CONCEPTUAL MODELLING OF KNOWLEDGE-BASED SYSTEMS USING UML

Mohd Syazwan Abdullah[1,3], Richard Paige[1], Carl Thompson[2], Ian Benest[1], and Chris Kimble[1]

[1] Department of Computer Science, [2]Department of Health Sciences, [3]Faculty of Info Tech
[1,2]University of York,, Heslington, York, United Kingdom, [3]Universiti Utara Malaysia, Sintok
[1]{syazwan, paige, idb, kimble}@cs.york.ac.uk , [2]cat4@york.ac.uk, [3]pathma @uum.edu.my

Abstract: Conceptual modelling is an important aspect in designing systems. However, the use of conceptual models in knowledge-based system (KBS) is limited as there is no particular consensus on which modelling language should be used and most of these systems are developed in a 'problem to code' manner. This paper focuses on the use ofa Unified Modeling Language (UML) Profile for conceptual knowledge modelling. The profile is created using the profile extension approach of UML and is based on the XMF (eXecutable Modelling Framework). An example of modelling a KBS based on the Ottawa Ankle Rule demonstrates the use of the profile.

Key words: conceptual modelling, knowledge-based system, UML Profile

1. INTRODUCTION

Knowledge-based systems (KBS) are developed using knowledge engineering (KE) techniques [1], which are similar to those used in software engineering (SE), but place an emphasis on knowledge rather than on data or information processing. As such, they inherently advocate an engineering approach to the process of developing a KBS. Central to this is the conceptual modelling of the system during the analysis and design stages of the development process. Many KE methodologies have been developed with an emphasis on the use of models and conceptual modelling, for example: CommonKADS [2], MIKE [3], Protégé [4], and KARL [3].

In first generation expert systems, the knowledge of the expert (or experts) was captured and translated into a set of rules. This was essentially a process of knowledge transfer [3]. The disadvantage of this approach is that the captured knowledge in the form of hard-coded rules within the system provides little understanding of how the rules are linked or connected with each other [2]. As a result, when the knowledge base needs updating, there is a substantial effort required to ensure that the knowledge base remains correct. KE is no longer simply a means of mining the knowledge from the expert's head [2]. It now encompasses *"methods and techniques for knowledge acquisition, modelling, representation and use of knowledge"* [2].

This paper demonstrates a systematic approach to modelling and designing KBSs in a purely object-oriented fashion through the use of UML's profile mechanism. The novelty of the system design lies in the profile that is used to create it. The profile is constructed using compliant standards for modelling software systems by adopting the XMF approach [12]. XMF uses standard modelling techniques taken from SE. It provides tool support for designing and verifying models as well as executing the models. It is one of the latest techniques in modelling and this work demonstrates the use of this approach. A case study on modelling a KBS based on the Ottawa Ankle Rule demonstrates the ability of the profile.

This paper is organised as follows: Section 2 generally describes the process of designing knowledge-based systems. Section 3 gives an overview of UML and the profile extension mechanism. Section 4 explains the XMF profile design approach and presents the abstract syntax model of the knowledge modelling profile and discusses the specialisation made to the existing meta-model of XMF. Section 5 illustrates how the profile can be used as part of the development of a KBS, while section 6 concludes and indicates the direction for future work.

2. DESIGNING KNOWLEDGE-BASED SYSTEM

Models are used to capture the essential features of real systems by breaking them down into more manageable parts that are easy to understand and to manipulate. Schreiber *et al* [2] argue that models are important for understanding the working mechanisms within a KBS; such mechanisms are: the tasks, methods, how knowledge is inferred, the domain knowledge and its schemas. A further benefit arising from the shift towards the modelling approach is that fragments of knowledge may be re-used in different areas of the same domain [3] making systems development faster and more efficient. In the past, most knowledge systems had to be developed afresh each time a new system was needed, and it could not interact with other systems in the organization.

Although a KBS is developed using knowledge engineering techniques, the modelling aspects of it are largely dependent on software engineering modelling languages. The development process of a KBS is similar to that used in any general system development; stages such as: requirements gathering, system analysis, system design, system development and implementation are common activities. The stages in KBS development are: business modelling, conceptual modelling, knowledge acquisition, knowledge system design and KBS implementation. Most of the modelling techniques adopt a mix of notations derived from different modelling languages such as: UML, IDEF, SADT, OMT, Multi-perspective Modelling and others. The object-oriented paradigm has influenced systems development activities in software engineering and this trend has also been reflected in knowledge engineering methodologies such as: CommonKADS [2], MOKA [6] and KBS in product configuration as described by Felfernig *et al.* [7].

As there is no standard way of modelling KBS, there is a need to extend the use of standardised software engineering modelling language, such as UML for knowledge modelling. This promotes the use of a common modelling language, so that the vision of integration, reusability and interoperability among enterprise systems can be achieved.

3. KNOWLEDGE MODELLING PROFILE

The Unified Modeling Language (UML) together with the Object Constraint Language (OCL) is the *de facto* standard for object modelling in software engineering as defined by the Object Management Group (OMG). The UML is a general-purpose modelling language that may be used in a wide spectrum of different application domains. The OMG [8] has defined two mechanisms for extending UML: profiles and meta-model extensions. Profiles are sometimes referred to as the "lightweight" extension mechanism of UML [9]. A profile contains a predefined set of Stereotypes, TaggedValues, Constraints, and notation icons that collectively specialize and tailor the UML for a specific domain or process. The main construct in this profile is the stereotype that is purely an extension mechanism. In the model, it is marked as <<stereotype>> and has the same structure (attributes, associations, operations) as defined by the meta-model that describes it. Nevertheless, the usage of stereotypes is restricted, as changes in the semantics, structure, and the introduction of new concepts to the meta-model are not permitted [10]. The "heavyweight" extension mechanism for UML (known as the meta-model extension) is defined through the Meta-Object Facility (MOF) specification [11] which involves the process of defining a new meta-model. Using this extension, new meta-classes and meta-constructors can be added to the UML meta-model. However, it is easier to create a profile using the "lightweight" extension as it is easier to use, easier to introduce new concepts through specialising existing meta-model and has better tool support compared with meta-model extension. It is unfortunate that both extensions are known as profiles. The work presented in this paper incorporates the "lightweight" extension mechanisms of UML using the XMF approach when designing the profile. A brief introduction of XMF is given in Section 4.

4. PROFILE DESIGN – THE XMF APPROACH

The XMF (eXecutable Meta-modelling Framework) is an object-oriented meta-modelling language, and is an extension to existing standards for meta-models such as MOF, OCL and QVT, which are also defined by OMG. XMF exploits the features of these standards and adds a new dimension that allows them to be executable using an associated XMT software tool. The most comprehensive use of these standards are seen in the UML in which its meta-models are described using MOF. Details of XMF can be found in [12]. The XMF approach to profile creation can be divided into three steps: the derivation of an abstract syntax model, a description of the semantics, and a presentation of the profile's concrete syntax. In this paper we only present the creation of the abstract syntax model of the profile. The knowledge modelling profile is supported by all UML tools.

4.1 Abstract Syntax Model

The abstract syntax of the knowledge modelling language has been derived using the modelling concepts reviewed from literature [1, 3-5] and the CommonKADS modelling language (CML) [2]. The modelling concepts are concept (class) that represents the category of things; inference, which performs the reasoning function of the KBS; transfer function used to transfer between the reasoning agent and external entities (system, user); task which defines the reasoning function; task method used to describe the realisation of the task through sub-function decomposition; static role used to specify the collection of domain knowledge (rules) that is used to make the inference; dynamic role which specifies the information flow (input/output) of meaningful facts between the working memory/data base and the inference; rule type which refers to the categorisation of domain knowledge; rule which are expressions that involve an attribute value of a concept; knowledge base that contains the collection of instances of domain knowledge in the form of rules; and fact base which are the collection of attribute instances of concepts stored in working memory or database, upon which the KBS reasoning will be based. The knowledge modelling profile has four packages: domain concept, knowledge base, rule type and inference.

The Domain Concept package within the profile describes the concept constructs of the profile that are related to knowledge elements. This package is shown in Fig.1.

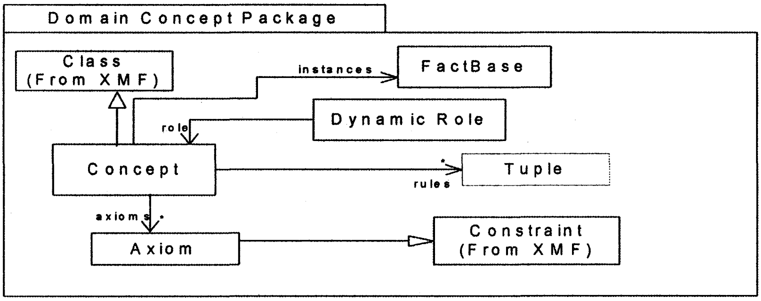

Figure 1. Domain Concept Package

The Knowledge Base package of the profile describes the modelling of a knowledge base that represents instances of knowledge elements (instances of rule type) within the domain concepts. These instances are important as they contain the actual knowledge on which the KBS reasoning process is based. Knowledge elements within the knowledge base are accessed by an inference through a static role. This package is shown in Figure 2.

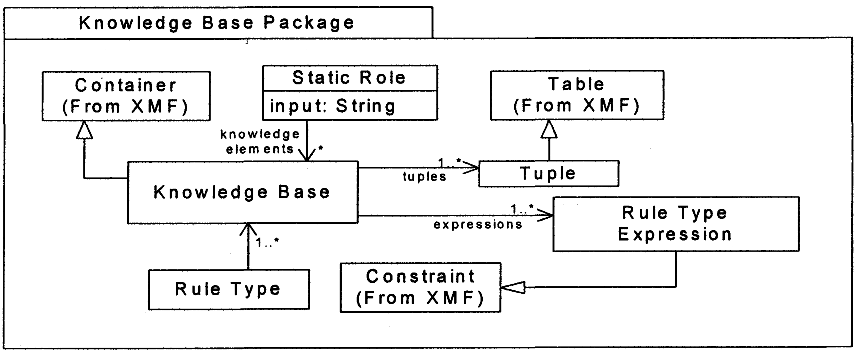

Figure 2. Knowledge Base Package

The Inference package of the profile describes the inference, inference method, task, task method, transfer function and both the static and dynamic knowledge roles. The inference package plays a pivotal role in designing the KBS as it defines the inference structure of the system, the type of knowledge used in the reasoning process and the task associated with the execution of the inference. An important point to note here is that the KBS is designed independently of the target implementation platform and inference engines, overcoming the difficulties of reusing implementation specific designs. This package is shown in Fig.3.

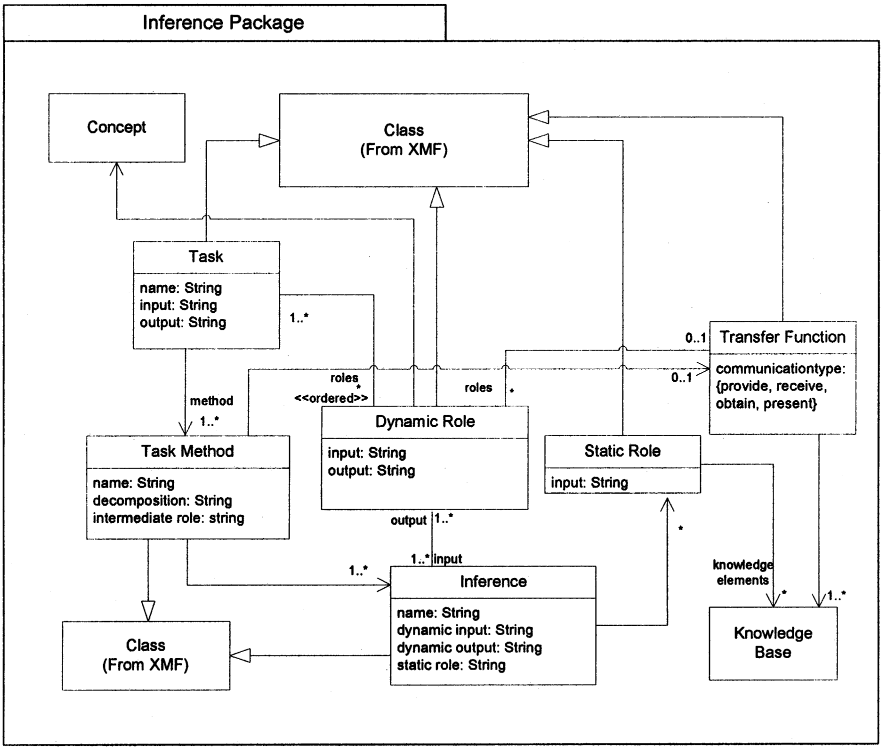

Figure 3. Inference Package

The Rule Type package (shown in Fig. 4) within the profile describes the modelling of rules. There are three types of rule: constraint rule, implication rule and decision table. A decision table is an addition to the used set of rule types. It is introduced here because certain rules are best expressed in the form of a decision table. This paper only concentrates on rule-based KBSs as it is the widely adopted KBS technology.

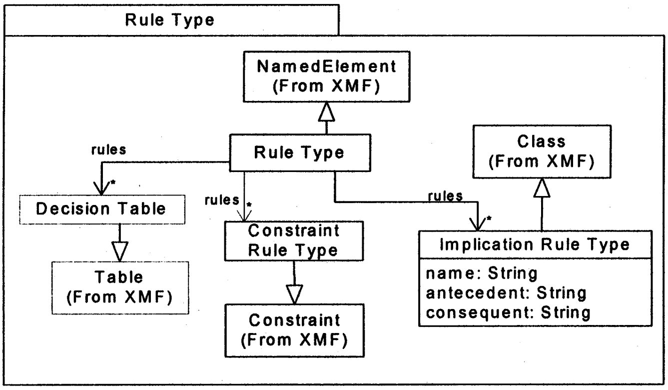

Figure 4. Rule Type Package

4.2 Model Specialisation

The knowledge modelling profile concept extends the existing meta-models of XMF by defining the profile's abstract syntax. There are five places where the profile can be viewed as an extension to XMF and these are: Class, Constraints, Named Element, Container and Table, all of which are central to the Core XMF meta-model.

The knowledge modelling class concept is viewed as a special class that is a subclass of the XMF Class. This enables the concept to inherit all the features of a class and allows it to define additional constraints such as "concepts do not have any operations or methods". The implication rule type, decision table and constraint rule type, are also examples of this. The inference package of the profile (which has the task, task method, inference, dynamic role, static role, and the transfer function concepts) can be viewed as a subclass of an XMF Class. This allows operations related to objects to be expressed, such as an execute inference call from the task method, the execution of the inference process and the access to knowledge in the knowledge base through the static role and at the same time allows the inference package elements to specify attributes. Rule Type is subclassed from NamedElement as all rule sets must have unique name.

The Constraint class is a subclass of the XMF meta-model that incorporates profile concepts such as axioms and rule type expressions. All these concepts need the ability to express constraints and this class allows for this. Knowledge base is a subclass of the Container class of XMF. It has a 'content' slot that is a table. This is a natural choice for a subclass as the knowledge base is actually a collection of tables grouped together in order to store rule type instances. The table class of XMF is extended to incorporate the profile's concepts of tuple.

5. 'OTTAWA ANKLE RULE' SYSTEM

The Ottawa ankle rules was devised by Stiell [13], as an assessment on ankle injury patients at Emergency Department. Physicians use these rules to rule out the possibility of having an ankle X-ray series or a foot X-ray series. The decision rule is based on assessment of the two different type of pain, bone tenderness and the inability to bear weight. Figure 6, adapted from [13] shows the major recommendation of the ankle rules.

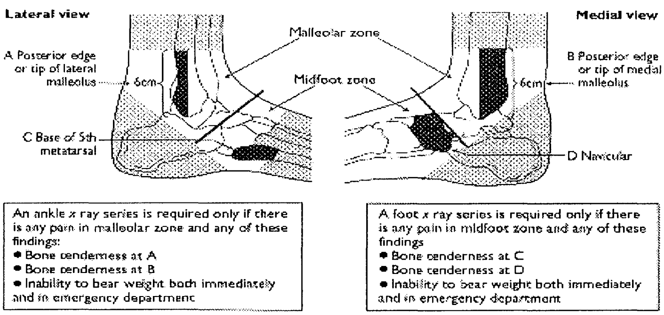

Figure 4. Ottawa ankle rule for use of radiography in acute ankle injuries (adapted from Stiell. *et al* [13])

Figure 5 shows the knowledge modelling profile used to represent the Ottawa ankle rules in a KBS. This example concentrates on showing the abstraction and matching process of the Ottawa application based on the modified task template of assessment task suggested by [2]. The profile packages used here are the domain concept, inference, knowledge base and rule type.

The task 'Assess Injury' will be realised by the task method 'abstract case' and 'match method'. The task method can be decomposed into other tasks or inferences. In this example of 'abstract case', it is decomposed into the inference "abstract". The concept patient has two attributes: name and pain area enumeration type. Here the assessment task will abstract all cases into two groups, based on the pain suffered by the patients. Patients will suffer pain, either in the malleolar zone or in the midfoot area. The purpose of this abstraction process is to provide useful categories of cases that need to be distinguished for assessment purposes. The inference will execute the task of abstracting the cases by adding case abstraction to the pain area data. The reasoning process will use the "pain abstraction knowledge" which is a static knowledge role. This knowledge is accessed from the knowledge base 'Ottawa ankle' and is based on the 'pain abstraction rule'. The next step is to match the whether an ankle or foot x-ray series is required based on the pain area. This is executed by the task method 'match method'. This method is decomposed into the transfer function 'obtain' and inference 'match'. The transfer function will obtain injury details from patient during consultation with the system based on the question regarding bone tenderness and inability to bear weight. The injury details will be used by the inference 'match' to make the decision on the different type of x-ray series to be performed on the patient. This inference will use the rule 'Ottawa ankle rules' to match the decision.

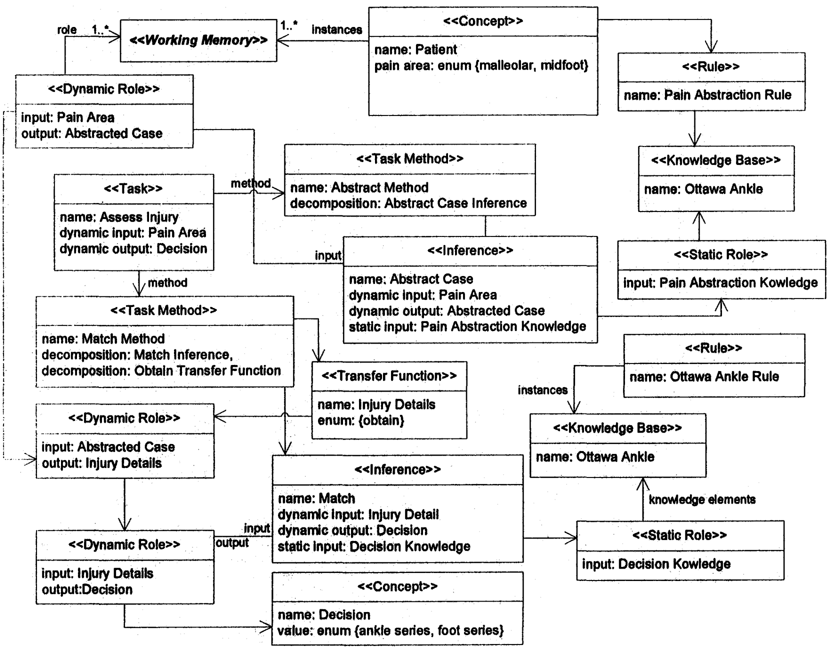

Figure 5. Ottawa Ankle Rule system model

6. CONCLUSION AND FUTURE WORK

Knowledge-based systems have evolved from being stand-alone machines to being part of the enterprise's group of systems. The process of constructing KBSs is similar to that required by other software systems, and conceptual modelling plays an important role in the development process. Software engineering has adopted UML as a standard for modelling, but the field of knowledge engineering is still searching for the right language. UML can be adopted for knowledge modelling by exploiting the profile extension mechanism. This paper has described the process of creating such an extension by basing the design of the profile on that of the XMF framework. This is a novel approach in profile design as the XMF approach is an extension to existing standards for meta-modelling such as MOF, OCL and QVT. The creation of the profile is important as it allows KBS to be designed using an object-oriented approach.

The future work involves the specification of the profile's semantics and construction of the concrete syntax model. Both activities involve the use of the XMT tool, and are in its final stage of development. The profile's ability to model the requirements of KBS has only be tested on two case studies adapted from [2] and [13]. Testing the profile in a number of real-world situations would be beneficial and would identify any limitations and assist in the refinement of the profile. Currently we are testing the profile on case studies from the Center for Evidence Based Nursing at the Department of Health Sciences at York.

REFERENCES

1. Studer, R., Benjamins, R.V., and Fensel, D.: *Knowledge Engineering: Principles and Methods.* Data & Knowledge Engineering, 1998. **25**: p. 161-197.
2. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., de Velde, W.V. and Wielinga, B.: *Knowledge Engineering and Management: The CommonKADS Methodology.* 1999, Massachusetts: MIT Press.
3. Angele, J., Fensel, D., Landes, D., Studer, R.: *Developing Knowledge-Based Systems with MIKE.* J of Automated Software Engineering, 1998. **5**(4): p. 389-418.
4. Grosso, W.E., Eriksson, H., Fergerson, R.W., Gennari, S., Tu, S., Musen, M.A.: *Knowledge Modelling at the Millennium (The Design and Evolution of Protege 2000).* 1999, Stanford Medical Institute.
5. Speel, P., Schreiber, A. Th., van Joolingen, W., and Beijer, G.: *Conceptual Models for Knowledge-Based Systems*, in *Encyclopedia of Computer Science and Technology.* 2001, Marcel Dekker Inc, New York.
6. Stokes, M., *Managing Engineering Knowledge: MOKA - Methodology for Knowledge Based Engineering Applications.* 2001, London, UK: Professional Engineering and Publishing Limited.

7. Felfernig, A., Friedrich, G.E., Jannach, D.: *Generating product configuration knowledge bases from precise domain extended UML models.* in *12 th International Conference on Software Engineering and Knowledge Engineering (SEKE'00).* 2000. Chicago, USA.
8. OMG: *Unified Modeling Language specification (version 1.4).* 2001.
9. OMG: *Requirements for UML Profile.* 1999.
10. Perez-Martinez, J.E.: *Heavyweight extensions to the UML metamodel to describe the C3 architectural style.* ACM SIGSOFT Notes, 2003. **28**(3).
11. OMG: *MOF Specification version 1.4.* 2002.
12. Clark, T., Evans, A., Sammut, P., Willians, J.: *Metamodelling for Model-Driven Development (draft): To be published.* 2005. (Accessible at www.xactium.com)
13. Stiell IG, McKnight RD, Greenberg GH, McDowell I, Nair RC, Wells GA, et al. Implementation of the Ottawa ankle rules. *JAMA* 1994;271:827-32.