Mapping Ontology with Probabilistic Relational Models
An Application to Transformation Processes

Cristina Manfredotti¹, Cedric Baudrit², Juliette Dibie-Barthélemy¹ and Pierre-Henri Wuillemin³

¹INRA AgroParisTech, 16, rue Claude Bernard, 75231 Paris Cedex 5, France
²Institut National de la Recherche Agronomique, Institut de Mécanique et d’Ingénierie, Talence, France
³Sorbonne Universités, UPMC, Univ Paris 06, CNRS UMR 7606, LIP6, Paris, France

Keywords: Ontology, Probabilistic Graphical Models, Probabilistic Relational Models.

Abstract: Motivated by the necessity of reasoning about transformation experiments and their results, we propose a mapping between an ontology representing transformation processes and probabilistic relational models. These extend Bayesian networks with the notion of class and relation of relational data bases and, for this reason, are well suited to represent concepts and ontologies’ properties. To easy the representation, we exemplify a transformation process as a cooking recipe and present our approach for an ontology in the cooking domain that extends the Suggested Upper level Merged Ontology (SUMO).

1 INTRODUCTION

A transformation process is a dynamic process composed of a sequence of operations which allows inputs to be transformed in several different outputs. It relies on data and knowledge coming from heterogeneous sources, often suffers from lack of information and contains uncertain data, the observations being acquired with seldom precise instruments, different from a process to another. Reasoning on a transformation process supposes to be able, for instance, to predict future outputs given certain inputs or given that some inputs are missing, to diagnose how to obtain the best output by determining the important inputs, to control the process and to suggest the best sequence of operations. In this paper, we provide a step forward toward reasoning on transformation processes. To do that, we have to face two main locks: (1) data and knowledge heterogeneity and (2) uncertainty quantification.

In order to face the first lock, a relevant solution is to use ontologies (Fridman Noy, 2004; Doan et al., 2012). Many works propose solutions to manage uncertainty in ontologies such as adapting the querying process using fuzzy sets (Buche et al., 2005), reasoning using a possibilistic and probabilistic description logic reasoner (Qi et al., 2010; Lukasiewicz and Straccia, 2008), reasoning in fuzzy ontology (Bobillo et al., 2013) or using existing knowledge to predict unfilled information (Saïs and Thomopoulos, 2014).

In this paper, we propose to quantify uncertainty in reasoning with probability theory.

We propose to explore a novel way to reason on transformation processes facing the two locks introduced above: we combine the representation expression of ontologies with the reasoning possibilities of probabilistic relational models which provides a consistent framework to process uncertainty. Probabilistic relational models add the notion of class to Bayesian networks which allows to do filtering, prediction, classification and smoothing. The notion of ‘class’, common to ontologies (concepts) and probabilistic relational models, leads us to choose this probabilistic model to be paired with the ontology’s representation model. The first step of this combination consists in proposing a mapping between a transformation process ontology and a probabilistic relational model. The next step, not presented in this paper, will be to learn the parameters of the model from an ontological database and then to implement methods able to reason on the learned model.

We present all our findings in the domain of cooking recipes because it well exemplifies a general transformation process, being simple and easy to understand. We first present background on probabilistic relational models. We detail, in Section 3, an ontology of transformation processes and, in Sections 4, its mapping with a probabilistic relational model. We discuss our findings in Section 5 providing a comparison with the state of the art.
2 PRMs

A Bayesian network (BN) (Koller and Friedman, 2009) is the representation of a joint probability over a set of random variables that uses a Directed Acyclic Graph (DAG) to encode probabilistic relations between variables (Figure 1).

![Figure 1: Two Bayesian networks.](image)

Probabilistic Relational Models (PRMs) extend the BN representation with a relational structure between (potentially repeated) fragments of BN called classes (Torti et al., 2010). A class is defined as a DAG over a set of inner attributes and a set of outer attributes from other classes referenced by so-called reference slots (Figure 2).

![Figure 2: A relationa schema formed by two classes X and Y. ρ is a reference slot in Y which indicates that attributes of class Y (D, E, F) can have parents in class X (A, B, C).](image)

The probabilistic models are defined at class level over the set of inner attributes, conditionally to the set of outer attributes and represent generic probabilistic relations inside the classes that will be instantiated for each specific situation. In this way, PRMs provide a high-level, qualitative description of the structure of the domain and the quantitative information provided by the probability distribution (Friedman et al., 1999).

In a PRM, the (relational) schema describes a set of classes C, associated with attributes \( A(C) \) and reference slots \( R(C) \). A slot chain is defined as a sequence of reference slots that allows to put in relation attributes of objects that are indirectly related. A system in the PRM provides a probability distribution over a set of instances of a relational schema (Wuillemin and Torti, 2012). In this paper we present a general approach to deduce relational schemas from a given ontology of transformation processes.

3 TRANSFORMATION PROCESSES

A cooking recipe is a well known transformation process. For this reason and its simplicity, we propose to illustrate our ontology on the cooking domain. We present, in the following, the ontology, its concepts and relations and an example of recipe. Finally, we illustrate examples of forms of uncertainty that can be found in a transformation process.

3.1 Our Ontology

An ontology is designed to represent the knowledge on a domain with concepts, relations between these concepts and instances of these concepts (Garzillo et al., 2009). When defining an ontology, it is important to refer to an upper level ontology to guarantee its genericity. Muljarto et. al. defines an ontology for food transformation extending the upper level ontology DOLCE (Muljarto et al., 2014). In this paper, we propose, instead, to extend the Suggested Upper level Merged Ontology (SUMO) because it separates physical from abstract entities and gives a definition of object, separated from the definition of process.

Despres presents an ontology of numeric cooking (Despres, 2014). We keep four of the concepts introduced in her work: ingrédient called product, matériel called device (using the SUMO concept’s name), technique de base called operation and étapes de réalisation, realization step. To these, we add two concepts, the concept attribute already defined in the SUMO ontology, and the concept observation that records the values assumed by the attribute during the process. Figure 3 presents the general relation schema of these concepts that are detailed below.

![Figure 3: The general relation schema of the concepts used to describe the proposed ontology. Subconcepts are connected with discontinuous lines.](image)
3.1.1 Concepts and their Relations

We define a recipe (a transformation process) as a sequence of realization steps. Each realization step is composed of one or more operation(s) applied either to one or more product(s) using one or more device(s) or to a device in order to change some of its properties. The product output(s) of one operation can be the input of another following it in the sequence given by the recipe. In Figure 4, we report part of the SUMO ontology highlighting the concepts we use and the ones we define.

SUMO’s Subconcept Hierarchy Tree

- entity
  - physical
    - object
      - self connected object
        - corpuscular object
          - organic object
            - artifact
              - device
            - ... 
      - context bearing object
        - observation
          - product
            - food
              - raw
              - fruit or vegetable
              - beverage
              - intermediary mixture
            - ... 
            - process
              - intentional process
                - making
                  - cooking
                    - realization step
                      - operation
                        - unitary operation
                        - temporal operation
                      - ... 
                        - ... 
            - ... 
            - abstract
              - quantity
                - number
                - physical quantity
              - attribute
              - ... 

Figure 4: Part of the SUMO ontology, highlighting in italic the concepts we use and in bold the concepts we define. We have omitted part of the concepts we do not use.

In the SUMO ontology, cooking is a subconcept of process. We define two subconcepts of the process cooking: operation and realization step. An operation can be applied to a device. For example, the operation of pre-heating the oven at a certain temperature has as input the device oven and operates changing its state. An operation can also be applied to one or more product(s). The device mixer can be used to whip eggs, whipping takes as input eggs and returns eggs with changed properties. The operation whipping uses the device mixer to modify some of the properties of the object eggs given as input. Another example of operation applied to one or more products is the operation of mixing flour and sugar. The device spoon and bowl are used by the operation. The device spoon is used to mix the two products in a bowl, to return a product that is an intermediary mixture.

In the SUMO ontology, food and device are subconcepts of the concept object. We define a subconcept of object that is superconcept of the concept food. We call it product. This can be a food or an intermediary mixture with its own recipe. For instance, flour is an ingredient of a recipe of a cake, it is a food and so a product; the mix made of flour and sugar ready to be added to eggs in the cake baking process is the output of the mixing operation; the cream to be put on top of a cake is an ingredient’s of the recipe which can be separately prepared with its own recipe.

The SUMO concept attribute represents qualities of objects or operations. The food flour has attribute type which can have value ‘whole grain’, the device oven has attribute temperature which can have value ‘280°’ and the operation mix has attribute speed with value ‘quick’. To record the values of the attributes we define the concept observation as a sub-concept of the content bearing object SUMO concept. While making a cake, we can observe the mixture of flour and sugar and record its color and temperature (color and temperature are attributes of the mixture, the observations about them are collected in the observation). While observing the mixture of butter and sugar we will register also its granularity. Observations cannot be modified by the transformation process.

In a recipe, there are operations that have a duration, we call them temporal operations and we differentiate them from unitary operations. Temporal properties can be described by the time ontology of the semantic web proposed in (Hobbs and Pan, 2004). Temporal operation is a subconcept of the time ontology concept interval; unitary operation is a subconcept of the concept instant; those are both subconcepts of the time concept temporal entity (Figure 5). Thus, we can use properties of the time concept temporal entity to represent temporal relations between operations and so partially ordering the operations of a recipe in realization steps.

3.1.2 A Recipe Example

The TAAABLE project has the purpose of solving

2A content bearing object is defined as a self connected object which expresses information.

3http://www.w3.org/TR/owl-time/

4http://intoweb.loria.fr/taaable3ccc/
cooking problems on the basis of a recipe book (Badra et al., 2008). They propose preparation graphs of a set of recipes that their system has analysed with the ontology presented in (Despres, 2014). Being produced automatically, the generated graphs may contain errors. Consider, for instance, the following recipe for the Aunt Lila’s cookies:

\begin{verbatim}
Aunt Lila’s cookies

\frac{1}{2} lb butter
2 c Nuts ground
2 c All-purposes flour
4 tb Sugar
2 ts Vanilla
to roll Powdered sugar

Preheat oven to 180\degree C. Cream sugar and butter until light and fluffy. Add vanilla and nuts. Add flour gradually. Roll into small balls. Place on baking sheet. Bake 15 to 20 minutes. Roll baked balls in powdered sugar while still warm.
\end{verbatim}

The graph for this recipe reported on the TAAABLE Wiki presents some errors. In particular, for the phrase ‘roll baked balls in powdered sugar’, the automatic system recognizes as ingredient the proposition ‘in’ and as operation the term ‘powdered’. Given the graph errors and the differences between the two ontologies, we propose the graph of Figure 6.

\begin{verbatim}
Figure 6: The preparation graph for the Aunt Lila’s snowball cookies based on our ontology.
\end{verbatim}

The operation preheat the oven is a temporal operation which relates with an observation (the $x^o$ in the rhombus in Figure 6). Representing the observation of the temperature of the oven during time, could help a decision process on when to put the cookies in the oven, which can be an uncertain information.

### 3.2 Uncertainty in Transformation Processes

Data and knowledge in transformation processes are widely tainted with uncertainty. Often the instruments used to take measurements during a transformation process are able to return only an estimation of the quantity observed. Devices are generally calibrated according to some environmental conditions that can be difficult to be repeated somewhere else. They also have some built-in characteristics that are different from device to device. Moreover, the problems of missing data (e.g. the salt ingredient is not always mentioned in a recipe) and missing values (e.g. “to roll” powdered sugar) are known problems in transformation processes. Our aim is to provide a model able to handle all these uncertainties.

Different languages model uncertainty in ontologies. BayesOWL (Pan et al., 2005), OntoBayes (Yang and Calmet, 2005) and PR-OWL (da Costa et al., 2008; Carvalho et al., 2013) are extensions of the Web Ontology Language called OWL to model uncertainty in semantic web. PROWL provides a method to write ontologies containing probabilistic information. This information can be processed but it cannot be enriched as in the case of learning or updating from new data. BayesOWL and OntoBayes add to the ontology a BN that models the uncertainty on the domain, providing a pair ontology-BN. In (Helsper and van der Gaag, 2002) BNs are built to integrate knowledge expressed by experts in an ontology. The BNs built with these approaches cannot summarize the information contained in the ontology because BNs cannot represent relational information. In this way, the two models need to be paired.

Different approaches have been presented that map ontologies into BNs, see for instance (Devitt et al., 2006) and (Fenz, 2012) where, with different approaches, BNs are built starting from a knowledge base modelled as an ontology. These approaches take advantage of the information provided by the ontology, simplifying the BN learning. Learning a BN, they flatten the information coming from the ontology loosing its relational aspect.

The method proposed in (Truong et al., 2005) brings together ontology and PRMs, merging them in a new model on which different types of reasoning are supported. To implement Bayesian reasoning on this model, a BN is constructed from the unified model. In this way, as in the works above, the reasoning is done on a BN and not on probabilistic relational model.
In (Ishak et al., 2011) an approach for learning probabilistic graphical models from ontology is presented. Their approach learns object-oriented BNs by morphing a given ontology. Object-oriented BNs are another extension of BNs using the object-oriented paradigm. Differently from PRMs, object-oriented BNs cannot manipulate reference slots but determine a set of “interface” nodes which allow the communication between objects. Thus, object-oriented BNs are less generic and, in our opinion, less suitable (because less similar) to ontology morphing than PRMs.

With the aim of maintaining the structural and relational information expressed in the ontology, we present, in this paper our mapping of an ontology of transformation processes into PRMs. Having a PRM for the Aunt Lila’s cookies recipe would help us reasoning about different questions that are not possible to be answered with an ontology. For instance, we could compute the probability of having tasty Aunt Lila’s cookies, given the fact that we have/haven’t cream very well butter and sugar (this is the prediction problem). We could also infer the probability of having done a good job in creaming butter and sugar having observed very tasty cookies (inference problem). The defined PRM can be used to suggest a specific sequence of operations to obtain a certain output. For instance, given the butter at a certain temperature, we could suggest the best speed at which using the mixer to cream it with sugar (process control). Finally, we could use the PRM to simulate experiences under different condition.

4 MAPPING

Our approach maps a transformation processes ontology into a PRM’s relational schema. We describe the mapping for the ontology’s concepts: object, unitary operation, attribute and observation.

The SUMO concept object and its subconcepts product, device and observation (see Figure 3) is represented by a class (called class object).

Definition. A class object in a PRM is a mapping between properties of the ontology concepts object and PRM attributes.

In Figure 7, the concept input1 with properties att1 and att2 is mapped into the class object Obj.input1 with attributes the variables att1 and att2. We propose to represent the concept unitary operation by a specific class: the operation class.

Definition. A class operation in a PRM is defined by (1) a DAG over

- the reference slots giving access to the properties of the classes mapping the input object(s) and the device object(s) of the operation,
- an attribute for each property of the operation and
- the attributes representing the properties of the output object(s) of the operation;

and (2) a probability distribution over the attributes representing the properties of the results objects of the operation given the values of the attributes representing the input and the device objects properties.

Figure 7 shows (at the top) the relational schema and (at the bottom) the PRM for two classes operation: operation1 and operation2. The output of the first operation is input for the other, so a reference slot (ρ3) exists between the two classes. Each class object representing the inputs and the device (Obj.input1, Obj.input2, Obj.Device1, Obj.input3 and Obj.Device2) are referred to by a reference slot in the class operation (p1, p2, p3, p5 and p6). The attributes representing the properties of the output object of the operation (att4, att5) define a class to which other classes operation can refer (see ρ4 in Figure 7)².

A temporal operation is mapped with a concatenation of (unitary) operation. Following the standard definition of dynamic BNs (Murphy, 2002) we can define a PRM mapping a temporal operation.

Definition. A temporal operation class maps a temporal operation as a pair of classes operation with a reference slot among them:

²With respect to the literature on PRMs, we should represent the attributes representing the properties of the object output of the operation as a class outside the class operation. Here, we represent it inside, to mean that the output is, indeed, a superclass of the operation itself.
• one (operation) representing the dependencies between variables at the beginning of the operation and
• another (operation→) representing the dependencies from the generic instant of time i to the next instant i + 1, with a reference slot to itself.

The second class operation (operation→) refers to itself, creating a (possibly infinite) loop. To avoid the loop to run forever, we fix the number of times this class can refer to itself. In this way, we ensure the overall model to describe a probability distribution.

Figure 8 shows the relational schema of the PRM for a temporal and a unitary operation classes. As before, the output of the temporal operation is input for the unitary one, so a reference slot exists between the two classes. The output of the class operation operation0 is input of the class operation→. A reference slot exists, also, between operation→ and itself. The number of time the temporal operation class can refer to itself is fixed (reported in the triangle).

Figure 8: The relational schema of the PRM for a temporal operation class linked to a unitary operation class.

An ontology of transformation processes is mapped into a relational schema of a PRM that is a concatenation of classes representing realization steps chained by reference slots. In our ontology, attributes are abstract entities representing properties of object or processes. We map ontology’s attributes, in the PRM, as attributes of the classes mapping the objects of which they represent the property. Finally, observations are ontology concepts that record a particular measurement done over an object or process. In a PRM, an observation is mapped to a class to which an attribute can refer to.

4.1 A PRM for the Example

Reasoning about mapping an ontology for transformation processes in a PRM leads us to better define the ontology. In a BN, the conditional probability distribution of a node depends upon the number of its parents. Referring to the Aunt Lila’s cookies example, the ontology of the operation add in Figure 6 is the same no matter the number of products we have to add together. For a PRM, instead, changing the number of parents of an attribute changes its conditional probability distribution. Following this observation, we enrich our ontology with concepts specifying the number of inputs each operation can have. We replace the operation add with two subclasses add2 (Figure 9). Then we map the new ontology into a PRM following the approach presented in the previous subsection. In the following, we report the mapping for only three operations.

The operation add2 is mapped in a PRM with three reference slots, two for the inputs of the operation (nuts and vanilla) and one for the device used by the operation (bowl). The PRM defines a class mixture1 output of the operation. In Figure 9 the relational schema of this PRM with arrows representing possible dependencies between the attributes of the classes are reported.

Figure 9: The PRM for the operation add2.

The operation bake is a temporal operation. It is represented by a pair of classes: one representing how the operation bake starts, the other representing the probability distribution of the process of baking. The PRM for the operation bake reported in Figure 10 is equivalent to a PRM consisting of the first class in the pair and 20 copies (if the duration of a time step is equivalent to 1 minute) of the second. Being mixture4 an output of the making balls operation, it is formed by small balls to be put in the oven. The concept mixture4 has property the diameter of the balls that is mapped as an attribute of the PRM class mixture4. The diameter attribute of mixture4 influences the consistency of the output of the baking operation mixture5, as expressed by the probabilistic dependency that exists between these two attributes.

The operation add gradually is a special temporal operation because the ontology does not give us the number of times the probabilistic model has to loop over the second class in the pair before passing to the operation that is next to it (Figure 11). We are cur-
Currently reasoning about two possible solutions to treat this problem. The first one being to rely on structure uncertainty. If a probabilistic distribution $p$ on the number of times the loop has to be done is given, we can make the structure uncertain. We add a parameter $\theta$ parent of the operation following the temporal one. The probability of the operation given $\theta$ is given by $p$. The second being to define a simulation process on top of the PRM ruled by the conditions underlining the exit of the loop (e.g. cook till brown). We condition the loop exit to the truth of this condition.

5 CONCLUSIONS

We presented how to map an ontology of transformation processes to a PRMs’s relational schema. The probabilistic model defined starting from the ontology is a powerful reasoning tool. It integrates data information into the relational schema obtained from the ontology. Incorporating this information, we could deal with common data mining problems such as missing data and data integration. We propose to combine the two models while maintaining them separate: each formalism can benefit from the strength of the other and be, at the same time, a standing-alone model. We illustrate our mapping on an ontology of transformation processes in the cooking domain, relying on the SUMO upper level ontology.

We propose a methodology able to automatically map SUMO physical concepts (objects and processes) into PRM classes and the SUMO abstract entity attribute into PRM attributes. We propose a mapping for the ontology concepts operation and temporal operation. To map the former into a PRM we extend the standard definition of PRMs with ideas used in dynamic Bayesian networks. To map temporal operations that have an uncertain stop criterion, we propose the use of structure uncertainty or the definition of a simulation process over the sequence of operations. These have drawbacks that we are studying.

Learning PRMs is an NP hard problem that can be compared to learning Bayesian networks. Acquiring the parameters of a PRM knowing its relational schema is much easier. Even if we do not have experimental result on that, we think that we can say that learning the PRM of a transformation process whose relational schema has been obtained mapping the ontology of that transformation process is much easier than learning it from scratch.

We plan to pair the proposed approach with an algorithm for learning PRM’s parameters. This will provide the possibility to experiment the proposed approach. Finally, we would like to apply our mapping to other transformation processes such as microorganism production and stabilization processes.

REFERENCES


