.questionGenerator* permettant la génération des questions et des justifications en langage naturel. L'annexe D présente une description plus complète de la réalisation.

8.3.3 Communication avec l'application Android : CALM-Context

8.3.3.1 Réalisation

L'application CALM-Context est une application serveur qui assure d'une part le maintien du contexte utilisateur et d'autre part la communication des données (annotations, suggestions, notices, jeux d'auto-évaluation) à l'application CALM-Mobile. Le maintien du contexte consiste à mettre à jour, en fonction des déplacements, le contexte physique de l'utilisateur, ainsi qu'à assurer les mises à jour des contextes historique et d'intérêt comme décrit dans le chapitre 5. CALM-Context est également chargée de déterminer les proximités contextuelles pour le choix des suggestions et des jeux d'auto-évaluation à proposer à l'utilisateur.

L'application nécessite deux fichiers d'entrées ainsi qu'un serveur SQL. Le premier fichier est le fichier d'extension *.calm* généré par l'application CALM-Gen ; ce fichier contient l'ensemble des questions et des recommandations possibles pour une base de connaissances donnée. Le second fichier correspond à la description physique du musée (position des œuvres) *via* l'ontologie DAIsy (chapitre 5). Le serveur SQL est chargé de la sauvegarde des annotations et de leur description sémantique et contextuelle.

Au lancement de l'application, un serveur RESTlet est mis en place sur l'ordinateur

exécutant l'application. C'est par le biais de ce serveur que sont effectuées les communications avec l'application CALM-Mobile.

Comme pour les autres applications serveur, nous avons défini des classes, alternatives à celles proposées par Jena, pour la représentation des concepts et instances de la base de connaissances.

Sur le plan fonctionnel, chaque nouvel utilisateur est associé à une base de connaissances contenant les informations sur son contexte historique, de localisation et d'intérêt. CALM-*Context* est en charge du maintien du contexte et des calculs de proximités contextuelles. Ainsi lors d'une requête de CALM-*Mobile* concernant les informations sur une œuvre par exemple, CALM-*Context* se base sur les scores de proximités sémantiques « absolus » issus du fichier généré par CALM-*Gen* et détermine les proximités contextuelles pour le choix des suggestions à transmettre. CALM-*Context* est également chargé de déterminer les proximités sémantiques et contextuelles lors des dépôts et consultations d'annotations.

Afin d'assurer la communication entre l'application mobile et CALM-*Context*, basée sur le protocole REST, nous avons mis en place pour chaque type de ressource des URL d'accès spécifique. Ces URL permettent le peuplement de la base de connaissances contextuelle ainsi que l'accès aux suggestions, jeux d'auto-évaluation et annotations (cf. 8.2.3).

8.4 Conclusion

Nous avons présenté deux scénarios d'utilisation complets basés sur notre proposition. Le premier décrit l'utilisation de CALM lors d'une visite informelle et le second son utilisation dans le cadre pédagogique d'une visite de classe. Malgré une certaine complexité tant dans la modélisation que dans la réalisation des calculs de proximités sémantiques, l'utilisation de CALM lors d'une visite informelle reste simple. Concernant le scénario de visite pédagogique, il apparaît qu'au prix d'un travail initial peut-être plus important pour l'enseignant, la construction d'une thématique de visite ainsi que la mise en place de règles de contrôle contextuel permettent une certaine souplesse dans la scénarisation de la situation d'apprentissage tout en préservant la liberté de mouvement et d'action des apprenants.

Nous avons ensuite présenté les réalisations techniques permettant l'utilisation de CALM en musée. Nous avons réalisé deux catégories d'applications : l'application client CALM-*Mobile* et trois applications serveur. Si l'application CALM-*Mobile* est spécifique à la situation de visite de musée, les applications serveur sont suffisamment génériques pour pouvoir être utilisées, moyennant quelques adaptations, dans d'autres cadres applicatifs que celui du musée.

Chapitre 9

Expérimentations et validation

La validation des systèmes d'apprentissage ubiquitaire est complexe, en particulier dans le cadre de l'apprentissage en musée. En effet, l'expérience muséale ne peut pas être réduite à l'acquisition de nouvelles connaissances mais comporte également une importante dimension esthétique et personnelle, rendant la comparaison entre différents systèmes d'apprentissage difficile. À notre connaissance, il n'existe d'ailleurs pas de méthode d'évaluation faisant consensus pour l'évaluation de l'apprentissage instrumenté par des dispositifs mobiles en musée.

Afin de disposer d'éléments d'évaluation sur notre proposition, nous avons cependant mis en place trois expérimentations correspondant à trois axes d'évaluation importants : d'une part la pertinence des informations et interactions proposées (suggestions, jeux...), d'autre part l'utilisabilité du système en situation réelle de visite de musée et enfin l'évaluation des performances du système, en particulier en cas de montée en charge (augmentation importante de la taille de la base de connaissances).

Afin de réaliser l'évaluation d'un système à base de connaissances, il est bien évidemment nécessaire de disposer d'une base de connaissances. Comme nous l'avons précisé au chapitre précédent, la base de connaissances est construite de manière semi-automatique à partir de sources de connaissances du web de données¹. Nous présentons tout d'abord les sources de connaissances disponibles ainsi que celles que nous avons utilisées et les raisons de nos choix. Nous présentons ensuite les expérimentations que nous avons menées à partir de ces sources de connaissances.

1. <http://linkeddata.org/>

9.1 Sources de connaissances utilisées

Le modèle sémantique que nous proposons (chapitre 5) n'est qu'une structure abstraite et il est nécessaire de disposer d'une base de connaissances muséales pour expérimenter l'application. La construction manuelle d'une telle base de connaissances est un travail particulièrement important. Nous avons donc cherché à automatiser la création de cette base de connaissances en utilisant les ressources du web de données, un ensemble de bases de connaissances liées entre elles et en accès libre. Nous avons identifié trois candidats principaux, qui offrent des informations d'intérêt sur les œuvres, styles, artistes et événements historiques : DBPedia, Yago2 et Freebase.

DBPedia² est la version sémantique de Wikipédia et est au centre de l'organisation du web de données : les URI définis par DBPedia servent souvent de pivot pour différentes bases de connaissances (par exemple l'URI identifiant Léonard De Vinci dans DBPedia est également utilisé dans Yago et Freebase). DBPedia est une base de connaissances de taille très importante, puisque les entités qui la composent sont directement issues (par un système de parsing) des pages Wikipédia. Cependant, après une étude plus approfondie, il s'est avéré que DBPedia n'était pas une solution viable pour constituer notre base de connaissances. Tout d'abord, la structure de la hiérarchie de concepts de DPedia est déduite de l'arborescence des pages de catégories de Wikipedia, il y a donc souvent des incohérences dans la hiérarchie de concepts. Par exemple, sur la figure 9.1.1, on voit que le concept d'appellation contrôlée d'un vin est un type de vin, ce qui est manifestement faux.

- [Food](#) ([edit](#))
 - [Beverage](#) ([edit](#))
 - [Wine](#) ([edit](#))
 - [ControlledDesignationOfOriginWine](#) ([edit](#))

FIGURE 9.1.1: Incohérence dans la hiérarchie de concepts

Par ailleurs, les propriétés des instances dans DBPedia sont extraites de manière automatique des infoboxs de Wikipedia, figure 9.1.2. Cependant, le remplissage des infobox, assuré par les utilisateurs de Wikipedia, n'est ni exhaustif, ni complètement normalisé. Les informations disponibles dans les infobox peuvent ainsi prendre plusieurs formats (par exemple pour les dates), rendant difficile le traitement automatique de ces données. Pour ces raisons, nous avons préféré ne pas nous baser sur DBPedia pour la création de la base de connaissances muséales.

YAGO2³ est une immense base de connaissances associée à une ontologie. Les connaissances présentes dans YAGO2 sont issues de DBPedia et GeoName⁴, l'ontologie

2. <http://dbpedia.org/>

3. <http://www.mpi-inf.mpg.de/yago-naga/yago/>

4. <http://www.geonames.org/>



FIGURE 9.1.2: Une infobox Wikipedia

sous-jacente est basée sur la taxonomie WordNet⁵, une classification hiérarchique des termes de langue anglaise qui fait référence par la quantité de termes traités et la qualité de sa hiérarchie (voir par exemple [Richardson *et al.*, 1994] et [Pedersen *et al.*, 2004]). YAGO2 offre donc une bien meilleure cohérence en terme de hiérarchie de concepts que DBPedia. Par ailleurs, YAGO2 assure un pré-traitement des données issues de DBPedia, permettant d'avoir une meilleure cohérence pour les valeurs des propriétés des instances (par exemple un format unique pour les dates). Cependant, nous n'avons pas choisi d'utiliser YAGO2. En effet, malgré la meilleure cohérence dans la hiérarchie de concepts et dans les valeurs des propriétés, YAGO2 est relativement pauvre en ce qui concerne la description des artistes, de leurs œuvres et des événements qui leur sont associés.

Freebase est une base de données sémantique dont l'originalité est d'être peuplée directement par les utilisateurs. Il s'agit en quelque sorte d'une version sémantique de Wikipedia, la création des instances et des liens étant assurée directement par les utilisateurs, et non par un parsing automatique de textes. Contrairement à DBPedia, les liens sémantiques entre ressources sont donc spécifiés par les utilisateurs. Freebase offre ainsi des données plus complètes et mieux structurées que DBPedia et YAGO2 (pas de multiplication des propriétés, format des données de type datatype cohérent entre les entités). Par ailleurs, Freebase est particulièrement riche dans la description sémantique des œuvres, artistes, événements historiques... Nous avons donc choisi d'utiliser Freebase pour la construction de notre base de connaissances. L'outil de création de bases de connaissances que nous avons développé, *CALM-Mapping* est décrit au chapitre 8.

5. <http://wordnet.princeton.edu/>

9.2 Evaluation des suggestions et des jeux

Un des aspects importants de notre travail est l'utilisation de proximités sémantiques pour la génération automatique de suggestions (et de leur justification) ainsi que la génération automatique de jeux d'auto-évaluation. Nous détaillons ici les expérimentations menées pour déterminer, d'une part, la pertinence des suggestions et de leur justification suivant un contexte donné et d'autre part, la pertinence des jeux d'auto-évaluation proposés, en particulier en terme d'adéquation au contexte et de difficulté.

9.2.1 Pertinence des suggestions

En raison de contraintes temporelles, nous n'avons pas pu évaluer la pertinence des suggestions en musée. Nous avons donc déterminé un processus d'évaluation réalisable par un expert depuis un poste fixe. Le principe de l'évaluation est de proposer à l'expert d'attribuer une note de pertinence à des suggestions et à leur justification.

Pour cette expérimentation, nous avons tout d'abord construit une base de connaissances comportant 37 artistes et 60 œuvres ainsi que leur description sémantique à partir des descriptions sémantiques de FreeBase. La base de connaissances étant réalisée, nous avons conçu une application sur poste fixe permettant de consulter les informations concernant les œuvres, similaire dans son fonctionnement à l'application mobile : l'expert choisit les œuvres sur lesquelles il souhaite avoir des informations et peut accéder aux informations complémentaires (auteur, style, événements historiques...).

Durant une première phase, l'expert consulte les informations relatives aux œuvres. Cette phase permet le peuplement du contexte historique et du contexte d'intérêt. Lorsque l'expert a consulté les informations correspondant à 10 entités distinctes (œuvres, auteurs...) la deuxième phase est démarrée.

Durant la deuxième phase, l'expert consulte toujours des informations sur les œuvres mais des suggestions sont cette fois présentées pour chaque consultation. Il lui est alors demandé de donner une note entre 0 et 10 à la pertinence de la suggestion ainsi qu'à la pertinence de la justification. Nous avons réalisé cette expérience avec plusieurs valeurs de pondération pour le rôle de la proximité contextuelle par rapport à la proximité sémantique. Nous avons assigné la même pondération $\frac{\alpha}{2}$ aux proximités vis-a-vis des contextes historique et d'intérêt, et la pondération β à la proximité sémantique. Le ratio des pondérations du tableau 9.1 est $\frac{\alpha}{\beta}$. Pour chaque valeur du ratio de pondération, nous avons demandé à l'expert de donner un score entre 0 et 10 à la pertinence et à la justification de 10 suggestions d'œuvres et d'artistes. Le tableau 9.1 donne la moyenne de ces évaluations.

On constate qu'en ne prenant pas du tout en compte le contexte (avec $\beta = 0$), les scores de pertinence sont relativement bons et les scores de justification sont très bons. En

Ratio des pondérations	Type de suggestion	Score de pertinence	Score de justification
0	Œuvre	7.3	8
0	Artiste	7.1	8.5
1	Œuvre	8.2	6.5
1	Artiste	8.5	5.8
2	Œuvre	6.4	3.2
2	Artiste	5.5	2.9

TABLE 9.1: Résultats de l'évaluation des suggestions

l'absence de prise en compte du contexte, les œuvres ou artistes suggérés sont les plus proches sémantiquement de l'œuvre ou l'artiste considéré. Ce sont des suggestions acceptables même en l'absence de prise en compte du parcours. Les scores des justifications sont très satisfaisants puisque les entités comparées sont effectivement proches en terme de style, de période... Il est donc possible de fournir des justifications pertinentes.

Pour un ratio de 1 (donc avec $\alpha = \beta$), on constate que le score de pertinence des suggestions est sensiblement meilleur. Cependant, le score des justifications est moins bon, les entités suggérées possèdent moins de liens avec les entités consultées. Enfin, si la pondération associée au contexte est deux fois supérieure à la pondération associée à la proximité sémantique ($\alpha = 2 \cdot \beta$), le score de pertinence baisse et le score de justification devient très faible. Il semble donc qu'une valeur proche de 1 soit souhaitable pour le ratio entre les pondérations. Ceci sera à confirmer dans nos expérimentations futures.

9.2.2 Evaluation des jeux d'auto-évaluation

Nous avons souhaité évaluer deux aspects dans les jeux d'auto-évaluation que nous proposons : la pertinence de ces jeux vis-à-vis d'un parcours donné, et l'adéquation entre la difficulté calculée et la difficulté ressentie. De même que pour les suggestions, nous n'avons pas pu, en raison de contraintes temporelles, évaluer ces aspects en situation réelle de visite de musée. Nous avons donc, de manière similaire à l'évaluation des suggestions, proposé un parcours virtuel avant de proposer des jeux relatifs à ce parcours.

L'expert était tout d'abord invité à consulter des descriptions d'œuvres, d'artistes, d'événements ou de personnages historiques. À l'issue de chaque série de cinq consultations, un jeu parmi les QCM et les jeux de type vrai-faux était proposé à l'utilisateur. Deux questions lui étaient posées : donner un score de pertinence entre 0 et 10 en fonction de son parcours et donner une évaluation de la difficulté du jeu, également entre de 0 à 10.

Afin d'évaluer la pertinence de la difficulté calculée, nous avons calculé le coefficient de corrélation de Bravais-Pearson entre la difficulté estimée et la difficulté calculée par le

système. Le tableau suivant présente la moyenne des résultats de pertinence ainsi que les scores de corrélation pour une série de 20 QCM et 20 questions de type vrai/faux.

Type de jeu	Pertinence vis-à-vis du parcours	Score de corrélation
QCM	9.3	0.21
Vrai-Faux	8.7	0.13

TABLE 9.2: Résultats des évaluations des jeux

On constate que si la pertinence des jeux est globalement bonne, la corrélation entre la difficulté calculée et la difficulté estimée est faible. Cette faiblesse peut s'expliquer par l'absence de prise en compte de la notoriété des œuvres, artistes, styles dans notre système. Par exemple, la question « Qui a peint la Joconde ? » sera élémentaire quelle que soit la proximité des distracteurs avec la réponse correcte, cette question pourra donc être considérée comme difficile (avec des distracteurs proches de « Léonard de Vinci ») par le système alors qu'elle est toujours élémentaire. De la même manière, une question portant sur un artiste peu connu du grand public sera délicate, même si les distracteurs sont éloignés de la bonne réponse. Nous décrivons dans les perspectives (chapitre 10) une méthode permettant de prendre en compte la notoriété d'une œuvre.

9.3 Evaluation de l'utilisabilité de l'application mobile

Nous avons effectué une première évaluation de l'utilisabilité du système en situation réelle de visite de musée. Cette évaluation a impliqué 37 visiteurs et s'est déroulée dans la Grande Galerie du Palais Impérial de Compiègne en avril 2011. La version de l'application qui a été évaluée est une version antérieure au prototype actuel ne proposant pas d'activités de jeux. Sept œuvres d'intérêt ont été sélectionnées pour cette expérimentation, ce nombre réduit s'expliquant par la contrainte que nous avons de fournir des descriptions très précises des œuvres, artistes suivant un processus de validation assez contraignant (validation par le conservateur, la chargée de communication ainsi que le directeur du musée lui-même).

Nous avons suivi une approche classique d'évaluation de l'utilisabilité des interfaces, inspirée de la méthode *System Usability Scale* [Brooke, 1996]. Les visiteurs étaient invités à réaliser quatre types de tâche : consulter la notice d'une œuvre, consulter les informations annexes sur l'œuvre, lire les annotations associées à une œuvre et déposer des annotations sur une œuvre.

À l'issue de l'expérimentation, un questionnaire permettait au visiteur d'indiquer sur une échelle de 1 à 4 (de *absolument pas d'accord* à *tout à fait d'accord*) (annexe **E**) son accord avec différentes propositions visant à évaluer l'utilisabilité et l'utilité des tâches proposées. Pour obtenir des résultats entre 0 et 100, nous avons associé un score de 100

aux réponses de type tout à fait d'accord pour l'utilisabilité ou l'utilité d'une tâche, un score de 66 aux réponses de type "assez d'accord", etc. Le tableau 9.3 synthétise les moyennes des scores d'utilité et d'utilisabilité des différentes tâches.

Tâche	Utilisabilité	Utilité
Consultation de la notice	83	97
Consultation des informations annexes	78	86
Consultation d'annotation	73	60
Dépôt d'annotation	63	52

TABLE 9.3: Résultats des évaluations d'utilisabilité

Le tableau 9.3 amène plusieurs commentaires. Tout d'abord, la perception de l'intérêt et de l'utilisabilité de l'application est globalement bonne, les interactions contextuelles (utilisation de la localisation) semblent effectivement faciliter l'accès aux ressources culturelles. Ce résultat encourageant doit cependant être considéré avec prudence car la présence des expérimentateurs peut causer un biais dans les réponses données par les utilisateurs. L'utilisation d'annotations sur les œuvres est l'activité perçue comme la moins utile et la plus difficile. Ceci peut être dû en partie à l'utilisation du clavier virtuel Android pour entrer le texte des annotations, ce clavier étant en effet assez peu pratique.

Les visiteurs étaient relativement homogènes en terme de classes d'âge, ce n'était pas le cas du niveau d'étude. Nous n'avons cependant pas constaté de différence significative dans l'utilisabilité de l'application en fonction du niveau d'étude. Ce fait peut s'expliquer par la banalisation de l'utilisation d'outils mobiles.

Enfin, plusieurs visiteurs ont exprimé le souhait de pouvoir disposer d'un "niveau de confiance" sur les annotations qu'ils consultent afin d'estimer leur fiabilité.

9.4 Evaluation des performances

La question du temps de réponse est cruciale dans le cadre des systèmes interactifs, en particulier si ces systèmes sont destinés à être utilisés par le grand public en mobilité. Il est indispensable que le temps de réponse du système soit quasiment instantané. Notre travail est basé sur des calculs de proximités sémantiques, pouvant impliquer des opérations nombreuses et complexes pour déterminer la proximité entre deux éléments de la base de connaissances. Nous avons conduit un ensemble de tests de performances à la montée en charge du système, inspiré de la méthodologie LUBM [Guo *et al.*, 2005]. Ces tests nous ont amené à revoir la conception du système, initialement pensé pour des proximités sémantiques calculées en temps réel, pour choisir une méthode basée sur un calcul en amont (cf. 8.3.2).

9.4.1 Méthodes et résultats de l'évaluation des performances

En négligeant la durée des échanges réseaux et de l'affichage des informations sur les terminaux utilisateurs, le principal facteur impactant le temps de réponse de notre système est le calcul des proximités sémantiques. Le temps de réponse de ce calcul est directement corrélé à la taille de la base de connaissances (i.e. le nombre d'instances).

Afin d'évaluer les performances des systèmes à base de connaissances, [Guo *et al.*, 2005] proposent le framework de test LUBM. Ces tests fournissent une métrique d'évaluation suivant plusieurs facteurs (ex. nombre de classes, inférences réalisées, etc.) et plusieurs critères (ex. temps de chargement de la base de connaissances, taille en mémoire, temps de réponse à une requête SPARQL, etc.). Nous nous sommes inspiré de cette approche pour réaliser une évaluation des temps de réponse de notre système lors de la montée en charge, c'est-à-dire lors de l'augmentation du nombre d'instances dans la base de connaissances.

Nous avons testé le système sur deux types d'opération. La première est une simple comparaison de la valeur prise par une propriété d'une instance par rapport à celle de toutes les autres instances (ce qui est une étape du calcul de proximité, chapitre 6). La deuxième est la génération de l'ensemble des questions possibles à partir d'une instance.

En nous inspirant du protocole LUBM, nous avons alors ajouté successivement des groupes de 100 instances dans notre base de connaissances et effectué les opérations de comparaison et de génération de questions. Les courbes de la figure 9.4.1 illustrent le temps de calcul nécessaire en fonction du nombre d'instances pour une comparaison et pour la génération de questions. On constate que pour une comparaison le temps d'exécution est faible et progresse de manière quasiment linéaire (courbe de gauche). Pour la génération de questions, le temps d'exécution est non seulement beaucoup plus long (jusqu'à 160 secondes pour 3000 instances) et semble de plus progresser de manière non linéaire.

Une augmentation du nombre d'instances risque donc d'augmenter de manière non linéaire le temps de réponse du système et d'engendrer des temps de réponses inacceptables pour un système interactif.

9.4.2 Prise en compte des résultats dans la conception du système

Nous avons identifié le problème de la croissance non linéaire du temps de réponse assez tôt dans le développement. Ceci nous a amené à choisir une méthode alternative à la génération dynamique des suggestions et des jeux d'auto-évaluation. Cette méthode consiste tout simplement à calculer l'ensemble des proximités entre instances et à générer les suggestions et les auto-évaluations avant l'utilisation du système. Nous avons décrit dans le chapitre précédent l'application CALM-Gen qui permet cette génération en amont.

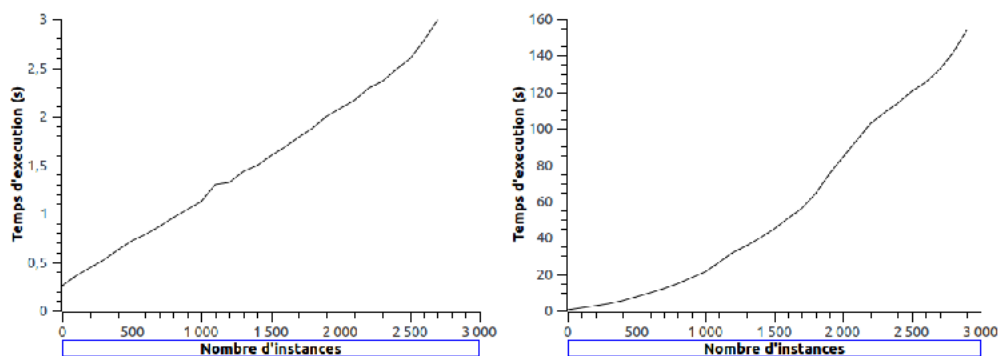


FIGURE 9.4.1: Temps de réponse pour les requêtes simples (gauche) et la génération de question (droite)

Le choix de générer en amont des questions et suggestions, s'il constitue une solution moins élégante qu'une génération *on the fly*, ne modifie aucunement le système et n'impose pas de limite particulière. En effet, la base de connaissances muséales étant peu susceptible d'évoluer rapidement, il suffira de recommencer cette génération à chaque modification de la base de connaissances.

9.5 Conclusion

Nous avons décrit dans ce chapitre les expérimentations menées pour valider le fonctionnement du système. Les aspects les plus importants sont la pertinence des suggestions et de leurs justifications, ainsi que la pertinence des jeux d'auto-évaluation proposés. Les résultats obtenus sont globalement satisfaisants et confirment la validité de l'utilisation de proximités sémantiques et contextuelles dans le cadre d'un environnement d'apprentissage ubiquitaire.

Cependant, nous n'avons pas pu tester le fonctionnement complet du système en situation de visite de musée, ni l'utilisation dans le cadre d'une visite pédagogique de classe. Ces éléments seront creusés dans des travaux à venir qui nous permettront également de nous pencher sur la question de la scénarisation de la visite, un aspect narratif pouvant en effet s'avérer intéressant en particulier dans le cas de jeunes visiteurs (dont l'attention est mieux retenue par des « histoires »).

Chapitre 10

Conclusion et perspectives

L'informatique mobile joue un rôle de plus en plus important dans notre vie de tous les jours et il est probable que cette tendance va aller en progressant. Il est intéressant de noter que les concepts d'informatique ubiquitaire ou de calm-computing, proposés par Weiser en 1981, sont aujourd'hui quasiment devenus la norme. Par exemple, n'importe quel smartphone permet de trouver en quelques secondes le commerce le plus proche via l'utilisation du GPS et l'interaction avec des services web types Google Map.

Si l'informatique ubiquitaire a bousculé notre vie quotidienne, elle n'a pas encore complètement investi le champ des EIAH. Pourtant, il semble que, en particulier lors d'apprentissages sur le terrain, les dispositifs ubiquitaires offrent des possibilités importantes pour suggérer des indications au bon endroit et au bon moment, ainsi que pour tracer et analyser l'activité de l'apprenant. Un frein possible au développement de l'informatique ubiquitaire pour l'apprentissage est la question du positionnement pédagogique. L'informatique ubiquitaire est en effet située à l'intersection entre apprentissage formel et apprentissage informel, à la frontière entre formation théorique et sur le terrain... Il peut donc être difficile de positionner clairement un EIAH ubiquitaire du point de vue pédagogique; la théorisation pédagogique de l'apprentissage ubiquitaire est encore en construction. Outre cette difficulté théorique, il existe des contraintes financières fortes qui peuvent freiner l'utilisation de l'informatique ubiquitaire pour l'apprentissage. L'obsolescence rapide du matériel informatique est bien connue et cette obsolescence est décuplée dans le cas des dispositifs mobiles.

Malgré ces limites, il semble indiscutable que les technologies de formations ubiquitaires seront de plus en plus présentes. A travers ce travail, nous avons montré que les techniques et outils de gestion des connaissances formalisées pouvaient avec profit être utilisées dans le cadre de situations d'apprentissage ubiquitaire. Nous avons présenté une approche, basée sur des technologies sémantiques, permettant l'utilisation de l'informatique ubiquitaire dans un cadre d'apprentissage informel. Si plusieurs travaux ont été proposés dans le cadre de l'apprentissage en musée, nous avons définis une représentation

sémantique et contextuelle originale des connaissances muséales. Par ailleurs nous avons défini des modes d'opérationnalisation originaux de ces représentations sémantiques à travers les proximités sémantiques et contextuelles. Nous avons décrit cette opérationnalisation suivant plusieurs axes, allant de la génération automatique d'activités pédagogiques au contrôle d'une situation d'apprentissage en musée par un enseignant. Si les modèles du domaine et du contexte que nous avons proposés sont propres au monde du musée, l'opérationnalisation de ces modèles par des calculs de proximités sémantiques et contextuelles sont largement indépendants du cadre musée et on peut envisager d'autres cadres applicatifs.

10.1 Rappel des objectifs

Nous avons décrit dans l'introduction notre problématique et les objectifs de ce travail. L'objectif théorique principal était d'organiser une modélisation duale des connaissances, entre les connaissances purement contextuelles et les connaissances propres à un domaine d'application, la difficulté étant que les connaissances propres à un domaine sont souvent « statiques » alors que les connaissances contextuelles sont dynamiques. Notre objectif était d'articuler ces deux catégories de connaissances afin de proposer aux utilisateurs des informations cohérentes tant sur le plan du domaine d'apprentissage que sur le plan contextuel.

Le second objectif était plus centré sur l'aspect applicatif de notre travail. Il s'agissait de déterminer comment générer des propositions d'interactions pertinentes lors de visites de musées, que ces visites soient informelles ou scolaires. Cet objectif implique directement d'offrir des possibilités de contrôle pédagogique aux enseignants, afin d'assurer un apprentissage efficace.

10.2 Apports de la thèse

Nous présentons les apports de nos travaux suivant trois axes principaux, qui ont été nos préoccupations majeures durant ce travail : la modélisation sémantique du patrimoine et du contexte, les calculs de proximités sémantiques et contextuelles pour la proposition d'interactions contextualisées et enfin la définition de modes de contrôle du système.

10.2.1 Modélisation du patrimoine et du contexte

Nous avons proposé un modèle issu de CIDOC-CRM, l'ontologie de référence pour la modélisation du patrimoine culturel. Plusieurs transformations ont dû être apportées à l'ontologie CIDOC-CRM. Trois raisons ont mené à ces transformations : le manque

de vocabulaire métier lié à l'art dans CIDOC-CRM, la difficulté d'intégration des connaissances issues du web de données à CIDOC-CRM et enfin le caractère complexe de CIDOC-CRM rendant les proximités sémantiques moins naturelles pour un public non-expert.

Les modifications apportées à CIDOC-CRM reposent sur l'ajout de deux hiérarchies de concepts décrivant les thèmes des oeuvres d'art (ICONCLASS) ainsi que leurs styles (Getty-AAT), ainsi que sur la création de propriétés par composition, permettant par exemple d'associer directement un artiste à la date de sa mort. Nous avons établi un mapping de la base de connaissances Freebase vers notre modélisation de CIDOC-CRM, ce mapping nous permettant de tester notre proposition sur des jeux de données divers et de taille relativement conséquente (plusieurs centaines d'instances).

Afin de lier les connaissances patrimoniales au contexte des utilisateurs, nous avons défini un modèle du contexte suivant différents espaces contextuels, chaque espace contextuel étant structuré par une ontologie (ex. ontologie de localisation pour l'espace contextuel de localisation). Ces espaces nous permettent de faire le lien entre les connaissances purement contextuelles et les connaissances issues de la modélisation du domaine. Par exemple de répondre à la question « Y-a-t'il dans l'historique récent de l'utilisateur des peintures de la Renaissance italienne ? ».

10.2.2 Proximités sémantiques et contextuelles pour l'interaction contextualisée

L'utilisation de proximités sémantiques et contextuelles pour l'interaction située constitue le cœur de ce travail. Si les modélisation du patrimoine et du contexte étaient nécessairement domaine dépendantes, nous avons proposé un mode de calcul indépendant du domaine pour les proximités sémantiques et contextuelles.

Concernant les proximités sémantiques, nous nous sommes basé sur une approche générique de la proximité entre deux objets vues comme la proximité entre les propriétés que prennent les valeurs des propriétés de ces objets. Cela nous a amené à définir plusieurs types de calculs en fonction de la nature de la valeur des propriétés utiles à la comparaison, c'est à dire la nature du codomaine de ces propriétés. Nous avons ainsi proposé différents calculs suivant que les propriétés aient pour codomaines des instances organisées hiérarchiquement (comme le style ou le thème), des instances non organisées hiérarchiquement ou des littéraux. Ces calculs nous ont amené à représenter la proximité sémantique entre deux instances comme un vecteur où chaque composante représentait la proximité vis-à-vis d'une propriété. Ces calculs de proximité nous permettent de proposer diverses interactions situées en fonction de la catégories d'élément considérée par l'utilisateur sur le dispositif mobile. Tout d'abord la proposition d'élément de même nature proche sémantiquement (par exemple des artistes de la même période si l'élément consulté est une biographie d'artiste), ensuite la proposition de différents types de jeux d'auto-évaluation

et enfin la possibilité d'annoter les œuvres et de retrouver les annotations sémantiquement proches les unes des autres.

Concernant les proximités contextuelles, nous nous sommes également basé sur une approche indépendante du domaine d'application. Les proximités contextuelles permettent de déterminer dans quelle mesure une instance ou un ensemble d'instances est présent dans les différents espaces contextuels de l'utilisateur. Nous avons défini plusieurs modes de calcul de proximités contextuelles en fonction de la nature de l'espace contextuel considéré. Une comparaison basée sur la relaxation et l'intersection d'ensembles d'instances pour le contexte de localisation, une comparaison basée sur une focalisation temporelle pour la proximité basée sur l'historique de visite, ainsi qu'une comparaison basée sur les liens existant entre deux ensembles d'instances pour la prise en compte du mode de visite du visiteur. Ces proximités contextuelles nous permettent de contrôler de manière fine l'impact du contexte dans les différentes activités proposées, par exemple de garder une cohérence dans la localisation en suggérant des œuvres impressionnistes si l'utilisateur est dans une salle contenant essentiellement des œuvres impressionnistes.

10.2.3 Modes de contrôle du système

Nous avons défini plusieurs modes de contrôle, destinés aux différents utilisateurs du système.

Concernant l'utilisation du système pour un apprentissage lors d'une sortie scolaire en musée, nous avons proposé plusieurs possibilités permettant à l'enseignant d'intégrer ses objectifs pédagogiques dans le fonctionnement du système. Nous proposons en particulier un contrôle thématique visant à orienter l'ensemble des activités de visite suivant des axes thématiques définis par l'enseignant. Afin d'avoir un contrôle plus direct, nous proposons également des règles de contrôle contextuel qui permettent spécifiquement d'associer certaines activités (ex. jeux, questions ouvertes), à certaines œuvres ou à certains états du contexte.

Nous avons également proposé un mode de contrôle spécifiquement dédié aux experts chargés de la mise en place du système. L'idée étant ici de permettre aux experts d'exprimer leurs préférences vis-à-vis de l'importance des diverses propriétés dans le calcul des proximités. Nous nous sommes basé pour cela sur l'utilisation de l'opérateur d'agrégation de Choquet qui permet de contrôler de manière fine les poids des propriétés et les interactions entre ces propriétés. Par exemple, de spécifier que deux propriétés sont complémentaires et donc que si deux œuvres sont proches vis-à-vis de ces propriétés, elles auront un score de proximité important. L'agrégation de Choquet demande de fixer de nombreux paramètres, pour un nombre de propriétés n l'ordre de grandeur du nombre de paramètres est $n^2 + n$. Nous avons présenté des éléments permettant de fixer de manière semi-automatique ces paramètres en fonction de l'ontologie et de la base de connaissances considérée.

10.3 Perspectives

10.3.1 Perspectives à court terme

Une limite du travail que nous avons présenté est l'évaluation de l'apprentissage permis par notre système, que cet apprentissage concerne la visite informelle ou la sortie pédagogique au musée.

Nous prévoyons donc de définir un protocole expérimental pour l'évaluation du système. Ce protocole devra prendre en compte les différents aspects de la visite : non seulement l'acquisition de nouvelles connaissances mais également contrôler que le plaisir esthétique de la visite n'est pas dégradé par l'application. Pour cela nous nous baserons sur des entretiens avec des visiteurs ayant utilisé le système que nous proposons, des visiteurs n'utilisant pas le système et des visiteurs en visite accompagnée par un guide-conférencier. Nous chercherons à déterminer l'impact du système sur le rappel des connaissances présentées, ainsi que sur les méta-connaissances muséales qui peuvent également être retenues (ex. suivant quels axes comparer deux œuvres ?). Nous chercherons également à déterminer si les visiteurs ayant utilisé le système seront plus enclins que les autres à effectuer d'autres visites ou à se renseigner sur des points qui ont été exposés dans le musée.

Concernant les groupes scolaires, nous prévoyons de mettre en place une expérimentation impliquant des enseignants pour la définition de thématiques de visites et de règles de contrôle contextuel. Ces règles étant définies, nous pourrions procéder à une expérimentation permettant de valider le gain en autonomie des apprenants par l'utilisation du système ainsi que l'acquisition de nouvelles connaissances. Nous pensons qu'en particulier le fait de répondre à des questions ouvertes pendant la visite de musée et donc l'utilisation de connaissances fraîchement acquises dans la construction d'un argumentaire, même réduit, devrait pérenniser l'acquisition de ces nouvelles connaissances.

Sur le plan applicatif, nous avons également une perspective à court terme : définir un processus permettant de juger de la « notoriété » d'une œuvre, d'un artiste, d'un style...

Un processus qui permet d'évaluer la « notoriété » d'une œuvre serait en effet intéressant à utiliser tant dans les suggestions que dans les jeux d'auto-évaluation. On pourrait ainsi définir des œuvres « immanquables », correspondant à celles qui ont le score de notoriété le plus important. Par exemple, une œuvre comme La Joconde devrait avoir un score de notoriété très élevé. Dans le cadre des jeux d'auto-évaluation, ce type de score permettrait d'affiner la difficulté des jeux : les jeux impliquants des œuvres à la notoriété importante seront plus faciles que des jeux impliquants des œuvres peu connues. Afin de déterminer un processus numérique permettant de donner un score d'importance aux différentes œuvres, artistes, etc. nous pourrions nous baser sur la structure de la base de connaissances, à la manière de notre calcul de proximité. Les instances entretenant des relations avec des instances dont la notoriété est forte auront une notoriété forte,

les instances entretenant peu de relations ou des relations avec des instances dont la notoriété est faible auront une notoriété faible.

10.3.2 Perspectives à moyen et long terme

Un aspect prometteur, qui n'a été qu'évoqué dans le chapitre 8, est la création d'espaces web personnels tant pour une visite informelle que pour une visite scolaire. Dans une perspective d'application du système à différents lieux culturels, ces espaces pourraient constituer une mémoire culturelle personnelle, permettant d'une part de revenir sur les objets, lieux, documents et d'autre part de présenter des rapprochements entre ces différents objets culturels au moyen des proximités sémantiques. Cette question d'un portfolio culturel personnel, qui s'alimenterait tout au long de la vie, permettrait de créer des liens forts entre les individus et le patrimoine culturel. Dans une perspective plus modeste, la création d'un portfolio éducatif personnel pour les élèves de l'enseignement primaire et secondaire pourrait s'avérer également intéressante en permettant d'offrir un accès numérique à des connaissances variées propres aux différentes matières enseignées et d'opérer les rapprochements entre ces connaissances. Un exemple de ce type de rapprochement est l'utilisation de fables de La Fontaine pour illustrer des vases ayant appartenu à Napoléon III et présents au Palais Impérial de Compiègne : ces vases sont des objets physique en lien à la fois avec les fables de La Fontaine étudiées au primaire et l'Empire étudié dans le secondaire. Par ailleurs, via des fonctions de communication, un tel portfolio permettrait aux apprenants de comparer différents regards ou commentaires sur les œuvres, offrant ainsi un aspect social propre à engager des relations personnelles entre les apprenants et les œuvres.

Le deuxième aspect que nous souhaitons creuser concerne la scénarisation de la visite. En effet nous n'avons présenté pour le moment que des éléments permettant d'orienter le parcours et de proposer certaines activités à certains moments (règles de contrôle contextuel). Il serait intéressant de proposer des éléments plus narratifs, particulièrement envers le public le plus jeune (école primaire), très sensible à la narration. Ces éléments narratifs devraient s'adapter de manière automatique à la visite et délivrer des informations culturelles en parallèle d'une « histoire » propre à captiver l'attention des plus jeunes visiteurs. De manière plus générale, si dans un musée la scénarisation est en quelque sorte implicite, tout simplement par l'agencement des salles, afin d'adapter le système à d'autres situations d'apprentissage nous devons pouvoir proposer aux enseignants de réaliser des scénarios précis. Ces scénarios devront par exemple spécifier qu'un ensemble de tâches doit être effectué, chaque tâche possédant des préconditions et postconditions. Afin de proposer une telle scénarisation, nous nous intéresserons aux outils tels que IMS-LD en essayant de déterminer une approche mixte, basée sur une scénarisation dynamique prenant en compte l'état actuel du scénario mais également les proximités sémantiques et contextuelles afin de déterminer les tâches à effectuer.

Enfin, nous souhaitons déterminer dans quelle mesure l'approche que nous proposons peut être appliquée à d'autres terrains applicatifs que le musée. En effet, les calculs de proximités que nous proposons sont génériques et il est possible de les utiliser sur d'autres types de connaissances formalisées. L'intérêt et la limite de l'apprentissage en musée est de pouvoir s'affranchir en partie d'un scénario d'apprentissage, cependant, en définissant un modèle du scénario nous pourrions appliquer nos résultats à des situations d'apprentissages plus formelles (ex. apprentissage sur le lieu de travail). Dans ce cadre, nous pensons en particulier que l'utilisation des modes de contrôle que nous avons proposés (contrôle thématique et contextuel ainsi que contrôle des proximités avec l'intégrale de Choquet) pourrait être particulièrement intéressante.

Annexes

Annexe A

L'intégrale de Choquet comme opérateur d'agrégation

Nous avons présenté au chapitre 7 l'utilisation de l'intégrale de Choquet comme alternative à l'utilisation de la moyenne pondérée pour l'agrégation du vecteur de proximités. Nous décrivons ici de manière plus précise les éléments théoriques et techniques sur l'intégrale de Choquet comme opérateur d'agrégation.

A.1 Définition (rappel)

Un opérateur d'agrégation est une fonction $Agreg(x_1 \dots x_n), [0, 1]^n \rightarrow [0, 1]$ vérifiant les conditions suivantes :

- $\forall x \in [0, 1], Agreg(x \dots x) = x$
- $Agreg(0 \dots 0) = 0$ et $Agreg(1 \dots 1) = 1$
- $x_k \leq y \Rightarrow Agreg(x_1 \dots x_k \dots x_n) \leq Agreg(x_1 \dots y \dots x_n)$

Les opérateurs d'agrégation sont particulièrement utilisés en aide multicritère à la décision (AMD). Une question typique en AMD est de comparer plusieurs solutions à un problème, chaque solution étant définie par un ensemble de valeurs de satisfaction dans l'intervalle $[0, 1]$ suivant un ensemble N de critères. Le score de satisfaction d'une solution S peut alors s'exprimer sous la forme : $\{x_1 \dots x_n\}$ avec chaque x_i correspondant à la valeur de satisfaction pour le critère $i \in N$. Il est difficile de comparer deux vecteurs de $[0, 1]^n$ entre eux pour déterminer quelle est la meilleure solution. Les opérateurs d'agrégations permettent de s'affranchir de cette difficulté en ramenant un vecteur de $[0, 1]^n$ à un réel dans $[0, 1]$. La meilleure solution étant alors celle obtenant le « score » le plus proche de 1.

Nous nous intéressons ici à un opérateur d'agrégation particulier : l'intégrale de Choquet. La particularité de cet opérateur est de prendre en compte les interactions qui peuvent

exister entre les critères. Plusieurs critères peuvent en effet être complémentaires, c'est-à-dire que le score global d'un objet satisfaisant sur chacun de ces critères sera supérieur à la somme des scores d'objets, chacun satisfaisant sur un des critères. De la même manière, plusieurs critères peuvent être redondants, c'est-à-dire qu'un objet satisfaisant sur l'ensemble de ces critères aura un score inférieur à la somme des scores d'objets, chacun satisfaisant un des critères. L'idée fondamentale de l'intégrale de Choquet comme opérateur d'agrégation est alors d'associer des poids non seulement à chaque critère d'agrégation mais, également à chaque ensemble de critères d'agrégation.

L'intégrale de Choquet est basée sur la notion de mesure non-additive ou mesure floue. Une mesure floue est représentée par une fonction appelée capacité. Soit $N = \{1 \dots n\}$ un ensemble des critères d'agrégations, une capacité sur N est une fonction $\nu, \mathcal{P}(N) \rightarrow [0, 1]$ telle que :

$$\begin{aligned} \nu(N) &= 1 \\ \nu(\emptyset) &= 0 \\ \forall i \in N, \forall T \subset P(N), \nu(T \cup i) &\geq \nu(T) \end{aligned}$$

Les deux premières conditions sont simplement des conditions aux bornes, la dernière condition est la monotonie par rapport à l'inclusion : l'ajout d'un nouveau critère ne peut pas faire décroître l'importance d'un groupe.

Le score d'un objet X ayant pour valeurs $\{X_1 \dots X_n\}$ sur les critères $N = \{1 \dots N\}$ est donné par :

$$C_\nu(X) = \sum_{i=1}^n X_{(i)} [\nu(A_i) - \nu(A_{i+1})]$$

Avec $(.)$ une permutation de N telle que $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)}$, $A_i = \{(i) \dots (n)\}$ et $A_{n+1} = 0$.

A.2 Caractérisation : indices d'importance et d'interaction, k-additivité

Afin de caractériser une capacité, deux indices classiques sont utilisés, les indices d'importance et d'interaction.

L'indice d'importance d'un critère quantifie l'importance de ce critère seul et l'importance de ce critère dans tous les groupes possibles de critères (un critère peut avoir une importance très faible s'il est seul, mais très forte s'il est associé à d'autres critères). Il

s'agit donc d'un indice d'importance global. Dans le cas de la moyenne pondérée, cet indice correspond au poids du critère. Il a été montré [Shapley, 1953] que cet indice $\phi(i), i \in N$ peut être calculé de manière unique si on fait l'hypothèse que la somme des indices d'importance est 1.

$$\phi(i) = \sum_{A \subseteq N \setminus i} \frac{(n - |A| - 1)! |A|!}{n!} (\nu(A \cup \{i\}) - \nu(A))$$

L'indice d'interaction quantifie, pour un groupe de critères, la quantité de synergie de ce groupe en présence d'un groupe de critères A , pour tout groupe de critères A possibles. Dans le cas de deux critères, il se calcule de la manière suivante [Murofushi & Soneda, 1993] :

$$I_{ij} = \sum_{A \subseteq N \setminus i, j} \frac{(n - |A| - 2)! |A|!}{(n - 1)!} (\nu(A \cup \{i, j\}) - \nu(A \cup \{i\}) - \nu(A \cup \{j\}) + \nu(A))$$

La connaissance des indices d'importances et des indices d'interactions pour chaque groupe de critères détermine entièrement la capacité. Les indices d'importance et d'interaction peuvent donc être utilisés de deux façons : pour confirmer qu'une capacité a le comportement souhaité ou bien ils peuvent être fixés *a priori* pour la construction d'une capacité.

Un problème apparaît rapidement dans la détermination d'une capacité : pour n critères, il y a $2^n - 2$ paramètres libres. Afin de résoudre ce problème, les capacités k -additives ont été introduites. Une capacité k -additive est une capacité dont les indices d'interaction sont nuls au-delà de k critères. En pratique on utilise souvent des capacités 2-additives, c'est-à-dire que les indices d'interaction sont nuls au-delà de 2 critères. Le nombre de paramètres à fixer est alors de $\frac{n(n+1)}{2} - 1$.

A.3 Utilisation de R pour déterminer des capacités

Le logiciel de calcul open-source R¹ permet de construire des capacités suivant un ensemble de contraintes et d'utiliser ces capacités pour le calcul d'agrégation basé sur l'intégrale de Choquet. Il faut pour cela utiliser la librairie Kappalab². Nous décrivons ici la méthode employée pour déterminer la capacité donnée en exemple en section 7.2.5.

Nous utilisons la fonction *mini.var.capa.ident* dont voici la signature :

1. <http://www.r-project.org/>
 2. <http://cran.r-project.org/web/packages/kappalab>

```
mini.var.capa.ident(n, k,  
A.Choquet.preorder = NULL,  
A.Shapley.preorder = NULL,  
A.Shapley.interval = NULL,  
A.interaction.preorder = NULL,  
A.interaction.interval = NULL,  
A.inter.additive.partition = NULL,  
epsilon = 1e-6)
```

Les paramètres que nous utilisons sont le nombre de critères n , la valeur de k -additivité k , un pré-ordre sur les indices d'importance des critères $A.Shapley.preorder$, un intervalle pour la valeur des indices d'interaction $A.interaction.interval$.

Les contraintes spécifiées dans la section 7.2.5 sont :

$$\begin{aligned}\phi(P_2) &> \phi(P_1) \\ I_{P_1P_3} &< 0 \\ I_{P_1P_3} &\geq 0 \\ I_{P_2P_4} &\geq 0 \\ I_{P_1P_4} &\geq 0 \\ I_{P_2P_3} &\geq 0\end{aligned}$$

Nous traduisons ces contraintes dans R de la manière suivante :

```
ASP<-rbind(c(2,1,0.5)) #Première containte  
  
AII<-rbind(c(1,2,-0.1,-0.1),c(1,3,0.01,1),c(2,4,0.01,1),  
c(1,4,0.01,1),c(2,3,0.01,1),c(3,4,0,1)) #Contraintes suivantes
```

Une fois les contraintes choisies, on peut déclencher le calcul de la capacité et des différents indices de la manière suivante :

```
min.var<-mini.var.capa.ident(4,2,NULL,ASP,NULL,NULL,AII,NULL,epsilon=1e-6)  
m=min.var$solution  
m # affiche les valeurs de la capacité solution  
Shapley.value(m) # affiche les indices d'importance  
interaction.indices(m) # affiche les indices d'interaction  
Choquet.integral(m,a) # affiche le score d'une alternative a
```


Annexe B

Architecture d'une application Android

Une application Android est basée sur quatre éléments fondamentaux : les activités, les services, les *content providers* et les *broadcast receiver*. Une activité représente un écran de l'application avec lequel interagit effectivement l'utilisateur, une application est généralement composée de plusieurs activités, chacune d'entre elles représentant un écran de l'application.

Les services sont des composants qui s'exécutent sans *feedback* visible pour l'utilisateur, ils peuvent être lancés dans des *threads* indépendants de l'activité principale et permettent donc généralement d'effectuer des opérations longues sans bloquer l'interface utilisateur comme par exemple des opérations réseau.

Les *content providers* sont semblables à des interfaces (au sens d'interface en programmation orienté objet) que possèdent les applications permettant l'échange de données entre différentes applications. Par exemple l'application répertoire téléphonique d'un appareil possède un *content provider* lui permettant d'échanger les contacts téléphoniques disponibles sur le téléphone avec d'autres applications disposant des autorisations adéquates.

Enfin, les *broadcast receiver* permettent la réception de messages émis par les autres applications, les services ou par l'environnement Android lui-même (par exemple lorsque la batterie est faible, l'environnement Android envoie le message batterie faible qui peut être capturé par tous les *broadcast receiver*).

La communication entre ces différents éléments est assurée par un mécanisme interne de message appelé *intents*. Un *intent* est typiquement émis par une activité pour démarrer ou arrêter une autre activité ou un service, ou pour envoyer un message de type *broadcast* à l'ensemble des applications.

La figure B.1 représente une application Android typique, séparée en différentes activités et utilisant des *content providers* pour l'accès aux données applicatives.

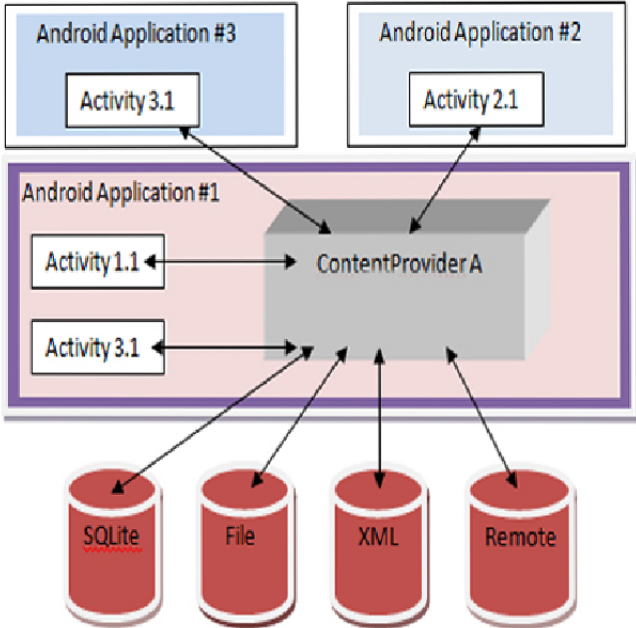


FIGURE B.1: Organisation des composants dans une application Android

Annexe C

Utilisation d'Eclipse-RCP pour les interfaces utilisateurs

C.1 Eclipse RCP

Nous avons conçu l'interface des applications serveur en utilisant le framework Eclipse RCP¹. Il s'agit d'un environnement de développement basé sur l'architecture OSGi permettant le développement de modules indépendants et réutilisables. Une application RCP est ainsi constituée de différents modules dont la cohabitation n'est pas nécessairement prévue au départ, le cycle de vie et les communications entre modules étant régis par OSGi. L'exemple le plus connu d'application RCP est l'environnement de développement Eclipse lui même.

Concrètement, Eclipse RCP permet de développer de manière rapide des applications ayant le *look and feel* d'Eclipse. Si l'apprentissage du développement pour RCP peut s'avérer fastidieux au départ, le gain de temps dans la création d'une application complexe par rapport à SWING est considérable. Le développement d'interfaces graphiques en utilisant Eclipse RCP est en grande partie déclaratif, chaque application RCP comprend un fichier *plugin.xml* qui permet de définir de manière déclarative les différentes vues ainsi que les menus et commandes de l'application, ces commandes pourront ensuite être dynamiquement associées à des classes Java. La figure C.1 représente la liste des commandes de l'application CALM-Gen

1. <http://www.eclipse.org/home/categories/rcp.php>

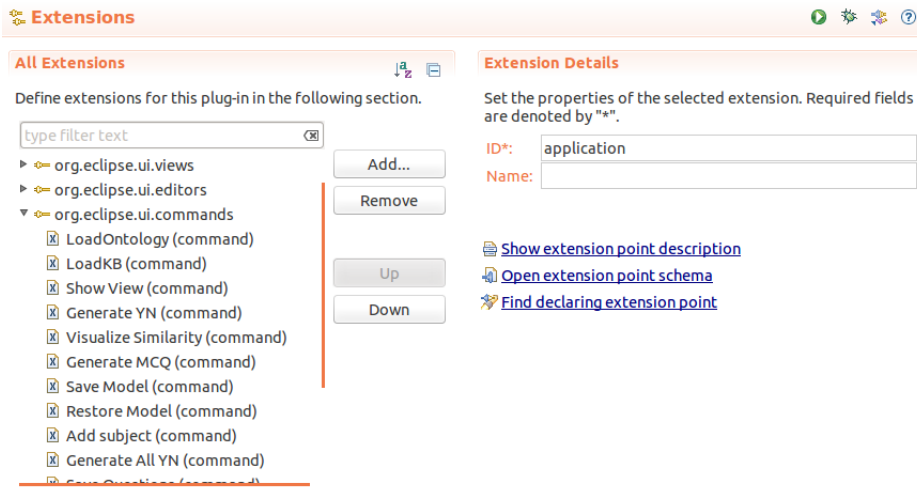


FIGURE C.1: Les commandes de CALM-Gen

C.2 Illustration

Nous décrivons à titre illustratif l'utilisation d'Eclipse RCP pour la création de l'interface de l'application CALM-Mapping.

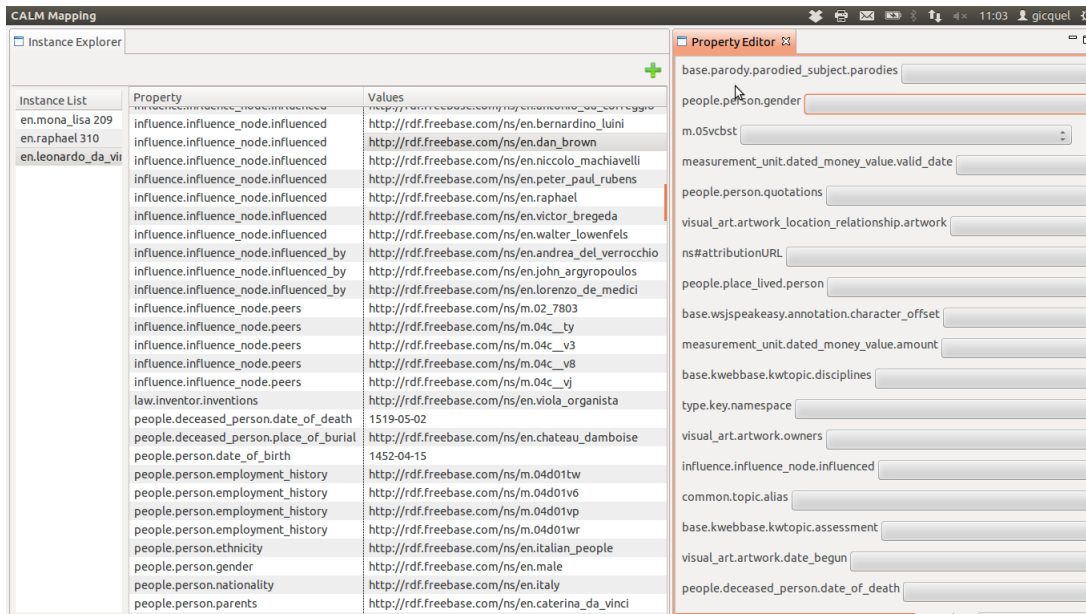


FIGURE C.1: L'application CALM-Mapping

L'interface utilisateur est composée de la vue *Instance Explorer* (classe *View* du framework RCP) et de l'éditeur *Property Editor* (classe *Editor*). La vue comporte un objet de type *ListViewer*, permettant l'affichage de l'ensemble des instances et un objet de type *TableViewer* permettant l'affichage des propriétés.

L'utilisation des objets de type *ListViewer* ou *TableViewer* est assez particulière. On associe aux *viewers* deux classes : une classe du type *ContentProvider* et une classe du type *LabelProvider*. Les *ContentProvider* sont chargés de transformer l'objet passé en entrée au *viewer* en un tableau d'objets. Par exemple dans le cas de la liste des instances, l'objet passé en entrée est de type *CalmInstanceGroup*, le *ContentProvider* transforme l'objet *CalmInstanceGroup* en un tableau d'objets de type *CalmInstance*. Les *LabelProvider* sont quant à eux chargés de déterminer ce qui sera affiché dans une case ou sur une ligne en fonction de l'objet considéré. Dans le cas du *ListViewer* affichant la liste des instances, le *LabelProvider* affiche le nom de l'instance ainsi que le nombre de ses propriétés. Ce mécanisme est intéressant, car les *ContentProvider* et les *LabelProvider* peuvent facilement être réutilisés.

Dans le cas de l'application *CALM-Mapping*, l'utilisation du pattern *Listener* sur l'objet de type *ListViewer* permet de mettre à jour le *TableViewer* quand une instance est sélectionnée. L'éditeur (section droite de la figure) est quant à lui indépendant de la vue, il est produit par itération sur l'ensemble des propriétés identifiées dans la base de connaissances source. Pour chacune des propriétés, une *combo-box* liste l'ensemble des propriétés de l'ontologie cible, permettant ainsi l'association des propriétés de la source aux propriétés de la cible.

Annexe D

L'application CALM-Gen

L'application *CALM-Gen* permet de générer, à partir d'une base de connaissances, l'ensemble des jeux d'auto-évaluation et des suggestions possibles. Cette application se compose de 14 packages pour 98 classes Java. Cependant, les packages les plus importants sont au nombre de quatre : *calm.distance.resource*, *calm.distance.compare*, *calm.distance.sentenceGenerator* et *calm.distance.questionGenerator*. Nous décrivons dans la suite ces packages.

Le package *calm.distance.resource* contient les représentations des différents types de ressources ontologiques que nous utilisons : la classe *CalmConcept*, la classe *CalmInstance* et la classe *CalmConceptGroup*. La classe *CalmConcept* est la plus importante, chaque instance de cette classe correspond à un concept distinct de l'ontologie. Chaque instance de *CalmConcept* est également associée un ensemble d'instances de *CalmInstance* de la même façon que les instances sont associées aux concepts dans la base de connaissance. À partir d'une instance de la classe *CalmConcept*, il est donc possible de récupérer l'ensemble des instances du concept en question, et, pour chaque instance, l'ensemble des noms et valeurs des propriétés de cette instance. C'est sur cette base que s'effectuent les comparaisons des propriétés des instances lors des calculs sémantiques.

Le package *calm.distance.compare* contient les classes nécessaires à la comparaison des instances entre elles. Ce package comporte deux classes principales, la classe *InstanceComparer* et la classe *InstanceSimilarity*. Les instances de la classe *InstanceSimilarity* représentent la proximité entre deux instances de la base de connaissances. Cette classe comprend donc trois champs principaux : *instanceA*, *instanceB* et *conceptValue*, en effet le calcul de la proximité entre deux instances dépend du concept à laquelle appartiennent ces deux instances. La classe *InstanceComparer* est chargée de l'organisation de la stratégie de comparaison. Elle permet, à partir d'une instance de *CalmInstance* et d'une instance de *CalmConcept* de déterminer l'ensemble des instances de la base de connaissances proches de *CalmInstance* et les valeurs de proximités.

Enfin, les packages *calm.distance.sentenceGenerator* et *calm.distance.questionGenerator*

sont consacrés à la génération automatique de questions ainsi qu'à la génération des justifications dans le cadre de la suggestion. Ces packages utilisent essentiellement l'API *simpleNLG* pour la construction des questions et des suggestions. Cette API permet la génération de phrases ou de questions facile, l'exemple de code suivant illustre le code nécessaire à génération de la question « Did Leonardo Da Vinci paint the Joconde ? ».

```
public static String realizeQuestion(){
    SPhraseSpec p = nlgFactory.createClause();
    p.setSubject("Leonardo da Vinci");
    p.setVerb("paint");
    p.setObject("the joconde");
    p.setFeature(Feature.TENSE, Tense.PAST);
    p.setFeature(Feature.INTERROGATIVE_TYPE, InterrogativeType.YES_NO);
    return realiser.realiseSentence(p);
}
```

FIGURE D.1: Exemple d'utilisation de SimpleNLG

Annexe E

Questionnaire de satisfaction : chateau de Compiègne

Le questionnaire de satisfaction que nous avons proposé à l'issue de l'expérimentation en situation réelle au chateau de Compiègne est présenté page suivante. Le fait de proposer un concours dans ce questionnaire nous a permis de récupérer des informations sur les catégories de personnes qui ont utilisé notre système (classe d'âge, niveau d'étude...).

Parallèlement au questionnaire, nous avons engagé des discussions avec plusieurs utilisateurs afin d'évaluer de manière plus qualitative le ressenti quant à l'utilisation d'une tablette durant une visite de musée. Dans l'ensemble, indépendamment de la classe d'âge ou du niveau d'étude, nous avons constaté un ressenti très positif. En particulier, de manière assez surprenante, les personnes âgées ont manifesté un vif intérêt pour l'application tablette et n'ont pas eu de problèmes particuliers pour son utilisation.

Le principal problème soulevé dans ces discussions a été la question de la confiance à accorder aux annotations. En effet, les visiteurs ont l'habitude d'un contenu culturel (ex. textes et autres média) validé scientifiquement par les conservateurs. Les annotations, posées librement par les visiteurs, sortent du champ de cette validation scientifique. La question d'un « niveau de confiance », attribué aux annotations, a ainsi été plusieurs fois soulevée.

Annexe E Questionnaire de satisfaction : chateau de Compiègne



Enquête de satisfaction – Test de l'application CALM

Bonsoir! Vous venez de participer à la première soirée de test de l'application CALM!
Pour nous aider à la perfectionner, merci de nous donner votre opinion sur les sujets suivants...

	Très satisfaisant	Satisfaisant	Peu satisfaisant	Pas du tout satisfaisant
Utilisation des tablettes Archos				
Taille de l'écran				
Poids de la tablette				
Réactivité de la tablette				

Autres remarques :

	Très satisfaisant	Satisfaisant	Peu satisfaisant	Pas du tout satisfaisant
Simplicité d'utilisation				
Consultation des notices				
Consultation des informations annexes				
Dépôt d'annotations				
Consultation d'annotations				

Autres remarques :

	Très satisfaisant	Satisfaisant	Peu satisfaisant	Pas du tout satisfaisant
Utilité d'une tablette en musée				
Intérêt de la consultation des notices				
Pertinences des notices proposées				
Intérêt de la consultation des informations annexes				
Pertinences des informations proposées				
Intérêt du système d'annotation				

Autres remarques :

Participez au tirage au sort en répondant à ces questions et gagnez des cadeaux !			
Marie-Louise était de Nationalité :	Française	Autrichienne	Hongroise
Le fils de Napoléon avait le titre de	Prince des Francs	Duc de Bretagne	Roi de Rome
Le prénom de Canova était :	Anne-Louis	Jean-Paul	Jean-Jacques
Informations personnelles			
Nom :	Email ou téléphone :		
Prénom :	Dernier diplôme obtenu :		
Age :	Spécialité du diplôme :		
Adresse :	Activité professionnelle :		

Merci d'avoir participé à cette enquête!



FIGURE E.1: Questionnaire de satisfaction

Bibliographie

- [Abowd *et al.*, 1999] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness. in *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing* pp. 304–307.
- [Aimé, 2011] Aimé, X. (2011). *Gradients de prototypicalité, mesures de similarité et de proximité sémantique : une contribution à l'Ingénierie des Ontologies* PhD thesis Université de Nantes.
- [Aimé *et al.*, 2011] Aimé, X., Furst, F., Kuntz, P., & Trichet, F. (2011). Semiose et proxima : mesures sémiotiques de similarité et de proximité conceptuelles. in *Actes des 22e journées francophones d'Ingénierie des Connaissances* pp. 539–554.
- [Al-Yahya, 2011] Al-Yahya, M. (2011). OntoQue: a question generation engine for educational assesment based on domain ontologies. in *Proceedings of ICALT 2011* pp. 393–395. IEEE.
- [Anderson *et al.*, 2003] Anderson, D., Lucas, K. B., & Ginns, I. S. (2003). Theoretical perspectives on learning in an informal setting. *Journal of Research in Science Teaching* 40(2), 177–199.
- [Anderson *et al.*, 2002] Anderson, D., Piscitelli, B., Weier, K., Everett, M., & Tayler, C. (2002). Children's museum experiences: identifying powerful mediators of learning. *Curator : The Museum Journal* 45(3), 213–231.
- [Anyanwu *et al.*, 2005] Anyanwu, K., Maduko, A., & Sheth, A. (2005). Semrank, ranking complex relationship search results on the semantic web. in *Proceedings of the 14th international conference on World Wide Web* pp. 117–127.
- [Aubry, 2007] Aubry, S. (2007). *Annotations et gestion des connaissances en environnement virtuel collaboratif* PhD thesis Université de Technologie de Compiègne.
- [Autere & Vakkari, 2011] Autere, R. & Vakkari, M. (2011). Towards cross-organizational interoperability: The LIDO XML schema as a national level integration tool for the national digital library of finland. in *Proceedings of the 15th international conference on Theory and practice of digital libraries: research and advanced technology for digital libraries* pp. 62–68.

- [Baader *et al.*, 2005] Baader, F., Horrocks, I., & Sattler, U. Description logics as ontology languages for the semantic web (2005). in *Mechanizing Mathematical Reasoning: Essays in Honor of Jörg Siekmann on the Occasion of His 60th Birthday* (Hutter, D. & Stephan, W., eds.) number 2605 in Lecture Notes in Artificial Intelligence (Springer).
- [Bachimont, 2000] Bachimont, B. Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances (2000). in *Ingénierie des Connaissances: Evolutions récentes et nouveaux défis* (Charlet J., Zacklad M., K. G. . B. D., ed.) (Eyrolles).
- [Bachimont, 2004] Bachimont, B. (2004). Arts et sciences du numérique : ingénierie des connaissances et critique de la raison computationnelle. Habilitation à diriger des recherches. Université de Technologie de Compiègne.
- [Baldauf *et al.*, 2007] Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing* 2(4), 263–277.
- [Barot *et al.*, 2011] Barot, C., Burkhardt, J.-M., Lourdeaux, D., & Lenne, D. (2011). V3S, a virtual environment for risk management training. in *Proceedings of JVRC11: Joint Virtual Reality Conference of EGVE - EuroVR* pp. 95–102.
- [Bauer, 2003] Bauer, J. (2003). *Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic* PhD thesis Technische Universität Berlin.
- [Baumgartner & Retschitzegger, 2006] Baumgartner, N. & Retschitzegger, W. (2006). A survey of upper ontologies for situation awareness. in *Proceedings of the 4th International Conference on Knowledge Sharing and Collaborative Engineering*.
- [Bergmann & Stahl, 1998] Bergmann, R. & Stahl, A. (1998). Similarity measures for object-oriented case representations. in *Proceedings of the 4th European Workshop on Advances in Case-Based Reasoning* pp. 25–36.
- [Berners-Lee, 2006] Berners-Lee, T. (2006). Notation 3 - an readable language for data on the web Technical report W3C.
- [Bettini *et al.*, 2010] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing* 6(2), 161–180.
- [Bourdieu & Darbel, 1969] Bourdieu, P. & Darbel, A. (1969). *L'amour de l'art: les musées et leur public* (Editions de Minuit).
- [Brachman & Levesque, 2004] Brachman, R. & Levesque, H. (2004). *Knowledge Representation and Reasoning* (Morgan Kaufmann Publishers Inc.).
- [Brooke, 1996] Brooke, J. (1996). SUS: a quick and dirty usability scale. *Usability evaluation in industry* 189.
- [Brown *et al.*, 1989] Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational researcher* 18(1), 32–42.

- [Bruns *et al.*, 2007] Bruns, E., Brombach, B., Zeidler, T., & Bilber, O. (2007). Enabling mobile phones to support large-scale museum guidance. *IEEE MultiMedia* 14(2), 16–25.
- [Carnegie Council on Adolescent Development, 1994] Carnegie Council on Adolescent Development (1994). A matter of time : risk and opportunity in the non school hours. report of the force task on youth development. Technical report.
- [Caron, 2007] Caron, P. A. (2007). *Ingénierie dirigée par les modèles pour la construction de dispositifs pédagogiques sur des plateformes de formation* PhD thesis Université des Sciences et Technologie de Lille - Lille I.
- [Caron *et al.*, 2008] Caron, P.-A., Le Pallec, X., Kaddouci, S., & Ritaine, F. J. (2008). Experimentations d'apprentissage pervasif, juste a temps, en situation de travail. in *Actes du Workshop EMAP : Environnements Mobiles et Apprentissage Pervasif, TICE 2008*.
- [Castro *et al.*, 2001] Castro, P., Chiu, P., Kremenek, T., & Muntz, R. (2001). A probabilistic room location service for wireless networked environments. in *Proceedings of the 3rd international conference on Ubiquitous Computing* p. 18–34.
- [Chen *et al.*, 2003] Chen, H., Finin, T., & Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *Knowledge Engineering Review* 18(3), 197–207.
- [Chen *et al.*, 2004] Chen, H., Perch, F., & Joshi, A. (2004). SOUPA: standard ontology for ubiquitous and pervasive applications. in *International Conference on Mobile and Ubiquitous Systems: Networking and Services* pp. 258–267.
- [Choi *et al.*, 2005] Choi, J., Shin, D., & Shin, D. (2005). Research and implementation of the context-aware middleware for controlling home appliances. *IEEE Transactions on Consumer Electronics* 51(1), 301–306.
- [Chou *et al.*, 2005] Chou, S. C., Hsieh, W. T., Gandon, F. L., & Sadeh, N. M. (2005). Semantic web technologies for context-aware museum tour guide applications. in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*. pp. 709–714.
- [Damala, 2009] Damala, A. (2009). *Interaction Design and Evaluation of Mobile Guides for the Museum Visit: A Case Study in Multimedia and Mobile Augmented Reality*. PhD thesis Conservatoire national des arts et métiers. CEDRIC Laboratory, Paris, France.
- [Davallon, 1992] Davallon, J. (1992). Le musée est-il vraiment un média ?. *Publics et Musées* 2(1), 99–123.
- [Davis *et al.*, 1993] Davis, R., Shrobe, H., & Szolovits, P. (1993). What is a knowledge representation?. *AI magazine* 14(1), 17–33.
- [Debenedetti, 2003] Debenedetti, S. Visite occasionnelle du musée d'art et confort de visite : le rôle des compagnons (2003). in *Le(s) Public(s) de la culture* (et Tolila P., D. O., ed.) (Presses de Sciences Po).

- [Detyniecki, 2001] Detyniecki, M. (2001). Fundamentals on aggregation operators Technical report University of California Berkeley.
- [Doerr, 2003] Doerr, M. (2003). The cidoc conceptual reference module: an ontological approach to semantic interoperability of metadata. *AI magazine* 24(3), 75–92.
- [Doerr *et al.*, 2010] Doerr, M., Gradmann, S., Hennicke, S., Isaac, A., Meghini, C., & van de Sompel, H. (2010). The europeana data model (EDM). in *Proceedings of the 76th International Federation of Library Associations and Institutions Conference*.
- [Doerr *et al.*, 2007] Doerr, M., Ore, C.-E., & Stead, S. (2007). The CIDOC conceptual reference model: a new standard for knowledge sharing. in *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling - Volume 83 ER '07* p. 51–56. Australian Computer Society, Inc.
- [Dourish, 2004] Dourish, P. (2004). What we talk about when we talk about context. *Personal and ubiquitous computing* 8(1), 19–30.
- [Du Sablon & Racette, 1991] Du Sablon, C. & Racette, G. (1991). Les effets d'un programme éducatif muséal chez des élèves du primaire. *Revue canadienne de l'éducation* 16(3), 338–351.
- [Edward *et al.*, 2010] Edward, L., Lourdeaux, D., & Barthès, J.-P. (2010). Une architecture cognitive pour la simulation de comportements d'agents autonomes. intégration de facteurs physiques, physiologiques et de personnalité. *Revue d'Intelligence Artificielle* 24(5), 673–694.
- [Fielding, 2000] Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* PhD thesis University of California, Irvine.
- [Gandon & Sadeh, 2004] Gandon, F. & Sadeh, N. (2004). Gestion de connaissances personnelles et contextuelles, et respect de la vie privée. in *Actes d'Ingénierie des Connaissances 2004*.
- [George & Lekira, 2009] George, S. & Lekira, A. S. (2009). MeCoCo, a context-aware system for mediated communications. *International Journal of Interactive Mobile Technologies (iJIM)* 3.
- [Ginsberg, 1991] Ginsberg, M. L. (1991). Knowledge interchange format: The KIF of death. *AI magazine* 12(3), 57–63.
- [Gob & Drouguet, 2006] Gob, A. & Drouguet, N. (2006). *La muséologie: histoire, développements, enjeux actuels* (Armand Colin).
- [Goncalves, 2010] Goncalves, A. SOAP web services (2010). in *Beginning Java™ EE 6 Platform with GlassFish™ 3* (Apress).
- [Grabisch, 2006] Grabisch, M. (2006). L'utilisation de l'intégrale de choquet en aide multicritere à la décision Technical Report 14 Newsletter of the European Working Group Multicriteria Aid for Decisions.
- [Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2), 199–220.

- [Guha & Brickley, 2004] Guha, R. V. & Brickley, D. (2004). RDF vocabulary description language 1.0: RDF schema W3C recommendation W3C. <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>.
- [Guo *et al.*, 2005] Guo, Y., Pan, Z., & Heflin, J. (2005). LUBM: a benchmark for OWL knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web* 3(2-3), 158–182.
- [Haake *et al.*, 2010] Haake, M., Hussein, T., BJÖRN, J., Lukosch, S., Veiel, D., & Ziegler, J. (2010). Modeling and exploiting context for adaptive collaboration. *International Journal of Cooperative Information Systems* 1(2), 71–120.
- [Hage *et al.*, 2010] Hage, W., Stash, N., Wang, Y., & Aroyo, L. (2010). Finding your way through the rijksmuseum with an adaptive mobile museum guide. in *Proceedings of the 7th international conference on The Semantic Web: research and Applications - Volume Part I* pp. 46–59.
- [Henricksen & Indulska, 2004] Henricksen, K. & Indulska, J. (2004). Modelling and using imperfect context information. in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*. p. 33–37.
- [Henricksen *et al.*, 2002] Henricksen, K., Indulska, J., & Rakotonirainy, A. (2002). Modeling context information in pervasive computing systems. in *Proceedings of the First International Conference on Pervasive Computing* pp. 167–180.
- [Hirzy, 1992] Hirzy, E. (1992). *Excellence and Equity: Education and the Public Dimension of Museums* (The Association of American Museums).
- [Hurtado *et al.*, 2006] Hurtado, C., Poulouvasilis, A., & Wood, P. (2006). A relaxed approach to RDF querying. in *Proceedings of the 5th international conference on The Semantic Web* p. 314–328.
- [Indulska *et al.*, 2003] Indulska, J., Robinson, R., Rakotonirainy, A., & Henricksen, K. (2003). Experiences in using CC/PP in context-aware systems. in *Proceedings of the 4th International Conference on Mobile Data Management* pp. 247–261.
- [Kim & Kim, 1990] Kim, Y. & Kim, J. (1990). A model of knowledge based information retrieval with hierarchical concept graph. *Journal of Documentation* 46(2), 113–136.
- [Kojadinovic, 2006] Kojadinovic, I. (2006). Minimum variance capacity identification. *European Journal of Operational Research*.
- [Kuffik *et al.*, 2011] Kuffik, T., Stock, O., Zancanaro, M., Gorfinkel, A., Jbara, S., Kats, S., Sheidin, J., & Kashtan, N. (2011). A visitor’s guide in an active museum: Presentations, communications, and reflection. *Journal on Computing and Cultural Heritage* 3(3), 1–25.
- [Laine *et al.*, 2010] Laine, T. H., Vinni, M., Sedano, C. I., & Joy, M. (2010). On designing a pervasive mobile learning platform. *Research in Learning Technology* 18(1).
- [Lave & Wenger, 1991] Lave, J. & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation* (Cambridge University Press).

- [Le Ber *et al.*, 2006] Le Ber, F., Lieber, J., & Napoli, A. Les systèmes à base de connaissances (2006). in *Encyclopédie de l'informatique et des systèmes d'information* (Akoka, J. & Comyn-Wattiau, I., eds.) (Vuibert).
- [Leacock *et al.*, 1998] Leacock, C., Miller, G. A., & Chodorow, M. (1998). Using corpus statistics and WordNet relations for sense identification. *Computer Linguistic* 24(1), 147–165.
- [Lee *et al.*, 1993] Lee, J. H., Kim, M. H., & Lee, Y. J. (1993). Information retrieval based on conceptual distance in a is-a hierarchy. *Journal of Documentation* 49(2), 188–207.
- [Lefebvre & Lefebvre, 1993] Lefebvre, A. & Lefebvre, B. (1993). L'adulte et l'apprentissage du musée et au musée. *Revue des sciences de l'éducation* 19(4), 775–779.
- [Lemerise & Soucy, 1999] Lemerise, T. & Soucy, B. (1999). Le point de vue d'adolescents montréalais sur les musées. *Revue Canadienne de l'Éducation* 24(4), 355–368.
- [Lenat & Guha, 1991] Lenat, D. B. & Guha, R. V. (1991). The evolution of CycL, the cyc representation language. *SIGART Bull.* 2(3), 84–87.
- [Lenne, 2009] Lenne, D. (2009). Modélisation des connaissances et de l'interaction, application aux environnements informatiques pour l'Apprentissage humain. Habilitation à diriger des recherches Université de Technologie de Compiègne.
- [Lin, 1998] Lin, D. (1998). An information-theoretic definition of similarity. in *Proceedings of the 15th International Conference on Machine Learning* pp. 296–304.
- [Luyten *et al.*, 2006] Luyten, K., Winters, F., Coninx, K., Naudts, D., & Moerman, I. A situation-aware mobile system to support fire brigades in emergency situations (2006). in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops Lecture Notes in Computer Science* (Springer Berlin Heidelberg).
- [Marichal & Roubens, 2000] Marichal, J. L. & Roubens, M. (2000). Determination of weights of interacting criteria from a reference set. *European journal of operational Research* 124(3), 641–650.
- [Marty & Jones, 2012] Marty, P. & Jones, K. (2012). *Museum Informatics: People, Information, and Technology in Museums* (Taylor & Francis).
- [Matheus *et al.*, 2000] Matheus, C., Kokar, M., Baclawski, K., Letkowski, J., & Call, C. (2000). SAWA : An assistant for higher-level fusion and situation awareness. in *Proceedings of SPIE Conference on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications* pp. 75–85.
- [McGuinness & Harmelen, 2004] McGuinness, D. L. & Harmelen, F. v. (2004). OWL web ontology language overview W3C recommendation W3C. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- [Mihalcea *et al.*, 2006] Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. in *Proceedings of the 21st national conference on Artificial intelligence* pp. 775–780.

- [Minsky, 1974] Minsky, M. (1974). A framework for representing knowledge Technical report Massachusetts Institute of Technology Cambridge, MA, USA.
- [Murofushi & Soneda, 1993] Murofushi, T. & Soneda, S. (1993). Techniques for reading fuzzy measures (iii): Interaction index. in *Proceedings of the 9th Fuzzy Systems Symposium* p. 693–696.
- [Naismith *et al.*, 2005] Naismith, L., Sharples, M., & Ting, J. (2005). Evaluation of CAERUS: a context aware mobile guide. in *Proceedings of mLearn 2005*. p. 50.
- [Norman, 1986] Norman, D. A. Cognitive engineering (1986). in *User centered system design: New perspectives on human-computer interaction* (Laurence Erlbaum Associates, Inc., Publishers).
- [Ogata *et al.*, 2009] Ogata, H., Matsuka, Y., El-Bishouty, M. M., & Yano, Y. (2009). LORAMS: linking physical objects and videos for capturing and sharing learning experiences towards ubiquitous learning. *International Journal of Mobile Learning and Organisation* 3(4), 337–350.
- [Ogata & Yano, 2003] Ogata, H. & Yano, Y. (2003). How ubiquitous computing can support language learning. in *Proceedings of KEST*.
- [O’keefe & Preece, 1996] O’keefe, R. M. & Preece, A. D. (1996). The development, validation and implementation of knowledge-based systems. *European Journal of Operational Research* 92(3), 458–473.
- [Oppermann & Specht, 1999] Oppermann, R. & Specht, M. (1999). A nomadic information system for adaptive exhibition guidance. *Archives and museum informatics* 13(2), 127–138.
- [Oppermann & Specht, 2000] Oppermann, R. & Specht, M. (2000). A context-sensitive nomadic information system as an exhibition guide. in *Proceedings of the handheld and ubiquitous computing second international symposium* pp. 127–142.
- [Pachler, 2010] Pachler, N. The socio-cultural ecological approach to mobile learning: An overview (2010). in *The Socio-Cultural Ecological Approach to Mobile Learning: An Overview* (Bachmair, B., ed.) (VS Verlag für Sozialwissenschaften).
- [Papasalouros *et al.*, 2008] Papasalouros, A., Kotis, K., & Kanaris, K. (2008). Automatic generation of multiple-choice questions from domain ontologies. in *Proceedings of IADIS 2008* pp. 427–434.
- [Pedersen *et al.*, 2004] Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet::Similarity: measuring the relatedness of concepts. in *Demonstration Papers at HLT-NAACL 2004* HLT-NAACL–Demonstrations ’04 p. 38–41 Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Pham-Nguyen, 2010] Pham-Nguyen, C. (2010). *Conception d’un système d’apprentissage et de travail pervasif et adaptatif fondée sur un modèle de scénario* PhD thesis Télécom Bretagne.

- [Pham-Nguyen & Garlatti, 2008] Pham-Nguyen, C. & Garlatti, S. (2008). Un système d'apprentissage pervasif intégrant des activités de travail et de formation. in *Proceedings of the 4th French-speaking conference on Mobility and ubiquity computing* p. 65–72.
- [Pirro & Euzenat, 2010] Pirro, G. & Euzenat, J. (2010). A feature and information theoretic framework for semantic similarity and relatedness. in *Proceedings of the 9th international semantic web conference on the semantic web* pp. 615–630.
- [Plouznikoff *et al.*, 2006] Plouznikoff, N., Plouznikoff, A., Desmarais, M., & Robert, J. M. (2006). Support mobile \‘a la t^ache: m\’ediation utilisateur-monde r\’eel par l’interm\’ediaire d’un ordinateur port\’e. in *Proceedings of the 18th International Conference of the Association Francophone d’Interaction Homme-Machine* p. 107–114.
- [Plouznikoff & Robert, 2004] Plouznikoff, N. & Robert, J. M. (2004). Caractéristiques, enjeux et défis de l’informatique portée. in *Proceedings of the 16th conference on Association Francophone d’Interaction Homme-Machine* pp. 125–132.
- [Poulot, 2001] Poulot, D. (2001). *Patrimoine et Musée, l’institution de la culture* (HACHETTE Supérieur).
- [Poulot, 2010] Poulot, D. (2010). *Musée et muséologie* (HACHETTE Supérieur).
- [Rada *et al.*, 1989] Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics* 19(1), 17–30.
- [Ranganathan *et al.*, 2002] Ranganathan, A., Campbell, R., Ravi, A., & Mahajan, A. (2002). Conchat, a context-aware chat program. *IEEE Pervasive Computing* 1(3), 51–57.
- [Resnik, 1995] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. 448–453.
- [Resnik, 1993] Resnik, P. S. (1993). *Selection And Information: A Class-based Approach to Lexical Relationships* PhD thesis University of Pennsylvania.
- [Richardson *et al.*, 1994] Richardson, R., Smeaton, A. F., & Murphy, J. (1994). Using WordNet as a knowledge base for measuring semantic similarity between words. in *Proceedings of AICS Conference*.
- [Rocchi *et al.*, 2004] Rocchi, C., Stock, O., Zancanaro, M., Kruppa, M., & Krüger, A. (2004). The museum visit: Generating seamless personalized presentations on multiple devices. in *Proceedings of the 9th international conference on Intelligent user interfaces*. p. 316–318. ACM Press.
- [Rodriguez & Egenhofer, 2003] Rodriguez, M. & Egenhofer, M. (2003). Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering* 15(2), 442–456.

- [Salber *et al.*, 1999] Salber, D., Dey, A., & Abowd, G. (1999). The context toolkit: aiding the development of context-enabled applications. in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* p. 434–441.
- [Salton & McGill, 1986] Salton, G. & McGill, M. J. (1986). *Introduction to Modern Information Retrieval* (McGraw-Hill, Inc.: New York, NY, USA).
- [Schilit *et al.*, 1994] Schilit, B., Adams, N., & Want, R. (1994). Context-aware computing applications. in *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*. p. 85–90.
- [Schmidt, 2000] Schmidt, A. (2000). Implicit human computer interaction through context. *Personal and Ubiquitous Computing* 4(2), 191–199.
- [Semper & Spasojevic, 2002] Semper, R. & Spasojevic, M. (2002). The electronic guidebook: using portable devices and a wireless web-based network to extend the museum experience. in *Proceedings of Museum and the Web 2002*.
- [Shapley, 1953] Shapley, L. S. A value for n-person games (1953). in *Contribution to the Theory of Games* (Kuhn, ed.) volume 2 (Princeton University Press).
- [Sparacino, 2002] Sparacino, F. (2002). The museum wearable: real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. in *Proceedings of Museums and the Web (MW2002)*.
- [Stefik, 1995] Stefik, M. (1995). *Introduction to Knowledge Systems* (Morgan Kaufmann Publishers).
- [Stevens & Martell, 2003] Stevens, R. & Martell, S. T. (2003). Leaving a trace: Supporting museum visitor interaction and interpretation with digital media annotation systems. *Journal of Museum Education* 28(2), 25–30.
- [Strang & Linnhoff-Popien, 2004] Strang, T. & Linnhoff-Popien, C. (2004). A context modeling survey. in *Workshop on Advanced Context Modelling, Reasoning and Management*.
- [Tallon & Walker, 2008] Tallon, L. & Walker, K. (2008). *Digital Technologies and the Museum Experience: Handheld Guides and Other Media* (AltaMira Press).
- [Thom-Santelli *et al.*, 2005] Thom-Santelli, J., Toma, C., Boehner, K., & Gay, G. (2005). Beyond just the facts: Museum detective guides. in *Proceedings of the Int'l Workshop 'Re-thinking technology in museums: Towards a new understanding of people's experience in museums*. p. 2006.
- [Tversky, 1977] Tversky, A. (1977). Features of similarity. *Psychological Review* 84(4), 327–352.
- [van Hage *et al.*, 2011] van Hage, W., Malaisé, V., Segers, R., Hollink, L., & Schreiber, G. (2011). Design and use of the simple event model (SEM). *Web Semantics: Science, Services and Agents on the World Wide Web* 9, 128–136.
- [Veron & Levasseur, 1983] Veron, E. & Levasseur, M. (1983). *Ethnographie de l'exposition*.

- [Wang *et al.*, 2009] Wang, Y., Stash, N., Aroyo, L., Hollink, L., & Schreiber, G. (2009). Using semantic relations for content-based recommender systems in cultural heritage. in *Proceedings of the Workshop on Ontology Patterns (WOP) at ISWC*.
- [Warin *et al.*, 2005] Warin, M., Oxhammar, H., & Volk, M. (2005). Enriching an ontology with wordnet based on similarity measures. in *MEANING-2005 Workshop*.
- [Weiser, 1991] Weiser, M. (1991). The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review* 3(3), 3–11.
- [Winston, 1992] Winston, P. H. (1992). *Artificial Intelligence, third edition*.
- [Wu & Palmer, 1994] Wu, Z. & Palmer, M. (1994). Verbs semantics and lexical selection. in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* pp. 133–138.
- [Ye *et al.*, 2007] Ye, J., Coyle, L., Dobson, S., & Nixon, P. (2007). Ontology-based models in pervasive computing systems. *The Knowledge Engineering Review* 22(4), 315–347.
- [Zancanaro *et al.*, 2003] Zancanaro, M., Stock, O., & Alfaro, I. (2003). Using cinematic techniques in a multimedia museum guide. in *Proceedings of Museum and the Web 2003*. volume 12.
- [Zimmermann & Lorenz, 2003] Zimmermann, A. & Lorenz, A. (2003). Listen: Contextualized presentation for audio-augmented environments. in *Proceedings of the 11th Workshop on Adaptivity and User modeling in Interactive Systems* p. 351–357.
- [Zimmermann *et al.*, 2007] Zimmermann, A., Lorenz, A., & Oppermann, R. An operational definition of context (2007). in *Modeling and Using Context Lecture Notes in Computer Science* (Springer Berlin Heidelberg).