FACILITATING AND EXPEDITING THE DEVELOPMENT OF SYSTEMS THAT HANDLE UNCERTAINTY THROUGH PROMODEL

Rosa López-Landa, Marcos Bernal and Julieta Noguez
Tecnológico de Monterrey, Campus Ciudad de México
Distrito Federal, México
email: {atzimba.lopez, A00922653, jnoguez}@itesm.mx

ABSTRACT
This paper introduces PRoModel, a Web tool that expedites the development of applications that handle uncertainty. PRoModel is based on a service-oriented, model-driven and multilayered architecture that provides flexible and reusable applications. The graphical user interface developed in the presentation layer facilitates the design of Probabilistic Relational Models (PRMs), allowing the inclusion of (1) the entities of the problem and their relationships from an Entity-Relationship (ER) model, and (2) the random variables and conditional dependence associations from some Probabilistic Graphical Models (PGMs), such as Bayesian networks. Once the user completes the design of the PRM, the logic layer of PRoModel transforms the model to generate a functional Web application prototype that facilitates the management of the domain data and provides the initial PGM. To demonstrate the use of PRoModel we have generated an Intelligent Tutoring System (ITS) for training operators of power plants. Initial tests show that PRoModel expedites and facilitates the development of Web applications with uncertainty models, providing significant savings in time and effort without requiring the user to be an expert in systems development.

KEY WORDS
Graphical user interfaces, intelligent tutoring systems, model-driven development, probabilistic relational models, software architectures, Web-based software engineering.

1. Introduction
Currently, many systems are built to deal with the uncertainty of the data collected through different sources. Unfortunately, these applications had many limitations, since they could not easily be extended or maintained, mainly because most of them are standalone applications that do not interact with databases or other systems. An additional problem is the difficulty of merging, in the same application, the domain data and the uncertainty model. In this direction, Koller and Friedman [1] provided a model called Probabilistic Relational Models (PRMs), which combine in the same model the entities, attributes and relations of the domain, with the expressive power of Bayesian networks (BNs) [2, 3] to manage the uncertain knowledge.

However, until the writing of this paper, there is no tool to develop these combined models. Taking this into account, we designed a service-oriented (SOA) [4, 5] and model-driven (MDA) [6] tool for the development of Web applications that handle uncertainty. The MDA and SOA concepts, standards and tools help us achieve a semi-automated domain-based design system, called PRoModel, that automates the generation of Web applications through the transformation of a PRM.

Web applications become popular recently because, unlike the standalone systems, they are distributed and accessible to a large number of users. Structurally, Web applications are based on a client-server architecture, where each client makes requests to one or more service and resource providers.

The remaining of this paper is organized as follows. Section 2 introduces the PRMs. Section 3 presents PRoModel, a Web tool for the development of systems that handle uncertainty. In section 4 we describe the development process of a system through PRoModel. We also address the results from an evaluation process of the tool. In section 5 we present some related work, along with several software tools that handle uncertainty. Finally we present our conclusions and future work in section 6.

2. Probabilistic relational models
Working with uncertainty means that we don’t have all the information associated with a problem of certain domain. Uncertainty arise when information is incomplete or incorrect or if we have deficient models that limit the way in which knowledge can be represented [7, 8], Bayesian networks are a widely accepted technique to represent uncertain knowledge. This models are able to infer information, through the probability theory, by calculating conditional or a posteriori probabilities from the unconditional or a priori probabilities of each random variable [8], through the chain rule (see Equation 1). Influence diagrams (IDs) [9, 10] are BNs extension that represent the current state of a problem, the possible actions to be performed and the usefulness of the action performed. These models merge the probability, decision and utility theories to choose actions based on what is believed and desired [8].
\[ P(X_1, \ldots, X_n) = \prod_{i=1}^{k} P(X_i | Pa(X_i)) \]  

Even though Bayesian networks and influence diagrams are very powerful techniques for the representation of uncertain knowledge, they have a significant disadvantage: each node unknowns the variable attributes that shape the domain problem [1, 11]. A probabilistic relational model extends BNs with the concepts of objects, properties and relations between them. This way, a PRM consists of [12, 13]:

- A set of entities \( E = \{E_1, E_2, \ldots, E_n\} \).
- A set of descriptive attributes \( A = \{A_1, A_2, \ldots, A_n\} \) for each entity \( E_X \in E \). Similarly each attribute has associated a domain type.
- A set of relationships, along with their multiplicity.
- A set of random variables \( V = \{V_1, V_2, \ldots, V_n\} \) for each entity \( E_X \in E \). In the same way, each random variable has a set of two or more states.
- A set of conditional dependence associations. Additionally, each association has attached a conditional probability table.

Like Howard et al. [13], in this paper we use a relational representation of the domain, given by an Entity-Relationship (ER) model [14], in combination with Bayesian networks to generate probabilistic relational models. Moreover, we intend to expand the types of applications that combine uncertain knowledge with the relational domain, so we have included influence diagrams and dynamic Bayesian networks (DBN) [15] in an extended PRM.

3. **PRoModel: System overview**

PRoModel is a Web tool for the development of systems that handle uncertainty. Our aim is to develop Web applications by transforming extended UML diagrams into Spring Roo [16] and Elvira [17] scripts, using the Atlas Transformation Language (ATL) [18], as in [19]. By doing so, domain experts will not require deep software engineering knowledge, since they only require modeling the domain problem.

An essential requirement of PRoModel was to improve the creation of systems with uncertainty, regardless of the inference or propagation tool that will be used. It was also required that PRoModel should be flexible enough to allow the integration of existing frameworks, PGMs or components. For this reason we used a service-oriented, model-driven and multilayered architecture that allows the generation of a flexible and reusable application. The architecture we use (shown in Figure 1) consists of five layers, that interact via pre-defined services.
the design of a PRM through a class diagram. This extended UML model is the initial platform-independent model (PIM) from which we extract the domain model and the uncertainty model into an entity-relationship model and a Bayesian network, respectively. Then, through another set transformation rules, we generate the platform-specific models (PSMs) for:

- A Web application, represented by a Spring Roo script.
- The uncertainty model for the inference tool, represented by an Elvira file.

At last, we execute the Web application script using the Spring Roo command tools to generate a functional Web application prototype and a working database. It should be noted that although for the moment we use Elvira, we can easily incorporate new tools for handling uncertainty (such as OpenMarkov [20]) by defining the corresponding PSM and transformation rules. This way, users can choose the inference tool that best suits their needs. Finally, it is important to note that the entire transformation process is automatic, and can be fully executed from the Web.

3.1. Graphical editor

It is not possible to create a PRM with the tools that are available today. For this reason we have developed a graphical user interface (GUI) that combines ER models and PGMs into a probabilistic relational model. PRoModel’s GUI (see Figure 3) allows the creation of a PRM through the definition of:

1. A domain model, represented by an entity-relationship model.
   a) Add the entities.
   b) Add the attributes of each entity, setting the appropriate data type.
   c) Create the associations among entities, along with their multiplicity.

2. A probabilistic graphical model, depicted by a Bayesian network or influence diagram.
   a) Add the random variables of the uncertainty model. At this point it is necessary to define the states that are associated with the variable.
   b) Add the decision and utility variables of the uncertainty model (if applicable).
   c) Define the conditional dependence associations within the random variables.
   d) Define the Conditional Probability Table (CPT) of each relation between random variables.

Once the PRM is completed, users can transform the model into a Web applicatio and a PGM, through the process described in the previous section. Any changes to the model can be made before or after the transformation process, so it will be easy to reuse models. It is important
to note that PRoModel has a collaboration mechanism through which many users can share and edit the same PRM, using concurrency control. Finally, it is also worth mentioning that we have developed an interface that allows the execution of the inference and propagation algorithms (from Elvira) through the generated Web application. This way users can infer knowledge from the information stored in the database of their system.

4. Evaluation

In order to evaluate the impact of PRoModel on developing systems that handle uncertainty, we have designed an Intelligent Tutoring System (ITS) [21] for training operators of power plants. Similarly, we have applied an usability test survey after a demonstration session of PRoModel. Below we describe the ITS development process and the results of the usability test survey.

4.1. Study case: An intelligent tutoring system for training operators of power plants

The ITS we have developed is based on previous work described in [22, 23, 24]. Briefly, intelligent tutoring systems are tools that incorporate artificial intelligence techniques to process knowledge, in certain domain, with the aim of helping students gain knowledge during the teaching-learning process. To achieve this, the ITS should be able to simulate human students and tutors [25]. An ITS can be designed as part of an academic course or to assist employees in their professional training [21]. The components of the classical architecture of an ITS, proposed by Wenger (1987), are: the domain model, the student model, the tutor model and the communication interface [26].

For the study case developed, we have focused on the student model [22, 27, 28], which evaluates and selects pedagogical actions required for each student. This model also contains knowledge about the progress of each student, their preferences, history, mistakes and learning styles. Modeling the student poses a challenge in ITS development, as knowledge, preferences and the student’s status is constantly changing over time, drifting incomplete and uncertain models. For this reason, some ITS use Bayesian networks and influence diagrams to treat the uncertainty of the student model [29].

For the definition of the student model we have designed a probabilistic relational model composed by (1) an entity-relationship model that represents the students, the course and the experiments; and (2) a Bayesian network that allows the inference of the student’s knowledge. It is important to note that this process involved the participation of experts in physics and pedagogy, as well as some staff members of the power plant. This process required that many hours were spent in defining and modeling the

![Figure 4. Probabilistic relational model of the ITS for training operators of power plants.](image-url)
domain. However, this process should be done to model any system with uncertainty.

Once the PRM has been defined, a user captured it through PRoModel’s GUI (see Figure 4). Then she had executed the transformation process of PRoModel, described in section 3. The GUI of the Web application generated by PRoModel is shown in Figure 5(a). In this case, the ITS required the use of a simulator that allows the operator to experiment with the variables in the power plant. For this reason, the interface generated by PRoModel was replaced by the 3D simulator shown in Figure 5(b). Nevertheless, the database and code artifacts associated with services, data access objects and entities were preserved.

In the same way, during the transformation process of PRoModel, the Bayesian network was automatically extracted, so users can utilize Elvira as propagation and inference engine. The extracted Bayesian network is shown in Figure 6.

It is important to note that the entire transformation process took less than ten minutes. In this way, PRoModel facilitated and expedited the development of the database, the code artifacts and the probabilistic graphical model of the ITS. Similarly, the layered and service-oriented architecture facilitates the substitution of the generated GUI with a 3D simulator.

The ITS was pre-evaluated by six professors from the e-learning research group of the Tecnológico de Monterrey, Campus Ciudad de México (ITESM-CCM). However, there will be tests with operators of power plants. These tests and their results are outside the scope of this paper, because our goal is to test the operation of PRoModel, and not to assess the knowledge of operators of power plants.
4.2. Usability test survey

Usability tests have been extensively used in industry to evaluate system prototypes and documentation. The aim of our evaluation process was to assess the performance, ease of use and attractiveness of PRoModel. To accomplish this, we conducted a demonstration session attended by twelve professors from the e-learning research group of the ITESM-CCM. These professors have different specialties, such as mathematics, physics, computer graphics, robotics and software engineering. During the demonstration session we presented an overview of PRoModel, introducing the concepts of software architectures and PGMs. Then we used the study case of the ITS described above to create a PRM through the graphical editor of PRoModel. Then, we executed the transformation of the model. After forty minutes into the session we had generated a functional Web application prototype (with a working database) and a Bayesian network.

As a post-test activity, participants were asked to rate PRoModel’s performance, ease of use and attractiveness. Results of the evaluation process are discussed in the following section. A demonstrative video of PRoModel can be found at http://ito.mx/M_b6.

4.3. Results and discussion

Table 1 and Figure 7(a) show the most representative questions and results of the usability test survey of PRoModel. In general we can see that all the participants would recommend this tool to their colleagues, since they considered that PRoModel facilitates and expedites the development of systems that handle uncertainty. These answers also indicate that the participants felt comfortable using our tool, due to the fact that the graphic editor is very friendly. However, we can also see that two participants felt that there are too many steps to generate a model, therefore, we will need to review the process described in section 3.1.

Additionally, the test participants were asked to rate PRoModel on a scale from 1 (excellent) to 6 (dreadful), taking into account that the aim of the tool is to facilitate and expedite the development of systems that handle uncertainty. As shown in Figure 7(b), ten of twelve participants believe that PRoModel is a very useful (or excellent) tool for the development of this kind of systems. However, one participant noted that this tool is very useless. This may be related to the fact that we presented a prototype that does not have all the functionality expected by the participants.

5. Related Work

In recent years there has been a lot of work supporting transformations from models into source code artifacts [30, 31, 32, 33]. Regarding Web application development, we can mention the work of Hou et al. [34], where PIMs, based on UML extensions, are transformed into Web application PSMs using algebra methods. Closer to our approach is the work of Abdella [35], who build Java Web applications based on extended UML class diagrams. However, we attempt to build functional prototypes of systems that handle uncertainty, from extended UML class diagrams, instead of just code artifact templates.
5.1. Software tools to handle uncertainty

There are a number of tools that provide support for the development of PGMs. Some of these tools are shown in Table 2, but a more complete list can be found in [36]. Even though these tools provide modeling support for Bayesian networks and/or influence diagrams, none allows embedding relational attributes with uncertain knowledge in the same model.

Table 2. Some PGM modeling tools.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Web</th>
<th>Model</th>
<th>PRM support</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesiaLab</td>
<td>No</td>
<td>BN/ID</td>
<td>No</td>
</tr>
<tr>
<td>Elvira</td>
<td>No</td>
<td>BN/ID</td>
<td>No</td>
</tr>
<tr>
<td>Netica</td>
<td>No</td>
<td>BN/ID</td>
<td>No</td>
</tr>
<tr>
<td>OpenMarkov</td>
<td>No</td>
<td>BN/ID</td>
<td>No</td>
</tr>
<tr>
<td>ProBT</td>
<td>No</td>
<td>BN/ID</td>
<td>No</td>
</tr>
<tr>
<td>RISO</td>
<td>Yes</td>
<td>BN</td>
<td>No</td>
</tr>
</tbody>
</table>

The essential difference between these tools and the tool we have developed is that our tool can represent a PRM through an extended UML diagram. Therefore, we could have, in the same model, a relational representation of our domain and an uncertain representation of our knowledge.

6. Conclusions and further work

In this paper, we have developed a Web application that facilitates the development of systems that handle uncertainty. ProModel, the presented tool, allows the automatic generation of Web applications and PGMs based on the extended probabilistic relational models, that we have also presented. Because at the present there are no tools for the design of PRMs, we have developed a graphical editor for a Web environment. Our tool is based on a service-oriented, model-driven and multilayered architecture that provides flexible and reusable applications.

We have evaluated the system with the development of an intelligent tutoring system for training operators of power plants, and an usability test survey, performed at the end of a demonstrative session. The results of the initial tests show that ProModel facilitates and expedites the development of Web applications, providing significant savings in time and effort without requiring the user to be an expert in systems development, because users only need to capture their PRM through the GUI of ProModel.

Finally, it is important to note that ProModel is a tool that is in development, so we have a lot of work to do, as: (1) improving the PRM editor; (2) adding explanations to the PGMs; (3) including dynamic models, such as DBNs; (4) adding PSM models to include other uncertainty tools; (5) including reverse engineering capabilities to be able to generate the ER model from an existing database; and (6) integrating learning tools to generate the PGMs automatically from an existing data set.

Acknowledgments

This work was supported by a grant provided by the Consejo Nacional de Ciencia y Tecnología (CONACyT) and the ITESM-CCM. In addition this tool is part of a project called "Dynamic Probabilistic Graphical Models and their Applications", number 95185, funded by CONACyT and the European Union through the Fondo de Cooperación Internacional en Ciencia y Tecnología (FONCICYT). We also would like to thank Alberto Reyes, Daniel Blancas, Jesse Jacobs and Iván Pérez for their collaboration in the development of the ITS. Finally, we appreciate the participation of the members of the e-learning research group at the ITESM-CCM during the demonstrative session and usability test of the system.

References


