L'interopérabilité des données historiques et la question du modèle : l'ontologie du projet SyMoGIH

Francesco Beretta

L'interopérabilité des données historiques et la question du modèle : l'ontologie du projet SyMoGIH

Francesco Beretta

(LARHRA UMR5190, CNRS – Université de Lyon)

Publié dans :

Enjeux numériques pour les médiations scientifiques et culturelles du passé,

(Brigitte Juanals / Jean-Luc Minel, sous la dir. de),

Paris, Presses Universitaires de Paris Nanterre, 2017, 87-127.

Le développement des sciences de l'information et des technologies du web sémantique ouvre des perspectives inespérées pour la recherche en histoire en termes de volume de données mise à disposition et d'interopérabilité entre silos issus de différents projets et institutions[1]. Mais une question se pose d'emblée, délicate et complexe : qu'est-ce qu'une *donnée historique* ? De la réponse à cette question dépend toute l'architecture d'un système d'information, de même que toute possibilité de faire dialoguer les données issues de différents entrepôts afin de produire de nouvelles connaissances susceptibles d'être utilisées pour la recherche en histoire.

Qu'est-ce qu'une donnée historique ?

Un article récent de la revue *Semantic Web* propose un tour d'horizon fort bien documenté sur les projets, modèles et technologies sémantiques qui permettent de traiter et d'échanger les 'données' historiques[2]. On constate toutefois dans ce texte un certain flou dans la définition et dans l'utilisation des concepts : les 'données' historiques sont appelées à tour de rôle « historical data », « historical dataset », « historical information », « historical sources » et aussi « historical background

1 Voir par exemple le projet *SESHAT: Global History Databank*, https://evolution-institute.org/project/seshat/. Remarque générale : tous les sites web mentionnés ont été consultés le 2 février 2015.

2 MEROÑO-PEÑUELA Albert, ASHKPOUR Ashkan, VAN ERP Marieke, MANDEMAKERS Kees, BREURE Leen, SCHARNHORST Andrea, SCHLOBACH Stefan, VAN HARMELEN Frank, « Semantic Technologies for Historical Research: A Survey », in *Semantic Web – Interoperability, Usability, Applicability* (IOS Press), http://www.semantic-web-journal.net/content/semantic-technologies-historical-research-survey-0. On trouvera dans cet article une bibliographie et une webographie développées concernant notre sujet.

knowledge », ce savoir qui doit être fourni avec les données pour permettre leur interprétation et qui en fait donc partie.

D'une part, cette indétermination conceptuelle résulte de l'ambition de l'article qui présente une vision synthétique de l'ensemble de la question, en traçant d'abord les contours d'une série de problèmes rencontrés par les chercheurs, puis en montrant comment les différents modèles et technologies sémantiques existantes pourraient apporter une réponse aux questions soulevées. Les auteurs reconnaissent que certaines questions —et pas des moindres, comme le développement d'« historical ontologies » permettant l'échange des données— restent ouvertes[3]. Mais leur visée synthétique les amène à comparer des projets allant de l'encodage de textes en XML, à l'exploitation de recensements pour l'histoire sociale, en passant par la conservation des biens culturels issus de l'Antiquité classique, qui comportent des niveaux d'analyse et des approches très diverses.

D'autre part, on sent clairement dans le texte l'influence de l'informatique lors de la présentation du processus de production des données, appelé « life cycle of historical information »[4]. Or, comme le soulignent les auteurs de l'article, en informatique la définition d'une donnée n'est pas univoque car on peut penser à des données *non structurées*, simples ensembles de chiffres et de chaînes de caractères, ou à des collections de variables qualitatives ou quantitatives, saisies sous forme de texte ou tabulaire, ou mises en forme grâce à des standards tels XML ou JSON afin d'en faire des *données semi-structurées*, pour aboutir enfin aux *données structurées* produites grâce à une modélisation qui explicite leur sémantique et stockées dans une base de données ou sous forme d'entrepôt de triplets[5]. L'articulation entre ces différents niveaux de structuration des données, et en particulier la question de l'extraction de connaissances sous forme de données (semi-)structurées à partir du texte brut, est un vaste sujet auquel ont été consacrés de nombreux travaux en matière de technologies sémantiques[6]. Le fait de traiter en même temps cette question, relevant plus spécifiquement de l'informatique, et celle de l'interopérabilité des 'données' historiques risque de provoquer des confusions.

Surtout, la conception d'un *life cycle* qui part de l'océrisation des sources pour en tirer des textes numériques qui seront annotés et enrichis afin d'en extraire des informations en vue de produire de nouvelles connaissances —à chaque étape de ce processus on rencontre des 'données' de nature

3 *Ibid.*, section 5.
4 *Ibid.*, section 3.1.
5 *Ibid.*, section 3.2.4. Cf. http://en.wikipedia.org/wiki/Semi-structured_data et http://en.wikipedia.org/wiki/Data_model.
6 BONTCHEVA Kalina et CUNNINGHAM Hamish, « Semantic annotations and retrieval », in *Handbook of semantic web technologies. Foundation and technologies*, DOMINGUE John, FENSEL Dieter et HENDLER James A. (dir.), Berlin / Heidelberg, Springer, 2011, p. 77-116 ; *Semantische Technologien. Grundlagen, Konzepte, Anwendungen*, DENGEL Andreas (dir.), Heidelberg, Spektrum Akademischer Verlag, 2012, chapitre 8.

différente— correspond certes à la méthode informatique mais ne tient pas suffisamment compte d'un acquis essentiel de la méthode historique selon laquelle toute 'donnée' historique, tout 'fait' historique même élémentaire, résulte d'une *construction,* c'est-à-dire d'une opération mentale qui prend son origine dans un questionnement[7]. Même les historiens de la fin du XIX[e] siècle, théoriciens et praticiens de l'historiographie méthodique et positiviste dont la conception de « fait historique » se rapproche peut être le plus d'une conception objectiviste de la donnée, étaient conscients de l'importance du questionnement et de la dimension interprétative de la discipline historique[8].

Dans ce chapitre, il s'agira donc tout d'abord de proposer une définition possible de ce qu'est une *donnée historique*, définition entendue non pas au sens normatif —car plusieurs définitions sont possibles— mais comme expression d'un croisement entre deux perspectives : celle de l'histoire (plus précisément de la réflexion sur la méthode historique) et celle de l'informatique (plus précisément de l'ingénierie des connaissances). Cette définition sera formalisée en adoptant un modèle conceptuel générique des données, exprimé en utilisant les vocabulaires de modélisation propres au web sémantique et notamment le *Resource Description Framework* (RDF) et le *RDF schema* (RDFS).

Le modèle présenté est issu de l'expérience du projet *Système modulaire de gestion de l'information historique* (SyMoGIH), développé dès 2007 au sein du Pôle histoire numérique du Laboratoire de recherche historique Rhône-Alpes (LARHRA)[9]. Ce modèle a été conçu afin de permettre l'interopérabilité entre données produites par différents projets de recherche, individuels et collectifs, relevant de différentes approches disciplinaires (histoire sociale, économique, intellectuelle, religieuse, etc.) dans le but de profiter du caractère cumulatif d'une plate-forme collaborative de stockage[10]. Après avoir présenté les traits essentiels de ce modèle et sa signification pour la définition d'une 'donnée' historique, je vais le comparer avec d'autres modèles existants afin d'en mettre en évidence les spécificités. Enfin, j'aborderai la question de l'interopérabilité des données produites selon différents modèles et des conditions de leur utilisation pour la recherche en histoire.

---

7 MARROU Henri-Irénée, « Comment comprendre le métier d'historien », in *L'histoire et ses méthodes*, SAMARAN Charles (dir.), Paris, Editions Gallimard, 1961, p.1465-1540 : 1494-1500.

8 BIZIÈRE Jean-Maurice et VAYSSIÈRE Pierre, *Histoire et historiens: Antiquité, Moyen-Âge, France moderne et contemporaine*, Paris, Hachette, 1995, p. 156-158.

9 Le projet SyMoGIH : http://symogih.org ; le Pôle histoire numérique du LARHRA : http://larhra.ish-lyon.cnrs.fr/pole-histoire-numerique.

10 BERETTA Francesco et VERNUS Pierre, « Le projet SyMoGIH et la modélisation de l'information : une opération scientifique au service de l'histoire », in *Les Carnets du LARHRA* 1(2012), p. 81-107 (http://halshs.archives-ouvertes.fr/halshs-00677658).

Je me limiterai ici au seul niveau des données structurées, c'est-à-dire au niveau de données construites grâce à un modèle sémantique qui en explicite le sens et qui les rend intelligibles et immédiatement utilisables par d'autres une fois que le modèle est partagé[11]. Il ne s'agit pas d'exclure par principe les autres niveaux que vous avons évoqués —car la relation entre données structurées et non structurées (textes, images, enregistrements sonores, etc.) est importante— mais il vaut mieux limiter le sujet sur lequel nous allons nous concentrer : comment permettre l'interopérabilité entre données historiques structurées ? Afin d'éviter toute confusion, j'utilise le terme *connaissance* pour ce qui relève du travail de l'historien et que je réserve le terme *donnée* à une *connaissance structurée par un modèle informatique*.

Les principes de la connaissance historique

Dans les ouvrages consacrés à la méthode en histoire on peut trouver quelques pistes de réponse à la question « qu'est-ce qu'une 'donnée' historique ? » en particulier autour de la notion de « fait historique » dont l'usage s'est établi avec l'école méthodique à la fin du XIXe siècle. Le fait historique est une *connaissance concernant le passé que l'historien tire des sources grâce aux procédés la méthode critique*. Les sources, qu'il s'agisse de textes ou de tout autre type d'objet ou de support matériel, iconographique, oral, etc., sont parfois appelées témoignages ou traces : « Un fait n'est rien d'autre que le résultat d'un raisonnement à partir de traces suivant les règles de la critique […]. [… il s'agit d'] affirmations vraies parce qu'elles résultent d'une élaboration méthodique, d'une reconstitution à partir de traces[12]. » Apparaissent ici deux éléments essentiels : d'une part, la *mention de la source* est indispensable à la vérification de la validité de la connaissance historique, elle est une condition de sa traçabilité ; d'autre part, les connaissances ne découlent pas 'spontanément' des sources, elle résultent de l'application d'une *méthode complexe* (critique externe et interne, critique de sincérité et d'exactitude, etc.) par un expert : l'historien[13].

A quel moment de son activité de recherche l'historien produit-il des 'données', c'est-à-dire des *connaissances concernant le passé* ? Une présentation rapide de sa manière de travailler permettra de répondre à cette question. Lorsqu'il s'occupe d'un objet historique —par exemple de l'état de l'astronomie au début du XVIIe siècle, ou des migrations de populations pour des raisons économiques ou de persécution religieuse— le chercheur se renseigne tout d'abord sur les travaux de ses prédécesseurs consacrés aux sujets qu'il veut étudier. Ayant constaté une lacune ou un aspect

11 EVANS Colin, SEGARAN Toby et TAYLOR Jamie, *Programming the Semantic Web,* Sebastopol (CA), O'Reilly, 2009, chapitre 1.

12 PROST Antoine, *Douze leçons sur l'histoire*, Edition augmentée, Paris, Éditions du Seuil, 2010, chapitre 3. Les faits et la critique historique, p.65.

13 *Ibid.*, p. 55-60.

à approfondir, il construit une problématique, un questionnement qui va le guider au cours de sa recherche. Il choisit ensuite ses sources et il en tire un *ensemble de connaissances* qui lui permettront de répondre à ses questions.

Ensuite, il formule des hypothèses, des pistes de réponse, qu'il va tester grâce aux connaissances qu'il a réunies. Si celles-ci sont suffisamment étoffées et couvrent une période de temps plus ou moins longue, il pourra comparer de manière diachronique l'évolution d'un phénomène en appliquant les méthodes d'analyse développées par l'histoire sérielle[14]. Si le volume et la qualité des connaissances sont suffisants, il pourra même appliquer des outils d'analyse statistique, de visualisation spatiale de l'information ou d'analyse de réseaux[15]. Au terme de cette exploration des connaissances récoltées l'historien sera obligé, dans certains cas, de revenir aux sources pour compléter un dossier lacunaire, ou pour en extraire de nouveaux aspects qu'il avait omis de prendre en considération. Ou alors l'analyse lui permettra de vérifier ses hypothèses et de produire une nouvelle synthèse, c'est-à-dire des connaissances d'un niveau d'abstraction plus élevé, des « faits globaux », des « faits de caractère complexe »[16] concernant —si on reprend les exemples mentionnés ci-dessus— la diffusion effective de l'héliocentrisme à une époque donnée ou les contours plus précis de l'impact social et économique des migrations.

Cette présentation rapide de la méthode de recherche en histoire permet de mettre en évidence, d'une part, le fait que la reconstitution du passé produit des connaissances historiques possédant *différents degrés d'abstraction* : du niveau des « données élémentaires, atomiques », généralement issues directement des sources, jusqu'à celui des « réalités d'ordre global, complexe », tel qu'il s'exprime dans les travaux de synthèse[17]. Ces différents niveaux de connaissance sont indispensables au travail de l'historien et ils doivent être pris en considération. D'autre part, s'il apparaît clairement que les « faits globaux », les connaissances issues de la synthèse, sont *construits,* ce principe s'applique également aux connaissances de type élémentaire car elles aussi découlent d'un questionnement, d'un regard porté sur la réalité du passé s'inscrivant dans une problématique de recherche[18].

A partir du même texte, ou de la même image, le chercheur va extraire des connaissances construites en fonction de différents questionnements : il peut s'intéresser, par exemple, au

14 FURET François, « Histoire quantitative et construction du fait historique », in *Annales. Économies, Sociétés, Civilisations* 26(1971)1, p. 63-75.
15 CELLIER Jacques et COCAUD Martine, *Le traitement des données en Histoire et Sciences Sociales. Méthodes et outils*, Rennes, Presses universitaires de Rennes, 2012.
16 MARROU Henri-Irénée, « Comment comprendre le métier d'historien », *op. cit.*, p. 1499-1500.
17 *Ibid*.
18 FURET François, « Histoire quantitative et construction du fait historique », *op. cit.*, p. 66-71.

caractéristiques physiques ou économiques d'un tableau, ou au sujet représenté dans sa dimension artistique ou culturelle, ou à une réalité historique décrite indirectement par l'image, telle la situation sociale du sujet représenté. Il est donc illusoire de vouloir extraire des sources les connaissances qu'elles contiennent grâce à un *life cycle* informatique presque 'automatique' : d'une part, le processus d'extraction doit nécessairement être paramétré et il le sera en fonction du questionnement implicite ou explicite de l'observateur ; d'autre part, seule une application fine de la méthode critique permet de produire des connaissances historiques de qualité éprouvée. A défaut de ce travail critique on possèdera un volume même très important de connaissances mais dont la pertinence et la fiabilité n'est pas assurée, et donc difficilement utilisables aux fins de la reconstitution de la réalité du passé.

L a *construction de connaissances* —qu'elles soient tirées de la lecture des travaux des prédécesseurs ou de l'analyse des sources en fonction d'un questionnement qui évolue sans cesse— apparaît donc comme un élément central du travail de l'historien[19]. Si le niveau d'abstraction peut être très différent, on relèvera toutefois que les connaissances issues directement des sources, ainsi que toute connaissance munie d'un certain degré d'objectivité, telle la reproduction précise de l'opinion d'un historien concernant tel sujet, représentent un socle important de l'édifice de l'histoire. De ce point de vue, le travail de description des objets conservés effectué par les bibliothécaires, les archivistes et les conservateurs des biens culturels, ainsi que l'identification des auteurs et des éléments significatifs de l'histoire de ces objets, relève de la même méthode et conduit à la production de quantité de connaissances qui sont précieuses pour la recherche historique et qui sont souvent d'excellente qualité. Il est donc important d'élargir la question de l'interopérabilité aux connaissances produites dans le domaine de la gestion des biens culturels.

Il nous reste enfin à mentionner deux difficultés que rencontre souvent l'historien, en particulier lorsqu'il s'intéresse à des époques reculées dans le temps : la datation des connaissances et l'incertitude, voire la contradiction entre les témoignages. La reconstitution de la temporalité, dimension essentielle de la recherche historique, est souvent délicate : grâce à la chronologie, science auxiliaire de l'histoire, il est possible de synchroniser avec le calendrier actuel —dit grégorien du nom du pape Grégoire XIII qui l'institua en 1582— les événements dont les traces se réfèrent à des ères ou à des calendriers qui ne sont plus en usage[20]. Toutefois l'incertitude reste parfois et la date retenue sera tronquée et limitée à la seule année, ou à l'année et au mois, ou elle sera approximative, c'est-à-dire qu'elle se situera à l'intérieur d'une fourchette de temps plus ou moins large, reconstituée à l'aide de la méthode critique.

19 PROST Antoine, *Douze leçons sur l'histoire*, *op. cit.*, chapitre 4. Les questions de l'historien

20 CORDOLIANI Alfred, « Comput, chronologie, calendriers », in : *L'histoire et ses méthodes, op. cit.*, p. 37-51.

L'incertitude peut également porter sur l'identifications des acteurs ou des autres objets dont parlent les sources, lorsque par exemple ils sont mentionnés uniquement par leur prénom. Ou encore il se peut que deux sources contiennent des affirmations non concordantes, voire contradictoires, concernant le même objet, qu'il s'agisse de la date ou du lieu d'un évènement, ou de son déroulement et des participants. Il revient au praticien de la méthode critique d'appliquer toute sa sagacité afin de découvrir ce qui s'est réellement passé ou de déceler les contradictions d'une interprétation erronée, ce qui produit une nouvelle connaissance, distincte de celle qui ressort immédiatement de la lecture de la source. Pour pouvoir réutiliser les connaissances historiques ainsi produites, il est indispensable d'indiquer leur origine et le degré de fiabilité de leur extraction des documents : telle est la condition de vérification de leur qualité et donc de leur réutilisation.

La production de données historiques : le modèle du projet SyMoGIH

Après avoir présenté quelques aspects de la méthode de travail de l'historien, avec toute sa richesse et sa complexité, nous pouvons poser la question qui est au coeur du chapitre : comment transformer les connaissances historiques en données structurées susceptibles d'être soumises à un traitement informatique en vue de les rendre intéropérables ? Il s'agit donc de s'interroger sur le *modèle* à adopter afin de prendre en compte les composantes essentielles de la méthode historiques : la construction des connaissances à partir d'un questionnement, le sourçage, la datation et la gestion de l'incertitude.

Une démarche classique, adoptée à cette fin par les projets en histoire depuis quelques décennies, consiste à utiliser les bases de données relationnelles et à proposer un modélisation qui utilise les formalismes Merise/ERD ou UML pour construire un modèle adapté à chaque projet[21]. Cette démarche soulève toutefois le problème de l'interopérabilité entre données produites à partir d e modèles relationnels différents car la sémantique de ces modèles est généralement liée à la problématique d'un projet précis même si les connaissances produites concernent des objets qui intéressent virtuellement d'autres chercheurs. En d'autres termes, la problématique introduit souvent un biais lors de la production des données.

A partir de ce constat nous nous sommes efforcés, dès les débuts du projet SyMoGIH, de développer un *modèle générique* permettant de mutualiser les données produites à partir de différentes approches disciplinaires. Deux principes essentiels ont guidé notre réflexion[22]. Il s'agit,

21 Pour un manuel d'introduction à la modélisation, voir : SOUTOU Christian, *UML 2 pour les bases de données*, Paris, Eyrolles, 2007. Pour un exemple d'application à un projet de recherche en histoire, voir : GAST Holger, LEUGERS Antonia et LEUGERS-SCHERZBERG August H., *Optimierung historischer Forschung durch Datenbanken. Die exemplarische Datenbank "Missionsschulen 1887-1940"*, Bad Heilbrunn, Verlag Julius Klinkhardt, 2010.

22 BERETTA Francesco et VERNUS Pierre, « Le projet SyMoGIH et la modélisation de l'information », *op. cit.*, p. 98.

d'une part, de la *séparation* entre la production des connaissances et la problématique de recherche qui guide leur collecte. Certes, toute connaissance trouve son origine dans un questionnement : toutefois, les réponses à celui-ci ne doivent pas être cherchées lors de la production des connaissances mais au moment de leur analyse. Les connaissances doivent donc être produites de la manière la plus objective possible, c'est-à-dire en évitant toute forme de biais lié à la problématique de recherche. Telle est la condition de leur réutilisation pour de nouveaux projets. D'autre part, il est nécessaire de procéder à une *atomisation*, c'est-à-dire à une décomposition des connaissances en éléments correspondants à des propositions simples et autonomes, afin de permettre leur réutilisation à partir de différents questionnements.

Ces deux principes nous ont amenés à *modéliser les connaissances historiques* en tant qu'*assertions qui parlent d'objets* —acteurs, institutions, lieux, concepts, etc.— et qui *les mettent en relation entre eux*. Cette approche de la modélisation inspirée du langage naturel amène à choisir un *modèle de données générique* car ce type de modèles offre un niveau d'abstraction suffisant pour permettre de traiter tout type de connaissance avec une structure relativement simple et concise [23]. Le coeur de notre modèle est représenté par des *assertions atomisées* qui mettent en relation des objets et que nous appellerons désormais *unités de connaissance* :

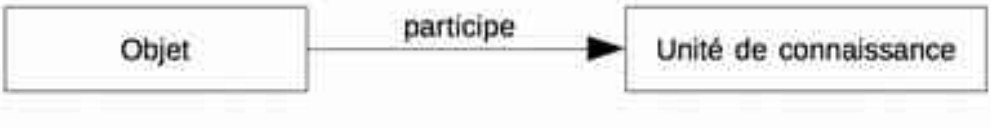

Fig. 1. Unités de connaissance et objets

La participation d'un objet à une unité de connaissance s'effectue selon des modalités différentes. Si, par exemple, la connaissance consiste dans l'assertion de l'existence d'une lettre, ce n'est pas la même chose que d'être son auteur ou son destinataire : les *rôles* des objets qui participent à cette connaissance seront donc différents.

Fig. 2. Les rôles

Les *rôles sont une composante importante des unités de connaissance* car ils indiquent de quelle manière chaque objet y participe. De plus, il permettent de préciser en quelle qualité et avec

23 http://en.wikipedia.org/wiki/Generic_data_model et la bibliographie qui y est citée.

quelles caractéristiques un objet intervient dans une assertion. Par exemple, l'auteur d'une lettre pourrait écrire à titre privé ou en tant qu'ambassadeur d'un prince : sa participation à la connaissance en sera modifiée, de même que l'interprétation de la lettre. Aussi, lors de l'achat d'un certain nombre de lettres par un collectionneur, ou par un organisme public, leur nombre pourra être indiqué en tant que propriété du rôle qui associe le concept 'lettre' à l'unité de connaissance 'achat', tandis que le prix de vente sera renseigné grâce à un autre rôle en lui associant la valeur et l'unité de mesure du montant de la transaction. Ce procédé permet un stockage simple de données quantitatives ou qualitatives de tout type.

Etant donné que le sujet de ce chapitre est l'interopérabilité des données, j'utiliserai le formalisme du *Resource Description Framework* (RDF) et, surtout, du *RDF schema* (RDFS) — vocabulaire basique de modélisation de données RDF[24]— pour exprimer le modèle du projet SyMoGIH. En complément, j'aurai également recours à un nombre restreint de termes du *Web Ontology Language (OWL)*[25]. Dans le formalisme retenu (Fig. 2), Objet, Rôle et Unité de connaissance (UC) sont trois ressources —au sens RDF— de type *rdfs:Class* qui sont associées entre elles par deux instances de la classe *rdf:Property* : 'a pour rôle' et 'appartient à'. Si on compare les Fig. 1 et 2, on constate que la classe Rôle représente une *réification* de la propriété 'participe', c'est-à-dire une matérialisation de l'association 'participer' qui se présente comme association à n dimensions ou n-aire[26]. Grâce à la réification, l'association est transformée en classe susceptible de posséder à son tour toute une série de propriétés, telle la qualité de l'auteur d'une lettre, le nombre des lettres achetées ou l'unité de mesure et la valeur du prix. La réification de la propriété 'participe' sous forme de la classe Rôle permet ainsi d'exprimer de manière concise les caractéristiques et les nuances de la participation de chaque objet à une UC précise.

Étant donné qu'une unité de connaissance est conçue dans ce modèle en tant qu'assertion atomisée qui en relation des objets, elle sera tout d'abord exprimée sous la forme d'un *libellé* : par exemple, « Lettre de Piero Dini à Galileo Galilei, Rome, 7 mars 1615 », ou « Johannes Kepler naît à

24 Les recommandations du consortium W3C constituent une excellente introduction aux différents standards liés au web des données: RDF : http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/ ; RDFS : http://www.w3.org/TR/2014/REC-rdf-schema-20140225/ . Pour une introduction à la modélisation en RDFS, voir : ALLEMANG Dean et HENDLER James, *Semantic web for the working ontologist. Modeling in RDF, RDFS and OWL*, second edition, Morgan Kaufmann Publishers, Waltham (MA), 2011.

25 Nous nous référens à la version OWL DL, cf. *OWL Web Ontology Language Reference* http://www.w3.org/TR/2004/REC-owl-ref-20040210. Voir également les autres documents auxquels renvoit cette page.

26 La réification est à entrendre ici non dans le sens de la réification d'un triplet RDF mais d'une manière de modéliser les relations n-aires selon le procédé présenté dans le document du consortium W3C *Defining N-ary Relations on the Semantic Web* (2006) http://www.w3.org/TR/swbp-n-aryRelations/, en particulier dans le sens de la première méthode proposée par ce document. À l'origine de la méthode SyMoGIH, la réification des rôles a été modélisée grâce à un modèle générique formalisé avec le langage Merise, cf. BERETTA Francesco et VERNUS Pierre, « Le projet SyMoGIH et la modélisation de l'information », *op. cit.*

Weil der Stadt le 27 décembre 1571 ». Grâce aux rôles on pourra associer les objets concernés à l'UC —dans nos exemples des acteurs et des lieux— et le sens de l'association sera exprimé par le *texte* de l'assertion. Si on s'arrêtait là on ne disposerait que de données semi-structurées car une partie du sens serait exprimé sous forme de textes, certes compréhensibles par les humains mais inexploitables par des requêtes ou des mécanismes d'inférence. Afin de disposer de données structurées, il faudra donc soit créer des sous-classes de l'UC —par ex. une classe Lettre ou une classe Naissance— qui précisent le sens des différents types de connaissances, soit définir une typologie des unités de connaissance sous forme d'une classe Type d'UC. Ces deux procédés sont équivalents au point de vue ontologique —car associer un type à chaque unité de connaissance revient virtuellement à définir des sous-classes de la classe UC— mais la création d'une ressource de type *rdfs:Class* regroupant les instances de la typologie, appelée Type d'UC, permet d'écrire un modèle plus concis.

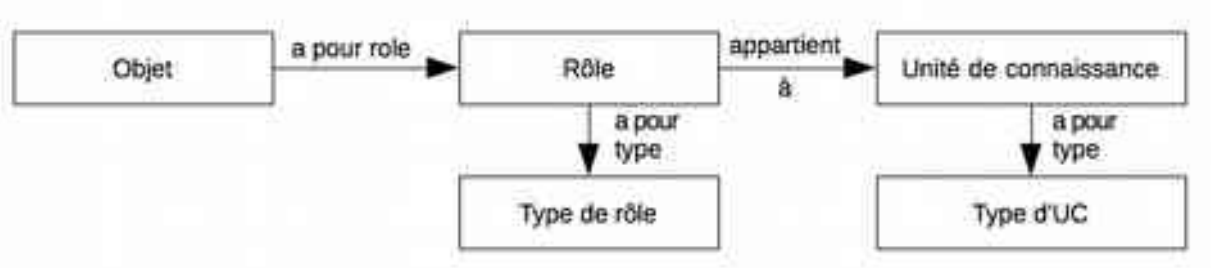

Fig. 3. Les Types d'UC et les Types de rôles

Cette deuxième voie a été retenue dans le modèle SyMoGIH car elle s'adapte mieux à la démarche collaborative et cumulative du projet. Au début, il n'existait que le modèle générique qui était pour ainsi dire 'vide'. Les instances des Types d'UC, par exemple Lettre ou Naissance, sont progressivement créées par les historiens —en tant que spécialistes du domaine— afin d'expliciter le modèle adopté pour construire chacune des connaissances qu'ils souhaitent stocker dans le système d'information. Ces instances du modèle générique ne définissent pas de quelle manière on doit traiter, dans l'absolu, une 'lettre' ou une 'naissance'. Leur fonction est de *documenter le modèle retenu pour la construction de chaque type de connaissance* grâce à un texte qui en explicite le sens et, éventuellement, le contexte, c'est-à-dire le questionnement qui se trouve à son origine.

En même temps que les types d'UC, les historiens participant au projet créent les instances de la classe Type de rôle afin de préciser à quel titre un objet participe à une unité de connaissance, par exemple en tant qu'auteur ou destinataire d'une lettre (Fig. 3). Comme un type de rôle peut intervenir dans plusieurs types d'UC, une classe supplémentaire a été introduite, la classe Composante du type d'UC (*sym:KnowledgeUnitTypeComponent*) (Fig. 4). Elle associe à chaque type d'UC ceux parmi les types de rôles qui sont susceptibles d'intervenir dans cette connaissance,

tout en fournissant une définition qui précise pour chacun d'entre eux le sens spécifique qui lui revient dans ce contexte. Elle permet également de préciser quelles sont les propriétés quantitatives ou qualitatives qui sont admises, ou requises, afin d'indiquer en quelle qualité et avec quelles caractéristiques un objet intervient dans l'unité de connaissance.

L'*application du modèle SyMoGIH aux connaissances historiques les transforme en données structurées* : l'assertion atomisée qu'on souhaite retenir est désormais qualifiée par un Type d'UC qui en explicite le sens tandis que les différents rôles, spécifiés par un Type de rôle, indiquent de quelle manière et avec quelles propriétés chaque objet intervient dans la connaissance. Quant au libellé de l'UC, il devient facultatif car le *sens de la connaissance est explicité par les instances du modèle générique*, les types d'UC et les types de rôles qui leur son associés. Les définitions fournies pour les instances de ces deux classes ont une importance capitale pour la compréhension du sens des données : par conséquent, elles sont exposées publiquement sur le site du projet SyMoGIH car elles *documentent la construction des connaissances* et représentent le fondement de la réutilisation des données par d'autres historiens[27].

En adéquation avec la méthode de recherche en histoire présentée précédemment, le niveau d'abstraction des connaissances historiques susceptibles d'être transformées en données peut être très divers : on pourra stocker des connaissances élémentaires issues directement des sources ou celles issues des travaux de synthèse des prédécesseurs, formulant une interprétation d'un phénomène historique complexe. La généricité du modèle retenu ne pose aucune limite de ce point de vue à condition de respecter les principes essentiels de la méthode que sont l'atomisation des connaissances, la poursuite du degré le plus élevé possible d'objectivité et la documentation précise de la sémantique des connaissances grâce à une définition explicite des instances des classes Type d'UC, Type de rôle et Composante du type d'UC.

27 Voir le site du projet : http://symogih.org .

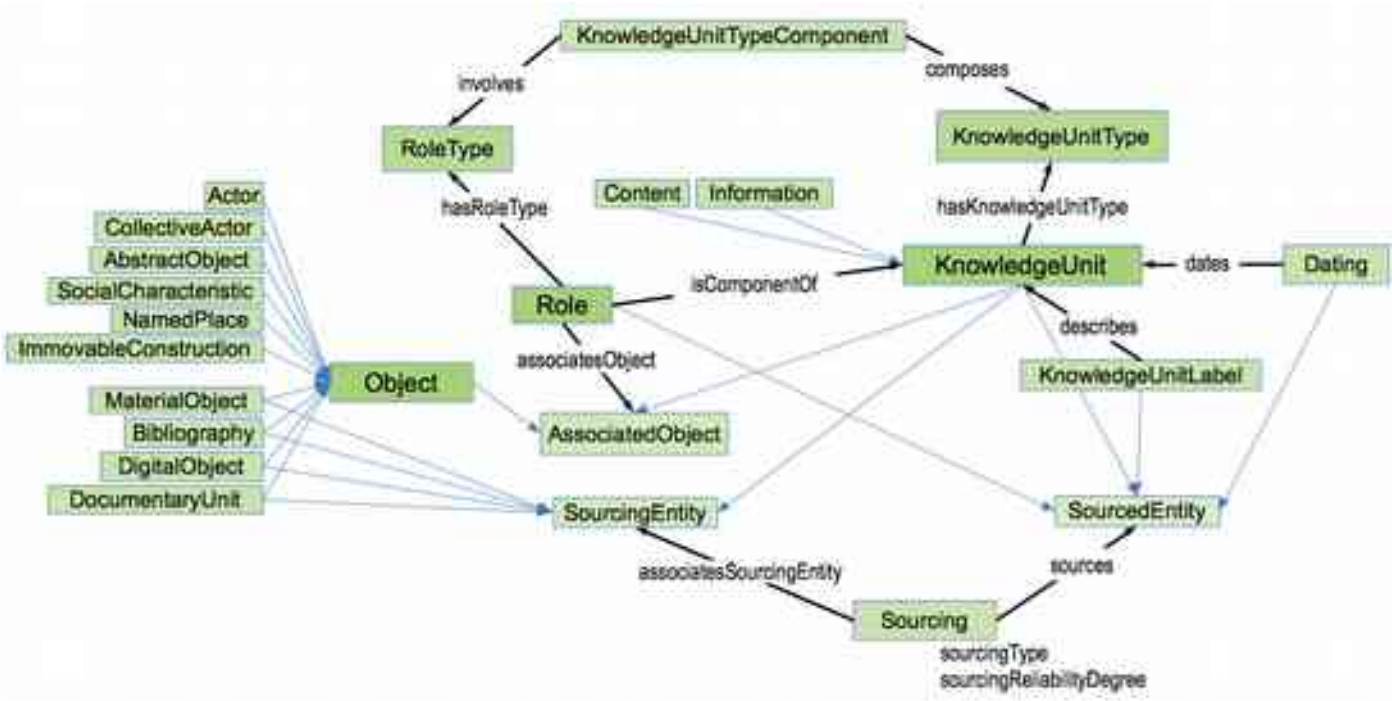

Fig. 4. Représentation simplifiée de l'ontologie du projet SyMoGIH

Au centre de la représentation sous forme de graphe de l'ontologie du projet SyMoGIH exprimée en RDFS (Fig. 4)[28], on retrouve les éléments essentiels du modèle générique, c'est-à-dire les classes Objet, Rôle et Unité de connaissance (UC), tandis que dans la partie supérieure on reconnaît les classes Type d'UC, Type de Rôle et Composante du type d'UC. Les relations entres classes et sous-classes, au sens de la propriété *rdfs:subClassOf*, sont représentées par des arcs fins en bleu, tandis que les associations entre classes, c'est-à-dire les propriétés au sens *rdf:Property*, sont représentées par des arcs en noir qui indiquent le nom et la direction de l'association. L'ensemble des ressources représentées sur le graphe appartient à l'espace de noms *http://symogih.org/ontology/* (abréviation usuelle 'sym'). L'identifiant correspondant sous forme de *Uniform Resource Identifier (URI)* est produit par concaténation selon la forme *http://symogih.org/ontology/Object* si on considère l'exemple de la classe Objet.

Relevons enfin que cette version simplifiée du graphe représente uniquement —à une exception près— les associations entre classes (au sens de *owl:ObjectProperty)* et non les propriétés qui associent à chaque classe les textes ou autres types de valeurs littérales (*owl:DatatypeProperty*) dont quelques-unes seront mentionnées ci-dessous. Concernant les propriétés représentées, la classe source de l'arc est définie en tant que *rdfs:Domain* alors que la classe cible est son *rdfs:Range :* les

28 Ontologie SyMoGIH version 0.2 en cours de validation par les membres du projet en vue de sa publication.

propriétés des classes sont donc héritées par les sous-classes par inférence. Ceci signifie que, par exemple, une instance de la classe Bibliographie (*sym:Bibliography*) peut être associée à un sourçage par une propriété 'associer l'entité qui source' (*sym:associatesSourcingEntity*).

Les spécificités des données historiques

Il nous reste à présenter les autres parties du modèle SyMoGIH qui permettent de rendre compte de toute la complexité des connaissances historiques. Comme le montre le modèle (Fig. 4), la classe Objet est articulée en sous-classes qui regroupent les différents types d'objets historiques : acteurs individuels, collectivités, objets abstraits (concepts), caractères sociaux (objets abstraits concernant les différents aspects de la vie sociale), etc. Il n'est pas possible dans le cadre de ce chapitre de détailler davantage la définition des types d'objets[29] : relevons simplement que nous avons choisi de les traiter en tant que sous-classes de la classe Objet, classe abstraite qui ne possède aucune instance, c'est-à-dire aucun objet qui lui serait directement rattaché.

Ce choix —qui est différent de celui adopté pour les types d'UC, traités en tant que types et non comme sous-classes— attribue une consistance 'forte' à chaque sous-classe d'objets, s'exprimant dans l'identifiant semi-sémantique retenu: 'Actr' pour les acteurs, 'CoAc' pour les collectivités, etc. Les instances des sous-classes seront donc identifiées par un URI sous la forme *http://symogih.org/resource/Actr195* (abréviation usuelle 'syr:Actr195') qui permet de reconnaître immédiatement que l'objet Johannes Kepler, identifié par ce URI, appartient à la sous-classe Acteur. De plus, les sous-classes possèdent quelques propriétés spécifiques aux objets qu'elles regroupent —par exemple la géolocalisation pour les lieux— qui ne sont toutefois pas représentées sur le modèle simplifié (Fig. 4).

Le découpage des sous-classes a été opéré de la manière la plus objective possible afin d'éviter le biais qu'on rencontre souvent et qui consiste à traiter sous forme de classes quelques propriétés spécifiques des objets, ce qui entraîne la création de vocabulaires complexes : notre expérience de modélisation montre qu'il est préférable en vue de l'interopérabilité de traiter les propriétés des objets en tant que connaissances qu'on leur associe. Le classement proposé vise donc à être le plus possible indépendant de toute problématique de recherche et à regrouper les objets dans un nombre limité de classes sans intersections (*owl:disjointWith*). Notons enfin que la classe Objet possède un nombre limité de propriétés associant aux objets des valeurs littérales (*owl:DatatypeProperty*) dont la fonction est de fournir quelques éléments de base pour leur identification, tel un ou plusieurs noms, une date de début et de fin, un descriptif succinct. Toute autre propriété —y compris un

29 Cf. la partie consacrée aux objets et aux connaissances qui les concernent sur le site http://symogih.org .

traitement précis de l'évolution de leur nom, ou de la date de début et de fin, etc.— sera traitée sous forme d'UC.

En effet, selon la méthodologie de la connaissance historique toute assertion de l'historien doit être associée à une ou plusieurs sources qui garantissent sa traçabilité et sa vérification ou falsification. La classe Sourçage (*sym:Sourcing*) a pour fonction d'associer à chaque unité de connaissance un ou plusieurs objets susceptibles de faire fonction de source —par exemple un manuscrit, un tableau, un dictionnaire ou un monographie— et qui appartiennent de ce fait à la classe *sym:SourcingEntity*. Selon l'ontologie SyMoGIH, il est possible de sourcer non seulement le corps de la connaissance, représenté par la classe UC, mais encore les rôles, datations et libellés, c'est-à-dire toutes les *composantes* de la connaissance. Les classes correspondantes appartiennent donc à la classe abstraite *sym:SourcedEntity* qui regroupe toutes les parties de la connaissance susceptibles d'être sourcées. La classe Sourçage résulte de la réification de la propriété 'sourcer', ce qui permet d'associer l'indication de la référence précise dans la source —par exemple le numéro du tome ou les pages, une URL, etc.— ou tout autre élément fournissant une précision quant à l'identification de l'entité sourçante.

Deux propriétés de type *owl:DatatypeProperty* —les seules visibles sur le modèle simplifié en raison de leur importance— permettent de préciser la qualité du sourçage en rapport avec la connaissance. La propriété 'type de sourçage' (*sym:sourcingType*) permet d'expliciter de quelle manière la source a été utilisée lors de la production de la connaissance. Elle prend les valeurs décroissantes 3, 2, 1 correspondantes respectivement à un sourçage 'littéral', 'par déduction directe' et 'par déduction indirecte', afin d'indiquer que l'assertion retenue se trouvait comme telle dans la source ou qu'elle en a été déduite de manière plus ou moins directe. On peut ainsi distinguer entre les sources dont l'analyse fonde explicitement ou seulement implicitement la connaissance. La valeur 0 indique que la propriété n'est pas renseignée. Si on connaît une source qui contient une assertion contradictoire par rapport à celle qui a été retenue en appliquant la méthode critique — parce qu'on dispose d'un document de plus grande fiabilité— on pourra créer un sourçage dédié à la source contradictoire avec une valeur négative pour la propriété 'type de sourçage'. On enregistrera ainsi un sourçage contradictoire 'littéral', 'par déduction directe' et 'par déduction indirecte', avec les mêmes valeurs mais de signe négatif, ce qui fournit un paramètre permettant de comparer les sourçages.

L'autre propriété porte sur le 'degré de fiabilité du sourçage' (*sym:sourcingReliabilityDegree*) par rapport à l'assertion exprimée par l'unité de connaissance. Elle prend trois valeurs décroissantes (3, 2, 1) qui correspondent au caractère 'certain', 'probable' ou 'incertain' du *sourçage de l'assertion*

afin d'indiquer quelle est la valeur attribuée par l'historien —à l'aune de sa méthode— à telle instance du Sourçage en tant que fondement de l'unité de connaissance concernée. Les raisons précises de ce choix, tout comme celles concernant le type de sourçage, peuvent être explicitées sous forme de texte (*rdfs:comment*). L'ontologie SyMoGIH ne propose donc pas un paramètre de certitude concernant la connaissance elle-même mais elle combine les différents sourçages, avec leur type et leur fiabilité respectives, afin de *documenter et d'expliciter la qualité de la connaissance produite* en accord avec le principe épistémologique qui fonde la méthode historique : la référence à une source. Nous verrons plus loin d'utilité de cette approche pour la question de l'interopérabilité.

Ces mêmes propriétés peuvent être utilisées pour fournir des précisions concernant la qualité du sourçage de chaque instance des classes Datation ou Rôle, ce qu'on effectue en les associant directement aux instances concernées de la classe Sourçage. En effet, si les instances des classes qui composent l'UC —Rôle, Datation et Libellé— héritent, par définition, du sourçage de l'unité de connaissance à laquelle elles sont associées, elles appartiennent également à la classe *sym:SourcedEntity* et elles peuvent donc être sourcées individuellement afin de préciser, si souhaité, la fiabilité et le type du sourçage de chacune.

La classe Rôle présente encore une particularité : elle possède deux propriétés supplémentaires qui permettent de gérer l'incertitude lors de l'identification des objets associés. La propriété 'certitude d'identification de l'objet' (*sym:associatedObjectIdentificationCertainty*), avec les trois valeurs 'certaine', 'probable', 'incertaine', permet d'indiquer avec quel degré de certitude on peut associer une instance précise de la classe Objet à la connaissance. Une propriété 'libellé de l'objet selon la source' permet de saisir la dénomination de l'objet telle qu'elle figure dans le document qui source le rôle : en cas de doute concernant l'identification d'un objet on dispose ainsi de la teneur exacte de sa dénomination sans devoir revenir à la source. Mais comment procéder si on souhaite enregistrer les différentes dénominations du même objet dans plusieurs sources qui concernent la même instance de la classe UC, ainsi que les spécificités de chacune d'elles ?

Pour répondre à cette question, il est nécessaire d'expliciter la distinction essentielle que nous avons introduite entre les classes Contenu (*sym:Content*) et Information (*sym:Information*), les deux étant des sous-classes de la classe UC (*sym:KnowledgeUnit*). Étant donné qu'une connaissance exprime une assertion de l'historien, deux cas de figure sont envisageables dont le sens est fondamentalement différent. D'une part, il y a les connaissances dont la finalité est de reproduire fidèlement les assertions que contient une source, qu'il s'agisse d'un document ancien ou d'un texte issu de la plume d'un autre historien : dans ce cas, la finalité du chercheur —qui se limite à la

fonction de 'rapporteur'— est de reproduire fidèlement le contenu de la source, et ce même s'il sait par ailleurs que l'assertion en question n'est pas correcte, voire fausse. Ce premier type d'assertions représente dans le modèle SyMoGIH une instance de la classe Contenu, c'est-à-dire une connaissance qui par définition ne peut avoir qu'*une seule source* puisque sa finalité est précisément d'exprimer le contenu de celle-ci.

D'autre part, il y a les connaissances qui résultent du travail critique de l'historien qui, ayant comparé différentes sources en leur appliquant l'outillage de sa méthode, arrive à une affirmation qui a un degré élevé de probabilité, voire de véracité. Le statut épistémologique des assertions de ce type, qui appartiennent à la classe Information (*sym:Information*) dans le modèle SyMoGIH, est celui, par exemple, des propositions que contient une notice de dictionnaire : à moins de découvrir une nouvelle source fiable qui entrainerait une falsification de la connaissance, ou d'une relecture critique plus approfondie des sources disponibles, l'assertion de l'historien est considérée comme exprimant un *aspect avéré de la réalité historique* ou —pour reprendre le concept utilisé ci-dessus — un « fait historique ». Dans ce cas, une unité de connaissance peut être virtuellement associée à plusieurs instances de la classe Sourçage. Etant donné le *statut épistémologique essentiellement différent* des assertions appartenant respectivement aux classes *sym:Content* et *sym:Information*, toute instance de la classe UC doit être nécessairement rattachée à l'une de ses deux sous-classes et, par conséquent, la classe UC est une classe abstraite sans instances propres.

C'est donc en utilisant plusieurs instances de la classe Contenu qu'on pourra enregistrer les spécificités de différentes sources concernant la même unité de connaissance. Comme la classe *sym:KnowledgeUnit* appartient elle aussi à la classe *sym:SourcingEntity*, on pourra utiliser une ou plusieurs unités de connaissance de type Contenu pour sourcer une instance de la classe Information, tout en spécifiant les différents paramètres de fiabilité grâce aux instances de la classe Sourçage. On pourrait également déduire une nouvelle connaissance de type Information à partir d'une ou plusieurs instances de la classe *sym:Information*, ce qui sera documenté grâce aux sourçages respectifs. À noter enfin qu'une unité de connaissance peut intervenir dans une autre en tant qu'objet associé (classe *sym:AssociatedObject*) : par ex. une unité de connaissance qui relate un événement historique peut-être associée via un rôle à une autre connaissance qui enregistre l'interprétation de l'événement par tel auteur.

Rappelons que dans l'ontologie SyMoGIH toute connaissance est conçue en tant qu'assertion de l'historien : la réification de l'ensemble des composantes de la connaissance permet d'associer à chacune d'entre elles les métadonnées concernant sa création et sa dernière modification. Ces propriétés sont partagées plus largement par l'ensemble des classes de l'ontologie car celles-ci sont

toutes définies en tant que sous-classes de la classe Entité (*sym:Entity*) dont elles héritent les propriétés (Fig. 5).

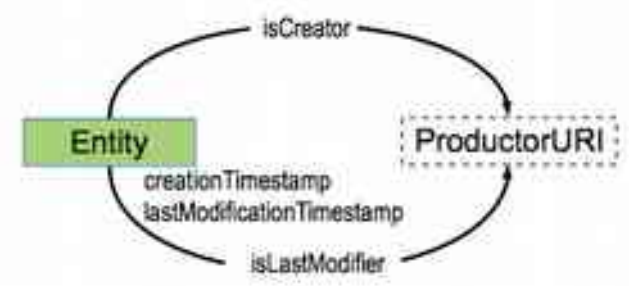

Fig. 5. Métadonnées de toute entité appartenant à l'ontologie SyMoGIH

On peut ainsi renseigner le créateur et le dernier modificateur de chaque instance de n'importe quelle classe grâce à deux propriétés qui lui associent un URI permettant d'identifier le producteur. Comme la définition de ces URI peut s'effectuer dans n'importe quel système d'identifiants, la symbologie de la classe 'URI du producteur' est différente que celle des autres classes. Une même personne physique peut ainsi figurer en tant que producteur de connaissances à différents titres, en fonction des différents projets auxquels elle participe : on pourra donc établir un lien avec le contexte et, si souhaité, l'institution qui a financé la production des données.

Enfin, la temporalité de la connaissance est traitée non par une date mais par une *datation*, c'est-à-dire par une ou plusieurs instances de la classe Datation (*sym:Datation*) qui indiquent de quelle manière a été déterminée la date de l'unité de connaissance ou la période concernée. Dans chaque datation, le format de la date est exprimé selon la norme ISO 8601 qui adopte le calendrier grégorien à partir du 15 octobre 1582[30]. Pour la période précédente, c'est-à-dire égale ou inférieure au 4 octobre 1582, nous proposons d'utiliser le calendrier julien en ajoutant le préfixe 'B' pour les dates avant l'ère commune. La propriété 'type de datation', qui prend les valeurs 'date unique', 'date de début', 'date intermédiaire', 'date de fin' permet de distinguer entre les unités de connaissance dont la datation est ponctuelle de celles qui correspondent à une période. Pour exprimer une durée dans le temps on indiquera le début et la fin de la période en utilisant deux instances de la classe Datation, ou une datation intermédiaire si les dates extrêmes ne sont pas connues. Chaque type de datation peut-être dédoublé afin de définir une fourchette entre deux dates au sein de laquelle se situe, par exemple, la date de début. Dans les cas les plus complexes on disposera donc de quatre datations qui indiquent que l'unité de connaissance concerne une durée, ainsi que deux fourchettes de temps à l'intérieur desquelles se situent respectivement le début et la fin de la période.

Chaque datation est caractérisée par une propriété qui en indique la certitude à l'aide des valeurs

30 http://fr.wikipedia.org/wiki/ISO_8601 .

'certaine', 'reconstituée', 'postulée', comportant un degré décroissant de précision. Le sourçage de chaque datation s'effectue conformément aux principes concernant la classe *sym:SourcedEntity* présentés ci-dessus. La teneur et le système de référence chronologique original de la date sera retenu, si souhaité, sous forme de texte (*rdfs:comment*) : une structuration plus détaillée n'a pas semblé nécessaire en raison de la variété des cas de figure et du fait que ce choix n'impacte pas l'interopérabilité, centrée autour du format de la date selon la norme ISO, des types de datation et des paramètres d'incertitude.

Ontologies et données historiques

Dans les pages qui précèdent, j'ai voulu mettre en évidence —au delà de la présentation de l'ontologie d'un projet précis— toute une série de questions qui se posent lorsqu'on veut transformer les connaissances historiques en données. Je l'ai fait en essayant de prendre en compte autant que possible les exigences qu'impose l'application fine de la méthode critique en termes d'explicitation de la construction des connaissances, de traçabilité, de datation, de gestion de l'incertitude et de la contradiction entre les sources. Avant d'aborder la question de l'interopérabilité des données produites avec ce type de modèle, il est nécessaire de le comparer avec d'autres ontologies existantes afin d'en mettre en évidence les spécificités.

Une des finalités des ontologies de haut niveau (*top-level ou foundational ontologies*) est de permettre une clarification des concepts utilisés par les ontologies qu'on souhaite comparer. La *Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)*[31], développée au début des années 2000, a été conçue comme moyen d'étudier les structures essentielles du langage naturel en tant que compréhension de la réalité —réalité qui reste dans son essence au-delà de l'ontologie et qui n'est pas modélisée comme telle— à partir des conditionnements culturels et des conventions sociales qu'exprime le langage[32]. DOLCE se situe donc dans la même perspective cognitive que l'ontologie SyMoGIH —qui décrit les assertions construites par les historiens— et représente par conséquent un point de référence important.

Quatre catégories principales (« categories of particulars ») sont prises en considération dans l'ontologie DOLCE : les « endurants » (entités qui subsistent avec la même essence dans le temps, tels les objets physiques, les concepts ou les humains) ; les « perdurants » (entités qui se développent dans le temps tout en se modifiant d'un instant à l'autre, tel les événements, les

31 Voir le résumé des résultats du projet: http://cordis.europa.eu/result/rcn/41438_en.html, ainsi que la présentation dans http://en.wikipedia.org/wiki/Upper_ontology#DOLCE_and_DnS .

32 MASOLO Claudio, BORGO Stefano, GANGEMI Aldo, GUARINO Nicola, OLTRAMARI Alessandro, *WonderWeb Deliverable D18 Ontology Library (final)*, Trento, Laboratory For Applied Ontology, 2003, téléchargeable en version PDF depuis le site http://wonderweb.man.ac.uk/deliverables.shtml, p. 13.

processus ou les accomplissements) ; les « qualities » (caractéristiques inhérentes aux entités, telle leur couleur ou leur localisation dans l'espace) ; les « regions » (concepts abstraits qui permettent de caractériser l'espace et le temps, tel un secteur précis du spectre des couleurs ou la seconde en tant qu'unité de mesure du temps).

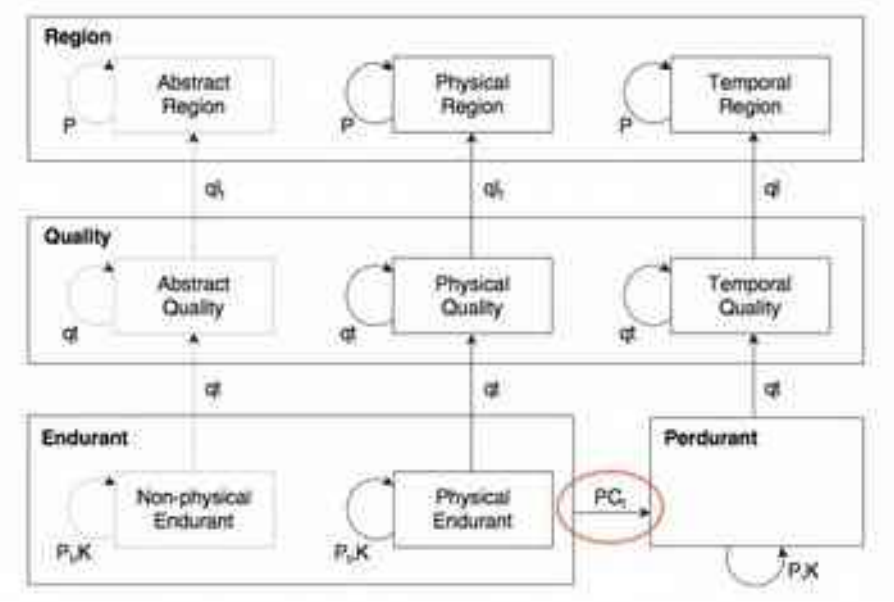

Figure 6. DOLCE : relations primitives entre les catégories de base[33]

La relation essentielle entre *endurants* et *perdurants* est celle de « participation » (« PC » dans la Fig. 6) : « an endurant 'lives' in time by *participating* in some perdurant(s). For example, a person, which is an endurant, may participate in a discussion, which is a perdurant »[34]. Les qualités, qui ne sont pas à comprendre en tant qu'universels —c'est-à-dire en tant que propriétés distinctes des entités, telle « la couleur rouge »— mais comme des caractéristiques inhérentes aux entités et inséparables de celles-ci —cette entité se présente avec une qualité que j'affirme se situer dans le spectre du rouge, sa « région »— permettent de décrire les entités et de les localiser dans l'espace et dans le temps.

Si de premier abord on reconnaît une homologie entre les ontologies DOLCE et SyMoGIH autour de la triade objet/*endurant*, rôle/*participation*, unité de connaissance/*perdurant*, l'articulation est, en fait, plus subtile. Du côté des objets, il n'y a pas de doute quant à l'équivalence de cette classe avec celle des *endurants*. En revanche, les *perdurants* sont considérés dans DOLCE comme phénomènes qui se déroulent dans le temps alors que la classe UC de SyMOGIH représente les assertions de l'historien décrivant les *perdurants* : en tant qu'assertions atomisées qui mettent en relation des objets, les *unités de connaissance* ne sont pas les phénomènes temporels eux-mêmes

33 *Ibid.*, p. 25
34 *Ibid.*, p. 16.

mais expriment une *construction linguistique qui les décrit*. Cette distinction, implicite dans DOLCE, est explicitée par l'ontologie *Descriptions and Situations ontology (DnS)* qui en représente un développement et qui introduit la classe *Description*, descriptif d'une situation. Cette classe est traitée en tant que sous-classe de *Endurant* du fait de sa nature d'entité abstraite individualisée[35].

Dans DnS apparaît également la notion de *functional role* concernant les parties des descriptions « that reify a functional property of DOLCE endurants (e.g. citzen or judge) ». Cette propriété réifiée relie les objets (*endurants*) et les instances de la classe *Description* afin de spécifier le sens de leur participation à la description : les *functional roles*, associés à la classe *Description* par la propriété *component-of*, se présentent ainsi —de manière analogue aux instances de la classe *sym:Role*— comme une *composante* de la description des *perdurants*[36]. Cette articulation ne semble pas être complètement stable dans l'écosystème DOLCE et a évolué dans l'ontologie DnS ainsi que dans l'ontologie *DOLCE+DnS Ultralite (DUL)*, une autre ontologie qui synthétise les précédentes[37]. Pour notre propos, il importe de relever la distinction essentielle entre l'assertion de l'historien, instance de la classe *sym:KnowledgeUnit*, et l'évènement ou autre phénomène temporel appartenant au passé, le *perdurant,* qui se situe au-delà de l'assertion même et reste 'insaisissable' : les instances de la classe *sym:Content*, issues chacune d'une source différente, ou celles de la classe *sym:Information*, qui synthétisent les contenus en une seule information grâce à l'application de la méthode critique, résultent d'une *construction qui décrit l'événement historique sans s'identifier à lui*.

Cette approche de l'UC, conçue en tant qu'assertion de l'historien, explique pourquoi l'ontologie SyMoGIH ne prévoit pas une classe équivalente à *Quality* dans DOLCE. Les propriétés des objets, telles leurs caractéristiques physiques ou leur localisation dans l'espace, sont décrites par les propriétés des rôles ou par des instances spécifiques de la classe UC parce que la *spécification d'une qualité* propre à un objet à une époque donnée se présente elle aussi comme *assertion critiquement fondées sur les sources*. Les unités de connaissance modélisent donc sous forme de relations entre objets —relations qualifiées par une typologie adéquate— la description des *perdurants* ou des *qualities* telle que l'historien la construit à partir de son questionnement. A relever que, si on admet que DOLCE modélise le langage naturel grâce à des catégories conçues en tant que « cognitive artifacts ultimately depending on human perception »[38], on pourrait conclure que les *perdurants* eux-mêmes —tout comme les UC— ne sont autre chose que des assertions concernant des 'faits' qui se déroulent dans le temps mais qui restent 'insaisissables' dans leur essence : dans ce cas, on

35 *Ibid.*, p. 96-98.
36 *Ibid.*, p. 102
37 http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite .
38 MASOLO Claudio *et al.*, *WonderWeb Deliverable D18 Ontology Library*, *op. cit.*, p. 13.

aurait une équivalence entre *perdurants* et unités de connaissance et, par conséquent, *Perdurant* et *Quality* seraient deux sous-classes de *sym:KnowledgeUnit.*

Cette clarification conceptuelle permet d'aborder la comparaison entre le modèle générique du projet SyMoGIH et quelques-unes parmi les ontologies qui modélisent les connaissances historiques, et plus précisément les *event ontologies,* l'ontologie des *factoids* et CIDOC-CRM. Parmi les premières, nous évoquerons les traits essentiels du *Simple Event Model (SEM)*, ontologie dont les auteurs se sont servis pour traiter, entre autres, des actes de piraterie ou les mouvements des bateaux dans un port. La visée du modèle est toutefois plus générique car il s'agit de traiter tout type d'événement, réel ou fictionnel, l'événement étant conçu de façon très large comme « everything that happens »[39].

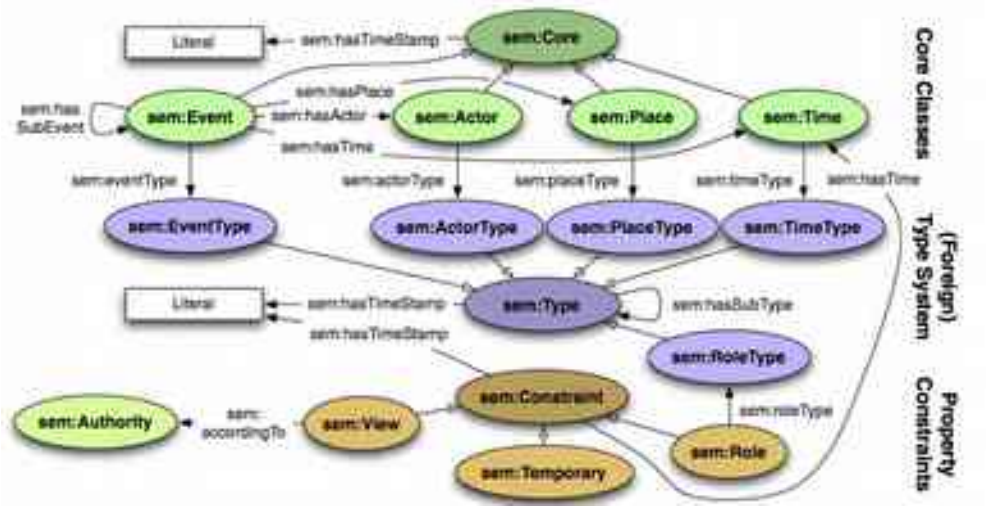

Figure 7. Les classes du *Simple Event Model (SEM)*[40]

Les quatre classes principales de l'ontologie (*core classes)* sont *Event*, *Actor*, *Place* et *Time*. Les acteurs peuvent être non seulement des personnes physiques mais encore tout type d'objet, par

39 VAN HAGE Willem Robert, MALAISE Véronique, SEGERS Roxane, HOLLINK Laura, SCHREIBER Guus, « Design and use of the Simple Event Model (SEM) », in *Web Semantics: Science, Services and Agents on the World Wide Web* 9 (2011), p. 128–136 : 129. Le PDF est disponible ici : http://www.websemanticsjournal.org/index.php/ps/issue/view/35.

40 *Ibid.*, p. 130.

exemple un bateau. Ils sont associés à l'événement par la propriété *sem:hasActor* qui correspond à la notion de 'participation' que nous avons rencontrée dans DOLCE et SyMoGIH. Pour spécifier à quel titre un acteur participe à un événement, SEM introduit une contrainte sous la forme d'une classe *sem:Role* qui s'applique à une réification de la propriété *sem:hasActor*[41]. Le type de chaque rôle est spécifié par une classe *sem:RoleType* dont la définition se fait dans une ontologie externe au SEM. En effet, la finalité de celui-ci est de permettre l'interopérabilité entre données issues de différentes sources : le modèle générique est donc défini par les classes de base (Fig. 7 partie supérieure) et par les contraintes des propriétés (partie inférieure), mais les instances des classes *sem:Type* (partie centrale) relèvent des ontologies d'origine des données, c'est-à-dire de classes ou propriétés définies à l'extérieur du SEM.

Ce souci d'interopérabilité explique le caractère extrêmement succinct du modèle ainsi que sa structure correspondante à celle d'une ontologie de haut niveau, ce qui est souligné par ses auteurs en proposant un alignement entre les classes du SEM et celles de DOLCE : les instances de la classe *sem:Actor* correspondent aux *endurants*, celles de *sem:Event* aux *perdurants,* alors que *sem:Place* correspond à *PhysicalRegion*. En même temps, les contraintes introduites dans SEM permettent une grande finesse dans la définition des temporalités qui peuvent être définies de manière spécifique pour chaque propriété grâce à la classe *sem:Temporary*. De plus, la gestion de points de vue différents sur chaque aspect de l'événement s'effectue grâce aux instances de *sem:View* : chacune de celles-ci peut être associée à une autorité par la propriété *sem:accordingTo* à laquelle revient la fonction d'assurer la traçabilité de l'information en spécifiant les sources des données.

SEM permet donc de modéliser une portion considérable des caractéristiques des connaissances historiques telles que nous les avons définies et, en vertu de son approche générique, il présente une structure assez proche du modèle SyMoGIH : les sous-classes des classes principales sont définies —en tant qu'instances de la classe *sem:Type*— par un typage souple qui vise à rendre interopérables les données issues de différents entrepôts. Plusieurs visions contradictoires des mêmes événements peuvent subsister, avec l'indication de leur source, sans toutefois qu'une pondération ou une analyse critique par le spécialiste puissent être enregistrées. SEM se situe par conséquent au niveau épistémologique de la classe Contenu (*sym:Content*) et ne permet pas de modéliser les connaissances issues de l'application de la méthode critique. Surtout, la restriction explicite de la classe principale à la notion d'événement exclut une portion importante des connaissances produites par l'historien : SEM peut donc apporter une contribution importante à la mise en place de

41 Trois manières différentes de gérer la réification sont proposées, cf. *ibid.*, fig. 2. Concernant les roles de l'ontologie SyMoGIH, nous en avons adoptée une quatrième qui semble être la plus adaptée au traitement des relations n-aires, voir ci-dessus, note 26.

l'interopérabilité d'une partie de données produites par la recherche mais ne permet pas de modéliser l'ensemble des connaissances historiques. Ces mêmes considérations s'appliquent à d'autres *event models* dont certains sont très développés, tel le *Event model F* fondé sur l'ontologie DUL[42], et d'autres limités aux propriétés essentielles des événements afin de faciliter l'interopérabilité, tel LODE[43].

Parmi les ontologies développées dans le contexte de la recherche historique, le modèle 'factoid-oriented' développé par le Departement of Digital Humanities du Kings's College de Londres mérite une attention particulière. Cette ontologie a été développée dans le contexte de différents projets ayant mis en place des bases de données prosopographiques allant de l'Antiquité au Moyen Age afin de disposer d'un modèle générique commun permettant l'interopérabilité. Le mot 'factoïde', qui ne semble pas être utilisé en français, se réfère à une assertion concernant un fait qui n'est pas suffisamment prouvée[44].

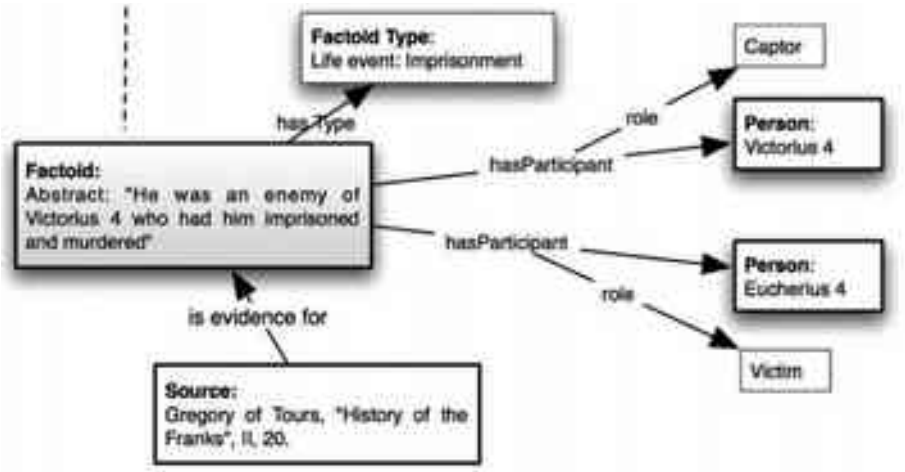

Fig. 8. Modèle générique de l'ontologie *Factoid*[45]

Selon les auteurs de ce modèle, un *factoid* reproduit une information qui se trouve dans une petite portion de texte, un « spot », et qui concerne quelques acteurs qui participent à différents titres. On reconnaîtra aisément l'homologie de ce modèle avec l'ontologie SyMoGIH : une assertion

42 Cf. http://ontologydesignpatterns.org/wiki/Ontology:Event_Model_F et http://www.weknowit.eu/content/model_events_based_foundational_ontology .
43 Cf. http://linkedevents.org/ontology/
44 PASIN Michele et BRADLEY John, « Factoid-based prosopography and computer ontologies: Towards an integrated approach », *Literary and Linguistic Computing*, Advance Access published June 29, 2013, disponible ici http://dsh.oxfordjournals.org/content/early/2014/12/02/llc.fqt037 . Cf. http://en.wikipedia.org/wiki/Factoid
45 *Ibid.*, p. 3

munie d'une source et définie par un type concerne différentes personnes qui jouent différents rôles définis par le modèle. Étant donné que la finalité du *factoid* est de reproduire l'information brute que contient la source, il est qualifiable de « souce driven »[46] et il a un statut épistémologique équivalent à celui de la classe *sym:Content*. Une comparaison avec l'ontologie SyMoGIH montre que, d'une part, seule une portion des connaissances produites par les historiens est prise en compte —les instances de la classe *sym:Information* ne peuvent pas être traitées avec ce modèle— et que, d'autre part, en dépit de l'homologie de sa structure essentielle, le modèle *factoid* n'a pas encore développé toute la généricité et la finesse permettant de traiter tout type d'assertion historique, en prenant en compte les paramètres de temporalité, d'imprécision et d'explicitation de la construction des connaissances.

En revanche, les auteurs du modèle *factoid* ont le mérite d'avoir poussé jusqu'au bout une réflexion sur l'interopérabilité avec les autres ontologies et en particulier avec CIDOC-CRM centrée sur l'analyse du processus de la connaissance historique. A cette fin, ils ont introduit une nouvelle classe, appelée « Document interprétation act », pour souligner un principe essentiel —présenté longuement dans ce chapitre— d'après lequel toute connaissance résulte d'une interprétation de la source, par conséquent d'une construction intellectuelle.

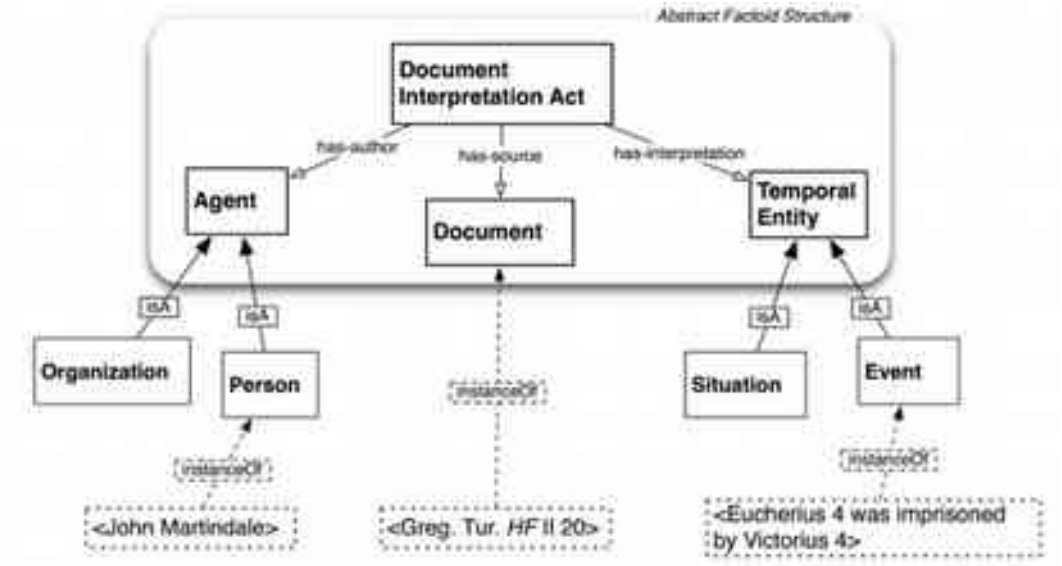

Fig. 9. Explicitation de l'acte d'interprétation des *factoids* (…, p.6)

La nouvelle entité *Document interpretation act* relie ainsi *l'agent* qui produit la connaissance —l'historien— avec *le document* analysé et avec *l'entité temporelle* dont il est question, c'est-à-dire avec l'événement ou la situation dont témoigne la source. L'entité temporelle est alignée avec la

46 *Ibid.*, p. 4.

classe *E2-Temporal-Entity* de CIDOC-CRM qui peut être à son tour considérée comme équivalente à *Perdurant* dans DOLCE comme nous allons le voir. Le modèle *factoid* se propose ainsi d'expliciter le processus de production des connaissances tout en distinguant entre ses différentes parties. Se pose toutefois le problème —déjà soulevé— de la légitimité d'une séparation entre l'essence d'une entité temporelle et le discours qui parle d'elle : comment définir l'événement qui s'est déroulé dans l'histoire, le *perdurant*, en dehors d'un regard de l'acteur qui le décrit ? Le fait même de reconnaître dans un texte un événement n'est-ce pas le résultat d'une construction mentale —culturellement et socialement déterminée— qui s'exprime dans un questionnement ? Comment séparer l'événement lui-même de la connaissance que nous en avons grâce à une analyse critique des témoignages du passé ?

Cette articulation du problème permet de comprendre la *spécificité de l'ontologie SyMoGIH* qui renonce à créer une classe permettant d'identifier les entités temporelles comme telles. Certes, c'est un événement dans l'histoire qui est visé lorsqu'on crée une unité de connaissance de type Naissance mais l'événement historique visé n'est pas identifié comme tel, il n'est pas, concrètement, muni d'un identifiant. En revanche, l'assertion qui le décrit, *le « fait historique » qui instancie la classe Unité de connaissance*, reçoit un URI. Ce choix pourrait paraître discutable lorsqu'il s'agit de modéliser une naissance, événement élémentaire de la vie humaine partagé par toute l'espèce, mais comment définir l'essence de phénomènes bien plus complexes comme la féodalité ou même le simple exercice d'une fonction politique ? Et ce dans l'abstrait, c'est-à-dire en dehors d'une représentation culturellement déterminée ?

Par conséquent, l'ontologie SyMoGIH renonce à cette ambition et se limite à modéliser, grâce aux UC, les *assertions atomisées des historiens qui mettent en relation des objets à un moment plus ou moins précis du temps*. L'instanciation du modèle générique grâce aux types d'UC permet d'expliciter, et donc de *documenter, le regard qu'on a porté sur la réalité historique*, le sens d'une connaissance qu'on a construite à partir des sources et qu'on a appelée, dans notre contexte culturel, une naissance. Ce qui n'empêche pas d'indiquer avec précision l'origine de chaque composante de la connaissance —datations, rôles, sourçages, libellé— en lui associant systématiquement l'identifiant du producteur et du dernier modificateur. Etant donné ce choix ontologique, est-il donc légitime de rendre interopérable une unité de connaissance avec les instances de la classe *perdurant* à laquelle se réfèrent plusieurs autres ontologies ? A notre avis oui car il semble difficile, sauf pour les cas vraiment élémentaires, de séparer l'entité temporelle de sa description, ou de distinguer entre le *Document interpretation act* et la *Temporal entity* faute de pouvoir connaître la deuxième en l'absence du premier. En d'autres termes, il semble difficile d'admettre que les *perdurant*s, de même

que les *factoids*, sont autre chose que des *assertions décrivant un aspect particulier du passé connu grâce aux sources*.

Il nous reste à discuter le rôle que pourrait jouer le *CIDOC Conceptual Reference Model* (CIDOC-CRM ou tout simplement CRM) pour modéliser les connaissances historiques. Parmi les arguments qui plaident en faveur de l'adoption de cette ontologie, créée pour permettre l'interopérabilité des données produites dans le domaine de la conservation des biens culturels, il y a sa maturité —publié pour la première fois sous forme complète en 1999, devenu en 2006 la norme ISO21127, CRM a connu la publication de sa sixième version en janvier 2015[47]— ainsi que sa diffusion. Mais est-il judicieux d'utiliser un modèle qui se veut explicitement voué à l'échange des données issues d'un domaine précis, « curated knowledge of museums »[48], pour modéliser tout type de connaissance historique ? Certes, comme nous l'avons indiqué ci-dessus, les acteurs du domaine des biens culturels, des archives et des bibliothèques sont des producteurs qualifiés d'une quantité importante de connaissances historiques. De plus, ils sont responsables de l'identification et de la description d'une large partie des objets dont l'historien se sert comme sources. Mais est-ce que l'ontologie CIDOC-CRM est suffisamment ouverte pour permettre l'interopérabilité des données historiques en général ?

Lorsqu'on parcourt la liste des classes qui structurent à son plus haut niveau l'ontologie, on retrouve une articulation inspirée de DOLCE : la classe *E77 Persistent Item* comprend les *endurants*, *E2 Temporal Entity* est explicitement assimilée aux *perdurants*, tandis que les classes *E52 Time-Span*, *E53 Place*, *E54 Dimension* et *E92 Spacetime Volume* peuvent être assimilées aux différentes *regions* de DOLCE. L'architecture de CRM reprend donc la structure d'une ontologie de haut-niveau, ce qui est un argument en faveur de l'interopérabilité. De plus, on trouve parmi les sous-classes de *E2 Temporal Entity* des classes telles *E5 Event, E7 Activity* ou *E63 Beginning of Existence* qui correspondent à des aspects essentiels de la connaissance historique. Enfin, la classe *E13 Attribute Assignment* possède à l'origine un sens clairement lié à la conservation des biens culturels mais elle pourrait être utilisée de manière générique pour spécifier le producteur de n'importe quelle assertion, comme le suggèrent les auteurs de l'ontologie *factoid*[49]. Il pourrait remplacer un sourçage des classes assimilables aux *perdurants* qui ne semble pas avoir été prévu.

Ce dernier exemple montre clairement que la plupart des classes du CRM sont construites pour traiter les connaissance liées aux biens culturels, ce qui paraît logique puisqu'il s'agit d'une

47 www.cidoc-crm.org/official_release_cidoc.html.
48 *Definition of the CIDOC Conceptual Reference Model. Version 6.0*, LE BOEUF Patrick, DOERR Martin, ORE Christian Emil, STEAD Stephen (dir.), janvier 2015, p. i.
49 PASIN Michele et BRADLEY John, « Factoid-based prosopography and computer ontologies », *op. cit.*, p. 7.

ontologie de domaine, « a formal ontology intended to facilitate the integration, mediation and interchange of heterogeneous cultural heritage information »[50]. A-t-on donc le droit d'en détourner quelque peu le sens, en rendant ces classes plus génériques, afin d'utiliser cette ontologie pour l'interopérabilité des données historiques ? Les difficultés apparaissent au niveau de la définition des sous-classes de *E77 Persistent Item*, c'est-à-dire dans le domaine des *endurants*. Comme nous l'avons indiqué, dans le contexte du modèle SyMoGIH les sous-classes de *sym:Object* ont été définies en évitant leur multiplication afin de garantir une certaine neutralité ontologique. Dans CRM, les objets sont classés grâce à un vocabulaire très articulé qui rattache une même classe à plusieurs classes supérieures : par ex. *E21 Person,* regroupant les êtres humains, est rattachée à *E39 Actor* et —par l'intermédiaire de trois autres classes— à *E72 Legal Object,* réunissant les objet susceptibles d'être soumis au droit. Dans une approche générique, le premier rattachement, plus neutre et objectif, aurait suffit. Le classement supplémentaire est sans doute utile pour le traitement des biens culturels mais risque d'entraver le traitement d'autres domaines car il ajoute une spécification dont il faudra tenir compte si on veut ajouter des extensions au CRM.

Cette difficulté est renforcée par le mode de traitement des propriétés, articulé en deux principes : chacune des propriétés est reliée à deux classes qui en indiquent respectivement le domaine (*rdfs:domain)* et l'étendue (*rdfs:range*) ; à partir des classes supérieures, les propriétés se propagent par voie d'héritage aux sous-classes. Par conséquent, pour connaître l'ensemble des propriétés d'une classe —par exemple celles de la classe *E21 Person* qui appartient à deux arborescences— il faut remonter à l'ensemble des classes qui constituent la hiérarchie dont elle dépend, ce qui demande une maîtrise approfondie de l'ontologie. Quant à la notion de « participation » —dont nous avons vu l'importance dans les ontologies de haut niveau pour mettre en relation *endurants* et *perdurants*— elle apparaît dans la propriété *P12 occurred in the presence of* dont la classe *E5 Event* est le domaine et la classe *E77 Persistent Item* l'étendue. Son domaine est donc restreint aux seuls événements, même s'il est vrai que ces derniers sont conçus de façon assez large puisque CIDOC-CRM résulte d'une approche *event based* ou *event driven*.

Si on souhaite spécifier le rôle joué par un objet précis on peut utiliser les sous-propriétés de P12, telle *P11 had participant.* Toutefois l'étendue de celle-ci (*rdfs:range*) est restreinte à la classe *E39 Actor* : la participation active à un événement est donc limitée par le modèle aux personnes ou aux groupes humains et ne pourrait comprendre ni les bateaux —pour reprendre l'exemple du SEM — ni un char de combat dans une bataille. Comme CRM est extensible, d'autres propriétés pourraient être ajoutées comme sous-propriétés de P12 afin d'étendre le modèle et de préciser

50 *Definition of the CIDOC Conceptual Reference Model, op. cit.*, Introduction.

d'autres rôles propres à d'autres classes, existantes ou à créer. Mais la cohérence avec le reste de l'ontologie, requise pour la validité de l'extension, pourrait être mise en cause du fait de toutes les autres propriétés virtuellement impliquées par l'héritage.

Plus fondamentalement, CRM a choisi la voie d'une modélisation fondée sur l'articulation en sous-classes qui héritent les propriétés des classes supérieures afin de construire une *architecture qui modélise un domaine précis*, alors que SEM a préféré une *modélisation générique fondée sur la réification de la propriété « participer »* (*sem:hasActor*) qui permet de lui associer tout type de rôle défini dans n'importe quelle ontologie. La première approche présente l'avantage d'être plus explicite et contraignante, ce qui facilite l'interopérabilité entre producteurs de données issus d'un milieu relativement homogène. En revanche, l'approche générique du SEM permet d'intégrer les données issues de contextes différents —car c'est au modèle de chaque producteur qu'il revient de définir les types d'évènements et de rôles— mais le prix de la généricité est un travail d'alignement entre classes et propriétés issues d'ontologies différentes qui peut être complexe.

L'ontologie SyMoGIH se situe à mi-chemin entre ces deux approches : d'une part, elle partage avec SEM un modèle générique de haut niveau, ouvert et facilement extensible ; d'autre part, la *construction progressive des instances du modèle* grâce aux types d'unités de connaissance et aux types de rôles établit, à partir de la pratique, une modélisation explicite de différents types de connaissances historiques —documentée sur le site du projet[51]— qui peut servir comme référence.

La question de l'interopérabilité

Nous disposons maintenant des outils conceptuels nécessaires pour traiter la question de l'interopérabilité des données historiques. Il faut tout d'abord s'interroger sur la finalité de celle-ci. Je présenterai trois cas de figure en lien avec le projet SyMoGIH. Comme le but du projet est de mutualiser et de valoriser les données produites par les chercheurs en histoire, il s'agira de mettre les données à disposition du public ; d'en favoriser l'intégration avec les données produites par les conservateurs des biens culturels ; de mettre ces données en réseau avec celles produites par d'autres chercheurs afin d'élargir le volume des données disponibles, de tester de nouvelles hypothèses et de produire de nouvelles connaissances. La suite de l'exposé sera articulée autour de ces cas de figure, tout en prenant comme exemple une connaissance élémentaire : la naissance d'un acteur.

Plusieurs difficultés apparaissent lorsqu'on soulève la question de l'interopérabilité des données à l'aide des technologies du web sémantique : celle de l'*alignement* des instances, des classes et des propriétés entre les différentes ontologies ; celle de la gestion de la *redondance* des données, c'est-à-

51 http://symogih.org.

dire de la multiplication des connaissances portant sur le même objet provoquée par l'intégration de plusieurs entrepôts ; celle de la mesure de la *fiabilité* et des critères permettant de choisir parmi les données redondantes celles qui présentent une meilleure qualité. Quelques solutions seront fournies au fil de l'exposé en rapport avec les trois axes retenus. Les exemples seront présentés en utilisant la syntaxe *Turtle* (*Terse RDF Triple Language*)[52] qui permet d'écrire les triplets RDF de manière relativement lisible et qui est également utilisée, avec quelques particularités, par le langage d'interrogation de triplets SPARQL[53].

L'unité de connaissance « Johannes Kepler naît à Weil der Stadt le 27 décembre 1571 », transformée en données grâce à l'ontologie SyMoGIH et exprimée en *Turtle*, se présente sous cette forme :

```
PREFIX  sym:  <http://symogih.org/ontology/>
PREFIX  syr:  <http://symogih.org/resource/>
PREFIX  viaf:  <http://viaf.org/viaf/>
PREFIX  xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX  rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

syr:Info29659       rdf:type  sym:Information ;
                    sym:hasKnowledgeUnitType  syr:TyIn14 .  # naissance
                    sym:hasCreator  viaf:14907585 .  # Francesco Beretta
                    sym:hasCreationTimestamp  "2010-05-26T14:55:33"^^xsd:dateTime .
                    sym:hasLastModifier  viaf:14907585 .
                    sym:hasLastModificationTimestamp  "2013-09-06T13:09:25"^^xsd:dateTime .

_:d1                rdf:type  sym:Dating ;
                    sym:dates  syr:Info29659 ;
                    sym:dateTime  "1571-12-27"^^xsd:date ;
                    sym:datingType  syr:AbOb246 ;  # date unique
                    sym:datingCertainty  3 .  # certaine
                    sym:hasCreator [...]

_:s1                rdf:type  sym:Sourcing ;
                    sym:sources  syr:Info29659 ;
                    sym:associatesSourcingEntity  syr:Bibl7144 ;  # Dictionary of scientific biography
                    sym:exactReference  "t.7, p.289" ;
                    sym:sourcingType  3 ;  # littéral
                    sym:sourcingReliabilityDegree  3 .  # certain
                    sym:hasCreator [...]

syr:InRo86726       rdf:type  sym:Role ;
                    sym:isComponentOf  syr:Info29659 ;
                    sym:associatesObject  syr:Actr195 ;  # Kepler, Johannes
                    sym:hasRoleType  syr:TyRo40 ;  # naître
                    sym:associatedObjectIdentificationCertainty  3 ;  # certaine
                    sym:hasCreator [...]

syr:InRo86726       rdf:type  sym:Role ;
                    sym:isComponentOf  syr:Info29659 ;
                    sym:associatesObject  syr:NaPl90073 ;  # Weil der Stadt
                    sym:hasRoleType  syr:TyRo8 ;  # localiser
                    sym:associatedObjectIdentificationCertainty  3 ;  # certaine
                    sym:hasCreator […]
```

52 http://www.w3.org/TR/turtle/ .
53 http://www.w3.org/TR/sparql11-query/ .

Fig. 10. La naissance de Kepler exprimée dans l'ontologie SyMoGIH

À la suite de la liste des espaces de noms utilisés, avec leurs abréviations usuelles, nous trouvons l'instance qui représente l'unité de connaissance elle-même, appartenant à la classe *sym:Information,* suivie de l'indication de la datation, du sourçage et des deux rôles qui associent respectivement la personne qui naît et le lieu de naissance. Chaque instance du modèle générique est assortie des propriétés qui en détaillent les caractéristiques y compris, en italique, l'indication du créateur et du dernier modificateur, avec l'horodatage respectif. Afin de faciliter la lecture, ces quatre propriétés, communes à toutes les classes, ne sont développées en entier qu'une seule fois.

L'exemple de la Figure 10 illustre concrètement la granularité fine de la structuration des connaissances historiques que permet l'ontologie SyMoGIH et, en même temps, sa lisibilité : en raison de la généricité du modèle, chaque unité de connaissance présente la même structure avec une articulation de ses différentes composantes qui permet d'en détailler les propriétés. Le sens de chaque assertion atomisée est indiqué grâce à une instance des Types d'UC ainsi qu'aux Types de rôle qui précisent le sens de l'association de chaque objet. On relèvera encore que la datation possède, dans cet exemple, le degré de certitude le plus élevé et qu'il en est ainsi également de l'identification des objets associés par les rôles (*sym:associatedObjectIdentificationCertainty*). Comme la source de la connaissance est un dictionnaire biographique de qualité, l'auteur de l'unité de connaissance a considéré que ce sourçage, repris à la lettre, est très fiable et il n'a émis aucun doute concernant les différents paramètres de certitude. Enfin les propriétés relatives à l'identité du créateur et l'horodatage montrent que ces données ne reproduisent pas directement un événement de l'histoire mais une assertion d'un historien concernant cet événement.

Concernant le cas de figure d'une interopérabilité qui vise la mise à disposition des données pour le public prenons en considération la même unité de connaissance telle qu'elle se trouve dans le projet DBPedia, un silos de donnée qui résulte de l'extraction sous forme de graphe d'une partie des connaissances de *Wikipedia*[54]. Les données concernant la naissance de Kepler sont présentes dans DBPedia sous cette forme :

```
PREFIX   dbpedia: <http://dbpedia.org/resource/>
PREFIX   dbpedia-owl: <http://live.dbpedia.org/ontology/>
PREFIX   xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX   syr:  <http://symogih.org/resource/>
PREFIX   owl: <http://www.w3.org/2002/07/owl#>
PREFIX   rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

dbpedia:Johannes_Kepler      dbpedia-owl:birthDate  "1571-12-27"^^xsd:date ;
                             dbpedia-owl:birthPlace   dbpedia:Weil_der_Stadt .
```

54 http://wiki.dbpedia.org .

```
# alignement avec les objets de l'ontologie SyMoGIH

dbpedia:Johannes_Kepler   owl:sameAs   syr:Actr195 .
dbpedia:Weil_der_Stadt   owl:sameAs   syr:NaPl90073 .
```

Fig. 11. La naissance de Kepler exprimée dans l'ontologie DBPedia

A la différence des ontologies présentées jusqu'ici, DBPedia ne construit pas une classe d'unités de connaissance ou d'événements (*perdurants*) mais se limite à exprimer les propriétés des objets sous forme de triplets. Une première limite de cette approche réside dans le fait qu'aucune indication concernant la source ou la fiabilité de la connaissance n'est associée à ces propriétés. Une deuxième limite est représentée par l'absence de lien entre deux propriétés portant sur une même connaissance : seul l'humain comprend, en lisant les données, que ces deux propriétés concernent le même événement, la naissance de Kepler.

Comme le montre la deuxième partie de l'exemple (Fig. 11), l'alignement entre instances du modèle relevant du domaine des objets (*endurants*), réalisé grâce à la propriété *owl:sameAs,* ne pose pas de problème particulier. La seule précaution à prendre est de ne pas confondre l'identifiant de l'objet, généralement sous forme d'URI[55], avec l'URL (*Uniform Resource Locator*) qui permet d'accéder à la notice qui déréférence l'identifiant[56], c'est-à-dire qui fournit les éléments d'information permettant d'identifier l'objet auquel se réfère l'URI. En revanche, la différence essentielle qui subsiste entre les ontologies DBPedia et SyMoGIH ne permet pas un alignement direct entre classes et propriétés au niveau des unités de connaissance (*perdurants*). Cette différence ne représente toutefois pas une entrave à l'interopérabilité car on peut exprimer dans le vocabulaire de l'ontologie DBPedia les données structurées au départ selon le modèle SyMoGIH grâce à la structure CONSTRUCT propre au langage de requêtes SPARQL :

```
PREFIX   dbpedia: <http://dbpedia.org/resource/>
PREFIX   dbpedia-owl: <http://live.dbpedia.org/ontology/>
PREFIX   xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX   syr:  <http://symogih.org/resource/>
PREFIX   sym:  <http://symogih.org/ontology/>
PREFIX   owl: <http://www.w3.org/2002/07/owl#>
PREFIX   rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

CONSTRUCT
   {
    ?person          dbpedia-owl:birthDate   ?date ;
                     dbpedia-owl:birthPlace   ?place .
   }
```

55 Dans notre exemple : http://dbpedia.org/**resource**/Johannes_Kepler .

56 Dans notre exemple : http://live.dbpedia.org/**page**/Johannes_Kepler (consulté le 2 février 2015). Au sujet du déréférencement, voir HEATH Tom et BIZER Christian, « Linked Data: Evolving the Web into a Global Data Space (1st edition) », in *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(2011)1, 1-136. Morgan & Claypool, chapitre 2.3 Making URIs Defererenceable. Accessible ici : http://linkeddatabook.com/editions/1.0/ .

```
WHERE
   {
    ?info      rdf:type   sym:Information ;
               sym:hasKnowledgeUnitType   syr:TyIn14.

    ?dating    rdf:type   sym:Dating ;
               sym:dates   ?info ;
               sym:dateTime   ?date ;
               sym:datingType   syr:AbOb246 ;   # date unique

    ?role_1    rdf:type    sym:Role ;
               sym:isComponentOf   ?info ;
               sym:hasRoleType   syr:TyRo8 ;   #  localiser
               sym:associatesObject   ?place ;

    ?role_1    rdf:type    sym:Role ;
               sym:isComponentOf   ?info ;
               sym:hasRoleType   syr:TyRo40 ;   #  naître
               sym:associatesObject   ?person ;
  }
```

Fig. 12. Conversion de l'ontologie SyMoGIH dans le vocabulaire de l'ontologie *DBPedia*

Cette requête renvoie toutes les unités de connaissance de type Naissance présentes dans l'entrepôt, puis récupère pour chacune les variables correspondantes à la personne qui naît, au lieu de naissance et à la date de celle-ci (en gras dans la Fig. 12) et réécrit les données en utilisant les propriétés de l'ontologie DBPedia. De cette opération d'alignement entre ontologies résulte inévitablement une simplification et un appauvrissement de la connaissance car on perd toutes les indications concernant sa production et sa fiabilité. Mais on met ainsi à la disposition du public des informations de qualité produites par la recherche historique. Il est d'ailleurs préférable d'utiliser à cette fin des ontologies telles LODE[57] ou BIO[58], voire le SEM même, plutôt que celle de DBPedia car, tout en gardant une structure très simple, elles utilisent des classes qui correspondent à la notion de *perdurant* et elles permettent donc plus facilement de regrouper tous les objets qui participent à une unité de connaissance.

En sens inverse, peut-on tirer profit pour la recherche historique de données disponibles dans DBPedia ou dans d'autres entrepôts possédant une structure ontologique comparable à celle-ci ? Grâce à une requête SPARQL appropriée on peut produire une unité de connaissance utilisant le vocabulaire de l'ontologie SyMoGIH :

```
PREFIX   dbpedia: <http://dbpedia.org/resource/>
PREFIX   xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX   sym:  <http://symogih.org/ontology/>
PREFIX   syr:  <http://symogih.org/resource/>
PREFIX   rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>


_:info_1       rdf:type   sym:Information ;
```

57 http://linkedevents.org/ontology/ .
58 BIO: A vocabulary for biographical information, http://vocab.org/bio/0.1/.html .

```
                sym:hasKnowledgeUnitType   syr:TyIn14 .  # naissance
                sym:hasCreator   viaf:14907585 .  # Francesco Beretta
                sym:hasCreationTimestamp   "2015-02-02T14:55:33"^^xsd:dateTime .

_:d1            rdf:type    sym:Dating ;
                sym:dates    _:info_1 ;
                sym:dateTime   "1571-12-27"^^xsd:date ;
                sym:datingType   syr:AbOb246 ;   # date unique
                sym:datingCertainty  3 .   # certaine
                sym:hasCreator [...]

_:s1            rdf:type    sym:Sourcing ;
                sym:sources   _:info_1 ;
                sym:associatesSourcingEntity    <http://dbpedia.org/page/Johannes_Kepler> ;
                sym:sourcingType  3 ;    #  littéral
                sym:sourcingReliabilityDegree  1 .   # incertain
                sym:hasCreator [...]

_:r1            rdf:type    sym:Role ;
                sym:isComponentOf   _:info_1 ;
                sym:associatesObject  dbpedia:Johannes_Kepler ;
                sym:hasRoleType   syr:TyRo40 ;    #  naître
                sym:associatedObjectIdentificationCertainty   3 ;  # certaine
                sym:hasCreator [...]

_:r2            rdf:type    sym:Role ;
                sym:isComponentOf   _:info_1 ;
                sym:associatesObject   dbpedia:Weil_der_Stadt ;
                sym:hasRoleType   syr:TyRo8 ;    #  localiser
                sym:associatedObjectIdentificationCertainty   3 ;  # certaine
                sym:hasCreator [...]
```

Fig. 13. La naissance de Kepler exprimée dans le vocabulaire SyMoGIH à partir des données DBPedia

Les deux triplets de DBPedia (Fig. 11) permettent de créer une nouvelle unité de connaissance qui a la même structure que celle des données produites directement selon l'ontologie SyMoGIH (Fig. 10) mais qui présente toutefois quelques différences importantes. Tout d'abord, le sourçage se fonde sur la page qui déréférence l'URI de Kepler dans DBPedia : étant donné que sur cette page il n'y a aucune indication concernant la source ou l'auteur des données, le degré de fiabilité du sourçage —souligné en gras dans l'exemple— a été défini au niveau le plus bas. Par conséquent, si cette unité de connaissance devait se trouver en concurrence avec une autre instance de la classe UC portant sur la même naissance, mais possédant une meilleure fiabilité du sourçage, la deuxième serait préférée à la première. Ensuite, l'unité de connaissance et toutes ses parties sont identifiées par des ressources anonymes (*blank nodes*) —sous la forme _:info_1— puisque dans l'ontologie d'origine il n'y a pas de *perdurant* correspondant qui disposerait d'un identifiant sous forme d'URI. Par contre, si l'ontologie de départ possédait une structure comportant une classe correspondante aux *perdurants,* munie d'identifiants sous forme d'URI, on pourrait aligner les instances de celle-ci avec les unités de connaissance identiques du projet SyMoGIH. Pour ce faire, on peut utiliser la propriété *owl:sameAs* s'il y a identité entre les deux assertions ainsi que leurs composantes, ou avec d'autres propriétés exprimant une équivalence moins stricte, par ex. la propriété *skos:closeMatch* du

Simple Knowledge Organization System (SKOS)[59].

Enfin, les objets associés sont identifiés en utilisant les URI du projet DBPedia afin de souligner le fait que l'ontologie SyMoGIH peut être utilisée pour permettre l'interopérabilité entre différentes sources de données tout comme le modèle SEM présenté ci-dessus mais avec une différence : l'alignement entre les classes et les propriétés issues de ces ontologies s'effectuera vers les instances des classes *sym:KnowledgeUnitType* et *sym:RoleType* déjà présentes dans le projet SyMoGIH et créées, ou améliorées, grâce à l'intervention des historiens qui participent au projet, c'est-à-dire aux spécialistes du domaine. Le modèle résultant ne sera donc pas à refaire chaque fois mais il sera progressivement instancié, complété et documenté sur le site *symogih.org*, tout en conservant une architecture de base générique et lisible.

Concernant le cas de figure d'une interopérabilité avec les données produites par les conservateurs des biens culturels, nous prendrons en considération l'exemple de l'ontologie CIDOC-CRM, développée précisément pour répondre aux besoins de ce domaine comme nous l'avons vu. L'exportation de l'unité de connaissance de la Figure 10 vers cette ontologie donnera le résultat suivant :

```
PREFIX   crm: <http://purl.org/NET/crm-owl#>
PREFIX   syr: <http://symogih.org/resource/>
PREFIX   sym: <http://symogih.org/ontology/>
PREFIX   owl: <http://www.w3.org/2002/07/owl#>
PREFIX   rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

syr:Info29659          rdf:type   crm:E67_Birth ;
                       crm:P98_brought_into_life  syr:Actr195 ;  #  Kepler, Johannes
                       crm:P7_took_place_at   syr:NaPl90073 ;   #  Weil der Stadt
                       crm:P4_has_time-span  _:ts1.
_:ts1          rdf:type                          crm:E52_Time-Span ;
               crm:P82a_begin_of_the_begin       "1571-12-27"^^xsd:date ;
               crm:P82b_end_of_the_end           "1571-12-27"^^xsd:date . .


# expression de l'équivalence des classes et des propriétés

crm:E67_Birth                 owl:equivalentClass       syr:TyIn14 .  # Naissance
crm:P98i_was_born             owl:equivalentProperty    syr:TyRo40 .   # naître
crm:P7i_took_place_at         owl:equivalentProperty    syr:TyRo8 .   # localiser
```

Fig. 14. La naissance de Kepler exprimée dans l'ontologie CIDOC-CRM

Tout comme dans le cas de la transcription dans le vocabulaire de l'ontologie DBPedia (Fig. 12), une requête SPARQL permet d'effectuer la réécriture de données SyMoGIH dans le modèle CRM au prix de la perte des propriétés qui spécifient le sourçage et la certitude de la connaissance. Mais, à la différence de celle de DBPedia, l'ontologie CRM comporte une structure centrée autour de classes de *perdurants* : il est donc possible —grâce à la propriété *owl:equivalentClass*—

59 www.w3.org/TR/skos-reference/#mapping .

d'aligner les classes existantes dans les deux ontologies et qui possèdent la même extension, c'est-à-dire qui regroupent le même ensemble d'individus même si leurs propriétés ne sont pas identiques[60]. Le même principe s'applique aux propriétés 'naître' et 'localiser' qui peuvent être alignées —grâce à la propriété *owl:equivalentProperty*— aux propriétés correspondantes du CRM (Fig. 14, partie inférieure). A noter que, afin de respecter la direction des propriétés (c'est-à-dire les classes qui constituent leur domaine et leur étendue respectives), l'équivalence se fait par rapport au sens inversé des propriétés du CRM (*inverse of*). Rappelons à ce sujet que dans l'ontologie SyMoGIH les rôles résultent d'une réification de la propriété générique 'participer' : 'naître' et 'localiser' sont donc virtuellement deux sous-propriétés de celle-ci.

La différence essentielle entre les ontologies SyMoGIH et CIDOC-CRM réside précisément dans la *réification* de la propriété 'participer' opérée en créant la classe *sym:Role*, de même que celle des autres composantes de l'unité de connaissance (*sym:Datation et sym:Sourcing*), qui permet d'associer à ces classes toutes les propriétés nécessaires pour prendre en compte une structuration fine de la connaissance propre à la méthode historique. Cette différence essentielle ne représente pas un obstacle à l'interopérabilité comme le montre la transcription dans le vocabulaire de l'ontologie SyMoGIH d'une imaginaire assertion en CIDOC-CRM concernant la naissance de Kepler :

```
PREFIX  dbpedia: <http://dbpedia.org/resource/>
PREFIX  xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX  sym:  <http://symogih.org/ontology/>
PREFIX  syr:  <http://symogih.org/resource/>
PREFIX  crm: <http://purl.org/NET/crm-owl#>
PREFIX  owl: <http://www.w3.org/2002/07/owl#>
PREFIX  rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

_:info_1      rdf:type  sym:Information ;
              sym:hasKnowledgeUnitType  crm:E67_Birth . # classe équivalente à syr:TyIn14 (naissance)
              sym:hasCreator  viaf:14907585 .  # Francesco Beretta
              sym:hasCreationTimestamp  "2015-02-02T14:55:33"^^xsd:dateTime .

_:d1          rdf:type  sym:Dating ;
              sym:dates  _:info_1 ;
              sym:dateTime  "1571-12-28"^^xsd:date ;  # date erronée
              sym:datingType  syr:AbOb246 ;  # date unique
              sym:datingCertainty  3 .  # certaine
              sym:hasCreator [...]

_:s1          rdf:type  sym:Sourcing ;
              sym:sources  _:info_1 ;
              sym:associatesSourcingEntity  [URI de la source de l'unité de connaissance ] ;
              sym:sourcingType  3 ;  # littéral
              sym:sourcingReliabilityDegree  2 .  # probable
              sym:hasCreator [...]

_:r1          rdf:type  sym:Role ;
              sym:isComponentOf  _:info_1 ;
              sym:associatesObject  dbpedia:Johannes_Kepler ;
```

60 http://www.w3.org/TR/2004/REC-owl-ref-20040210/#equivalentClass-def .

```
                sym:hasRoleType          crm:P98_brought_into_life ; # propriété réifiée équivalente à
syr:TyRo40 (naître)
                sym:associatedObjectIdentificationCertainty   3 ; # certaine
                sym:hasCreator [...]

_:r2            rdf:type    sym:Role ;
                sym:isComponentOf   _:info_1 ;
                sym:associatesObject   dbpedia:Weil_der_Stadt ;
                sym:hasRoleType          crm:P7_took_place_at ;      # propriété réifiée équivalente à
syr:TyRo8 (localiser)
                sym:associatedObjectIdentificationCertainty   3 ; # certaine
                sym:hasCreator […]
```

Fig. 15. Le modèle SyMoGIH comme ontologie au service de l'interopérabilité

Dans ce cas de figure fictif, qui pousse à l'extrême les possibilités d'interopérabilité offerte par l'ontologie SyMoGIH —de manière analogue au SEM—, les URI des objets associés sont définis en utilisant les instances de DBPedia, souvent utilisée comme étape intermédiaire pour les alignements d'identifiants d'objets. En revanche, les types d'unité de connaissance et les types de rôle sont issus du vocabulaire de CIDOC-CRM. Cet exemple indique de quelle manière on pourra transcrire dans le vocabulaire de l'ontologie SyMoGIH les données relevant d'un type d'assertions qui n'existe pas encore parmi les instances du modèle générique déjà construites par les historiens qui participent au projet. Quant à la date, j'ai volontairement introduit une erreur qui me permettra d'illustrer le mécanisme d'interopérabilité. Enfin, la fiabilité du sourçage (*sym:sourcingReliabilityDegree*) possède la valeur 'probable' afin d'exemplifier le cas de figure d'une connaissance issue d'une institution patrimoniale, par exemple le site web d'un musée —ce qui garantit virtuellement la qualité de l'information—, mais sans mention de la source dans les données (cf. l'exemple de la Fig. 14), ce qui diminue la degré de fiabilité de l'assertion.

La notion de réification des composantes de la connaissance propre à l'ontologie SyMoGIH comporte donc, d'une part, une possibilité beaucoup plus fine de traiter le sourçage de chaque partie d'une donnée historique et, d'autre part, une plus grande flexibilité dans l'extension de l'ontologie et l'alignement avec d'autres modèles. Nous avons indiqué ci-dessus les raisons de cette différence en comparant les ontologies SEM et CIDOC-CRM. A partir de ces considérations, et des exemples de données produites selon ces différents vocabulaires, il résulte que le modèle SyMoGIH apparaît comme l'ontologie la plus adaptée à une interopérabilité des données historiques dans le contexte de la recherche.

Dans ce troisième cas de figure de l'interopérabilité, en accord avec les principes de la méthode historique, l'objectif est de disposer d'un volume suffisant de données de qualité pour pouvoir appliquer un questionnement aux connaissances récoltées et pour tester de nouvelles hypothèses d'explication. Il s'agit donc d'intégrer les données issues de multiples entrepôts et d'éliminer la

redondance des connaissances tout en retenant celles qui sont de meilleure qualité. C'est à ce stade que vont devenir opératoires plusieurs concepts présentés précédemment. Une première étape du processus consiste à retranscrire toutes les données importées dans le vocabulaire de l'ontologie SyMoGIH grâce à la structure CONSTRUCT du langage SPARQL. Suite à cette opération, une requête SPARQL appropriée nous permettra de savoir que nous disposons de trois unités de connaissance, représentées respectivement sur les figures 10, 13 et 15, qui portent sur le même fait historique, la naissance de Kepler.

La comparaison entre ces trois assertions conduit à un premier constat : elles sont toutes qualifiées en tant qu'instances de la classe *sym:Information*. Elle relèvent donc toutes du statut épistémologique d'assertions formulées par l'historien *après* analyse critique et comparaison entre les sources. En effet, une assertion issue d'une notice de Wikipedia, dont sont tirées les données de DBPedia (Fig. 11), est équivalente —au point de vue du statut épistémologique— à une notice de dictionnaire biographique. Une connaissance produite par les conservateurs des biens culturels (Fig. 14) a la même visée : reproduire autant que possible la réalité historique comme telle. Pour mémoire, les instances de la classe *sym:Content* ont en revanche pour fonction d'exprimer une connaissance telle que l'historien la reconnaît dans un seul document. Ces assertions sont construites afin de reproduire la teneur précise de la source même si on sait par ailleurs qu'elle se trompe. Relevons à ce sujet que les données structurées produites par l'ingénierie des connaissances par extraction automatique de textes relèvent du même niveau épistémologique des assertions de type *sym:Content*. Pour être utilisées pour la recherche en histoire, elles demandent à être soumise à une analyse critique.

Si on disposait de deux assertions, l'une de type *sym:Information,* l'autre de type *sym:Content*, portant sur la même unité de connaissance, et que l'objectif visé était de collecter un ensemble de faits historiques, la première assertion serait à préférer à la deuxième à cause de son statut épistémologique : elle vise en effet à être la meilleure description possible —en l'état des connaissances— de la réalité historique alors que la deuxième reproduit le contenu d'un document précis. Dans notre exemple, nous sommes donc en présence de trois assertions qui appartiennent à la classe *sym:Information* et qui portent sur la même unité de connaissance. Lorsqu'on les compare en détail, on constate qu'elles associent les mêmes objets et que ceux-ci jouent des rôles équivalents. On constate en revanche une divergence dans la datation de l'une des unités de connaissance. Si on ne disposait pas de la classe *sym:Sourcing*, avec ses propriétés, on ne saurait pas laquelle des datations est à préférer. Dans notre cas, l'une des trois assertions possède le meilleur degré de fiabilité de son sourçage et elle a été extraite littéralement de la source : il s'agit donc de celle dont il

faut retenir la datation. Plus généralement, la qualité du sourçage de cette unité de connaissance amène à la retenir, avec toutes ses composantes qui héritent du même sourçage, et à écarter les deux autres.

Les données de l'exemple retenu peuvent être traitées de manière assez simple. Mais on peut imaginer des cas de figure beaucoup plus complexes, comportant des sourçages multiples pour une même unité de connaissance qui posséderaient différentes valeurs de fiabilité, voire même des sourçages contradictoires ; ces sourçages multiples pourraient porter uniquement sur l'une des composantes de la connaissance, par ex. la datation ; l'identification de la période temporelle, au niveau de la datation, ou celle des objets associés grâce aux rôles pourrait être qualifiée de seulement 'probable' voir 'incertaine'. Pour éliminer la redondance des informations il faudrait alors mettre en place un algorithme, sous forme d'une ou plusieurs requêtes SPARQL ou en utilisant des moteurs de raisonnement, prenant en compte ces différents éléments.

L'algorithme aura cette structure : préférer les instances appartenant à la classe *sym:Information* à celles de type *sym:Content* si les deux sont présentes ; mesurer la valeur des sourçages en combinant leur type (littéral, par déduction, contradictoire) et leur degré de fiabilité grâce à un calcul arithmétique qui utilise les valeurs du codage ; à parité de qualité du sourçage, les autres paramètres de certitude permettent de choisir entre les datations disponibles ou les objets associés avec le même type de rôle ; en cas d'égalité de toutes ces propriétés on choisira l'unité de connaissance la plus récente en termes de création ou, en cas de parité, de modification, ou alors, en tout dernier recours, et à parité de tous les autres propriétés, la première des occurrences d'une assertion portant sur la même unité de connaissance. On pourrait également privilégier un producteur de connaissances, individuel ou collectif, par rapport aux autres.

Avec cette méthode, on atteindra virtuellement l'objectif de l'élimination de la redondance en produisant un ensemble de données dans lequel il n'y en aura qu'une et une seule exprimant le même fait historique. Comme on n'aura conservé que les données qui répondent le mieux aux critères de l’algorithme retenu, elles possèderont la meilleure fiabilité parmi celles mises à disposition grâce à la fusion des entrepôts. Cette méthode, fondée sur la retranscription des données dans le vocabulaire de l'ontologie SyMoGIH pourra être appliquée automatiquement à un volume important de données et permettra, grâce à l'utilisation de logiciels de visualisation et d'analyse, de produire de nouvelles connaissances. Celles-ci pourront être transformées à leur tour en données grâce aux instances —existantes ou nouvellement créées— du modèle générique SyMoGIH.

Conclusion

Dans ce chapitre j'ai essayé d'apporter une réponse, à partir de l'expérience du projet Système modulaire de gestion de l'information historique (SyMoGIH), à la question de l'interopérabilité entre données issues de différents entrepôts, susceptibles d'être utilisées par les chercheurs en histoire afin de produire de nouvelles connaissances. Il a d'abord fallu clarifier et délimiter le concept de donnée historique car cette notion se situe au croisement entre deux disciplines : la réflexion sur la méthode historique et l'ingénierie des connaissances. Après avoir parcouru les étapes essentielles de la recherche en histoire il a été possible de proposer une définition du « fait historique » conçu en tant qu'assertion atomisée construite à partir d'un questionnement qui décrit un aspect du passé grâce aux témoignages conservés, analysés à l'aide de la méthode critique.

L'application de l'ontologie du projet SyMoGIH aux connaissances historiques —les « faits »— les transforme en données structurées dont le sens est explicité et documenté par les instances du modèle génériques accessibles aux chercheurs et au public sur le site web du projet (*symogih.org*). La présentation détaillée de cette ontologie, exprimée dans ce chapitre à l'aide du formalisme RDFS, permet de soulever toute une série de questions qui se posent lorsqu'on veut transformer les connaissances historiques en données en prenant en compte les exigences qu'impose l'application fine de la méthode critique en termes d'explicitation de la construction des connaissances, de traçabilité, de datation, de gestion de l'incertitude et de la contradiction entre les sources. A l'aide de l'ontologie DOLCE, et en particulier autour de la notion de réification de la relation de « participation » qui associe les objets (*endurants*) avec les connaissances (*perdurants*), j'ai proposé une comparaison entre la structure ontologique du modèle SyMoGIH et celles du *Simple event model* (SEM) et de CIDOC-CRM, en posant en même temps la question de l'adéquation de ces modèles avec la finalité de l'interopérabilité entre donnes historiques issues de différents producteurs.

La présentation de quelques exemples concernant trois cas de figure différents —la mise à disposition du public des données de la recherche, l'interopérabilité avec celles produites par les conservateurs des biens culturels et la mutualisation de données entre projets de recherche en histoire— montre, d'une part, que l'utilisation du modèle SyMoGIH pour produire des données structurées permet aisément de les retranscrire dans les vocabulaires des autres ontologies. D'autre part, et en sens inverse, l'ontologie SyMoGIH présente une granularité fine dans le traitement des composantes de la connaissance historique (datation, sourçage, qualification de la participation des objets impliqués) qui permet, une fois qu'on a reformulées dans le vocabulaire de ce modèle les données issues d'autres ontologies, de traiter le problème de la redondance des données : grâce à un

algorithme approprié on pourra ainsi hiérarchiser les connaissances en fonction d'un indicateur de fiabilité et de certitude et ne retenir que celles présentant le score le plus élevé. L'historien disposera ainsi d'une procédure lui permettant d'accroître le volume de données de qualité à sa disposition en vue de produire de nouvelles connaissances en leur appliquant les méthodes propres à sa discipline.