

Knowledge Modelling Using The UML Profile

Mohd Syazwan Abdullah^{1,2}, Richard Paige², Ian Benest², and Chris Kimble²

¹ Faculty of Information Technology, UUM, 06010 Sintok, Kedah, Malaysia

²Department of Computer Science, University of York, Heslington, York, YO10 5DD, United Kingdom

¹pathma@uum.edu.my

²{syazwan, paige, idb, kimble} @cs.york.ac.uk

Abstract. This paper discusses platform independent conceptual modeling of a knowledge intensive application, focusing on the use of knowledge-based systems (KBS) in the context of model-driven engineering. An extension to the Unified Modeling Language (UML) for knowledge modeling is presented based on the profiling extension mechanism of UML. The UML profile discussed in this paper has been successfully captured in a Meta-Object-Facility (MOF) based UML tool – the eXecutable Modeling Framework (XMF). The example is that of modeling a knowledge-based system for the Ulcer Clinical Practical Guidelines (CPG) Recommendations. It demonstrates the use of the profile, with the prototype system implemented in the Java Expert System Shell (JESS).

1 Introduction

Knowledge-based systems (KBS) were developed for managing codified knowledge in the field of Artificial Intelligence (AI). Widely known as expert systems, these were originally created to emulate the human expert reasoning process [1] and is one of the successful inventions that has been derived from AI technologies. KBS are developed using knowledge engineering (KE) techniques [2], which are similar to those used in software engineering (SE), but have an emphasis on knowledge rather than on data or information processing.

Central is the conceptual modelling of the system during the analysis and design stages of the development process; this is widely known as knowledge modelling. Many knowledge engineering methodologies have been developed with an emphasis on the use of models, for example: CommonKADS [3]. KBS continue to evolve as the need to have a stable technology for managing knowledge grows. Their current

Please use the following format when citing this chapter:

Abdullah, Mohd Syazwan, Paige, Richard, Benest, Ian, Kimble, Chris, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 70–77

role as an enabler for knowledge management (KM) initiatives has led to greater appreciation of this technology [4, 5]. It has matured from a non-scalable technology to one that can be adopted for managing the knowledge used in demanding commercial applications; it is a tool that is widely accepted by industry [6, 7]. Because it is a maturing technology, the Object Management Group (OMG), which governs object-oriented software modelling standards, has started a standardisation process [8] for knowledge-based engineering services and production rule representation (PRR).

This paper is organized as follows: Section 2 describes and discusses knowledge modelling issues in designing KBS. Section 3 explains the rationale for having an extension to UML for modelling knowledge. Section 4 gives an overview of the UML extension mechanism, while section 5 presents the knowledge modelling profile. Section 6 describes a case study that illustrates how the profile can be used to develop a KBS. Section 7 concludes and indicates future directions for the work.

2 Knowledge Modelling

The *knowledge-level* principle, popularised by [9] for KE purposes, requires that knowledge be modelled at a conceptual level independent of the implementation formalism. Knowledge modelling is similar to that of conceptual modelling, which is widely used to refer to implementation-independent models in SE; both the terms are used inter-changeably in the KE domain. The *knowledge-level* principle is fundamental to the process of conceptualisation for problem solving [10] and is used in KE for the explicit representation of the real world problem that is to be solved by the proposed system [11].

While knowledge about the domain is usually addressed through the use of ontologies, the independent reasoning process is specified with Problem Solving Methods (PSM) [5]. Both ontologies and PSM provide components that are reusable across domains and tasks [12] enabling KBS to be designed, built and deployed quickly. Ontologies are formal declarative representations of the domain knowledge; that is, they are sets of objects with describable relationships [13]. Thus an ontology used for knowledge modelling defines the content-specific knowledge representation elements such as domain-dependent classes, relations, functions and object constants [14]. PSM however describe the reasoning-process (generic inference patterns) at an abstract level, which is independent of the representation formalism (e.g. rules, frames, etc.) [12]. PSM have influenced the leading KE frameworks such as Task Structures, Rôle-Limiting Methods, CommonKADS, Protégé, MIKE, VITAL and others [12]. PSM can be considered to be design patterns in KE for KBS development [3].

It is commonly agreed by researchers [15] that conceptual modelling is an important stage in any software system construction. However, both SE and KE communities have developed different modelling techniques that are now almost unrelated [16] as a result of the fundamental computational difference between them in solving the same problem [17]. As a result, although both field's ultimate goal is to build software systems, the different experiences are difficult to interchange [17].

Nevertheless, most KE modelling notations are derived from the SE field as these are better established.

3 UML Extensibility Mechanism

The OMG's Model Driven Architecture (MDA) – a model-driven engineering framework – provides integration with, and interoperability between, different models developed using standards [18] such as the UML. The growth of MDA will fuel the demand for more meta-models to cater for domain specific modelling requirements [18]. The development of a profile, by constructing a meta-model, will enable it to be integrated into the MDA space. Integration with MDA is important for this knowledge modelling language since it cannot exist in isolation. The UML is a general-purpose modelling language [18] that can be used in very different application domains. It can be enhanced to model domains that are not currently supported, by extending the modelling features of the language in a controlled and systematic fashion. The OMG [19] has defined two mechanisms for extending UML: profiles and meta-model extensions and this work adopts the current UML 2.0 standard.

4 Knowledge Modelling Profile

The work presented in this paper adopts the XMF approach [20] in designing the knowledge modelling profile as the OMG only specifies how profiles should be constituted and not how to design them. By adopting the XMF approach, the profile development is structured into well-defined stages that are easy to follow and methodologically sound. The XMF is a newly developed object-oriented meta-modelling language, and is an extension to existing standards defined by OMG. The XMF approach to creating a profile can be divided into three steps: the derivation of an abstract syntax model of the profile concepts, a description of the profile's semantics, and the presentation of the profile's concrete syntax (not discussed here) if this is different from UML diagrams. Details of the XMF approach and the profile development stages can be found in [20]. XMF was adopted in the original design [21], but since XMF is not MOF compliant, UML tools were not able to support the resulting profile.

Profiles are sometimes referred to as the “lightweight” extension mechanism of UML [22]. A profile contains a predefined set of Stereotypes, TaggedValues, Constraints, and notation icons that collectively specialize and tailor the UML to a specific domain or process. The main construct in the 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 is used for its description. 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 [23]. In the case of knowledge

modelling, the existing constructs of UML are sufficient in representing the KBS concepts.

The Profile Concept

The concepts that underpin the profile are those taken from the existing BNF definition of the CommonKADS Conceptual Modelling Language (CML) [3], providing a well-defined and well-established set of domain concepts. Most of these elements are generally those adopted in the KBS literature and are widely used for representing the concepts of KBS in the KE domain. These knowledge modelling concepts are itemised in Table 1 and the abstract syntax model of the profile is shown in Figure 1 on the next page.

Table 1. Main Knowledge Modelling Concepts

| Modelling Concept | Description |
|-------------------------|--|
| Concept (class) | Class that represents the category of things |
| FactBase/Working Memory | Collection of information/fact that will be matched against the rule |
| Inference | The lowest level of functional decomposition consisting of primitive reasoning steps |
| Transfer Function | Transfers information between the reasoning agent and external entities (system, user) |
| Task | Defines the reasoning function |
| Task Method | Describes the realization of the task through subfunction decomposition |
| Static Knowledge Role | Specifies the collection of domain knowledge that is used to make the inference |
| Dynamic Knowledge Role | Run-time inputs and outputs of inferences |
| Rule Type | Categorization and specification of knowledge |
| Rule | Expressions that involve an attribute value of a concept |
| Knowledge Base | Collection of data stores that contains instances of domain knowledge types |

Model Extension

The knowledge modelling profile concept extends the existing meta-models of UML by defining the profile's abstract syntax. There are three places where the profile can be viewed as an extension to UML and these are: Class, Named Element and Constraints, all of which are central to the core UML meta-model and are also found in UML. The knowledge modelling concept class enables the concept to inherit all the features of a class and allows it to specify attributes and constraints on the attribute values. Other concepts such as inference, task, task method, dynamic role, static role, and the transfer function are also viewed as a subclass of the UML Class and inherit its features. This allows operations relating to objects to be expressed through the static role and at the same time allows these elements to specify attributes. Such operations are: an execute inference call from the task method, the execution of the inference process and access to knowledge in the knowledge base. Knowledge base is a subclass of the UML class. It has a 'content' slot for specific tables. 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 profile's tuple concept is also extended from Class. Constraint class is a subclass of the UML 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. Rule Type is subclassed from the UML Named Element, which allows rules to be identified using a name. All the associations described in the profile are extensions of the UML association class. However, they are not shown in the profile, as it would clutter the diagram.

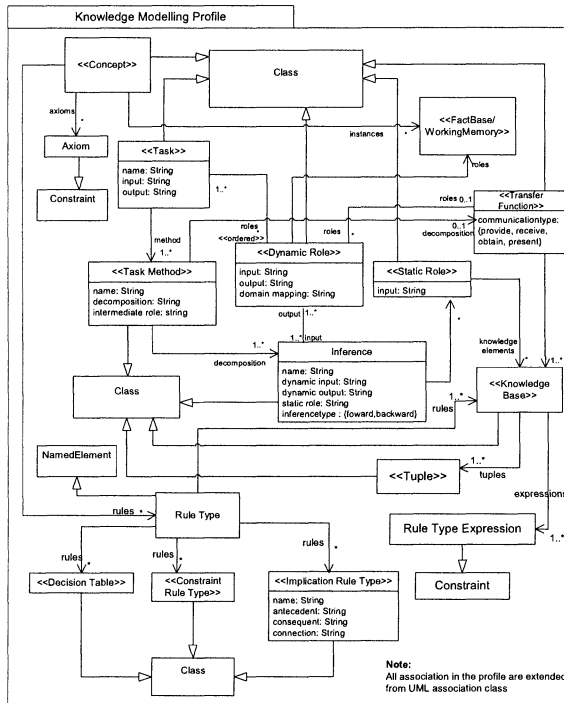


Fig. 1. Knowledge Modeling Profile

6 Case Study - Clinical Practice Guideline Recommendations

The Clinical Practice Guideline (CPG) Recommendations are guidelines that contain statements, which are graded according to the following three levels of evidence: (I) generally consistent findings in a majority of multiply acceptable studies; (II) either based on a single acceptable study, or weak or inconsistent findings in multiply acceptable studies; (III) limited scientific evidence that does not meet all the criteria of acceptable studies of good quality. The guideline contains recommendations for assessment of leg ulcers, management of venous leg ulcers, cleansing, removal of

debris, dressing and contact sensitivity, education and training, and quality assurance categories. A knowledge-based system for educational purposes was designed and developed based on the listing of recommendations in terms of: (a) evidence strength; (b) evidence strength and category; (c) category alone. Figure 2 shows how the profile was used to represent part of the CPG case study.

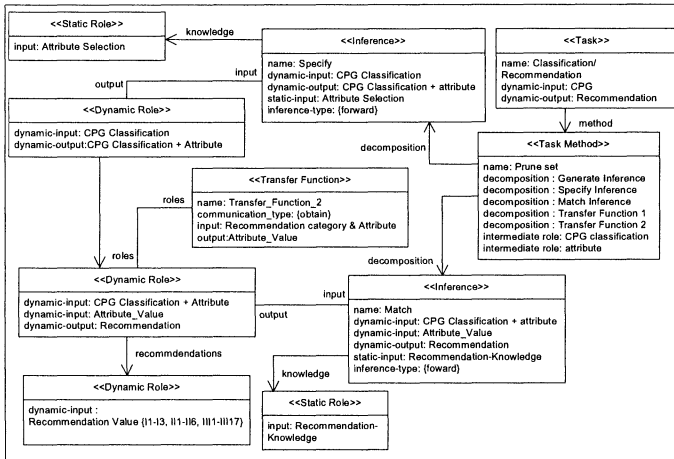


Fig. 2. CPG case study model

The profile here only concentrates on showing the task of making recommendations (*considered as a classification task-type*) based on the user-selected criteria. The task is executed by the method “prune set” which is carried out by several inferences and intermediate roles. For the matching process to provide recommendations, different sets of rules are used depending on the criteria selected by the user. To arrive at a recommendation, the inference would need to access the knowledge or rules from the knowledge base, which will be provided by the static role. The corresponding facts are gathered from the dynamic role that takes user inputs and the CPG recommendations from the factbase.

The case study was implemented as a prototype system in JESS, which is based on the popular CLIPS program [24]. Because of the declarative nature of expert system shells, the concepts of the profile cannot be directly matched to a JESS metamodel entirely. This is due to the fact that some of the JESS elements could not be directly mapped to the profile and future work will improve these mappings. However, the knowledge modelling profile was very useful in understanding the KBS requirements for the CPG recommendations. Given below is a sample Jess program that lists recommendations based on the following evidence strength (in the actual recommendation each has a brief explanation rather than the ID shown as I1, I12, III4 and so on):

```
If evidence.strength = 1 Then Recommendation = {I1, I2, I3, I4}
```

If evidence.strength = II Then Recommendation = {II1, II2, II3, II4, II5, II6}
 If evidence.strength = III Then Recommendation = {III1, III2, III3, III4, III5,.... to III19}

Sample program code of Jess for the CPG Recommendations

```
( defrule strength-I
  ( user (strength ?i&:(= ?i 1)))
  => assert (recommendation I1 , I2 , I3 , I4)
            (explanation "Strength equals 1"))
( defrule f strength-II
  ( user (strength ?i&:(= ?i 2)))
  => (assert (recommendation III1 , II2 , II3 , II4 ,II5 ,II6)
            (explanation "Strength equals 2"))
( defrule strength-III
  ( user (strength ?i&:(= ?i 3)))
  => (assert (recommendation III1 , III2 , III3 , III4 ,III5 ,III6
            ,III7, III8,III9,III10,III11,III12,III13,III14,III15,III16,III17)
            (explanation "Strength equals 3 "))))
```

7 Conclusions and Future Work

KBS development is similar to that of SE where they both rely on conceptual modelling of the problem domain to provide an orientation on how the system should address the problem. UML has been adopted in the SE domain as a standard for modelling, but there is still no consensus in the field of KE. This paper describes an extension to UML using the profile mechanism for knowledge modelling that allows KBS to be designed using an object-oriented approach. The profile has been successfully tested on several case studies involving KBS design and development from scratch and in re-engineering an existing KBS. The future work in this area involves improving the mapping of the profile to a specific inference engine meta-model. It is expected that Jess in the first instance, will help assess not only the utility of the profile for building realistic KBS, but also the utility of XMF for capturing the meta-models and building the transformations.

References

1. Giarratano, J.C. and G.D. Riley, *Expert Systems: Principles and Programming*. 4 ed. 2004, Boston, Massachusetts: Course Technology - Thomson.
2. Studer, R., Benjamins, R.V. and Fensel, D. *Knowledge Engineering: Principles and Methods*. Data & Knowledge Engineering, 1998. **25**: p. 161-197.
3. Schreiber, G., Akkermans, H., Anjewierden, A., deHoog, R., Shadbolt, N. de Velde, W.V., and Wielinga, B. *Knowledge Engineering and Management: The CommonKADS Methodology*. 1999, Massachusetts: MIT Press.
4. Ergazakis, K., Metaxiotis, K., and Psarras, I. *Knowledge Management in Enterprises: A Research Agenda*. Intelligent Systems in Accounting, Finance and Management, 2005. **13** (1): p. 17-26.
5. Studer, R., Decker, S., Fensel, D., and Staab, S. *Situation and Perspective of Knowledge Engineering*, in *Knowledge Engineering and Agent Technology*.

- IOS Series on Frontiers in Artificial Intelligence and Applications.*, J. Cuenca, et al., Editors. 2000, IOS Press: Amsterdam.
6. Liebowitz, J., *If You Are A Dog Lover, Build Expert System; If You Are A Cat Lover, Build Neural Networks*. Expert Systems With Applications, 2001. **21**: p. 63.
 7. Preece, A., *Evaluating Verification and Validation Methods in Knowledge Engineering*, in *Micro-Level Knowledge Management*, R. Roy, Editor. 2001, Morgan-Kaufman: San Francisco. p. 123-145.
 8. Tabet, S., Wagner, G., Spreeuwenberg, S., Vincent, P., Jacques, G., de Sainte Marie, C., Pellant, J., Frank, J., and Durand, J. *OMG Production Rule Representation - Context and Current Status*, in *W3C Workshop on Rule Languages for Interoperability*. 2005. Washington, D.C., USA.
 9. Newell, A., *The Knowledge Level*. Artificial Intelligence, 1982. **18**: p. 87-127.
 10. Gómez, A., Moreno, A., Pazos, J., and Sierra-Alonso, A. *Knowledge maps: An essential technique for conceptualisation*. Data & Knowledge Engineering, 2000. **33**(2): p. 169-190.
 11. Juristo, N. and A.M. Moreno, *Introductory paper: Reflections on Conceptual Modelling*. Data & Knowledge Engineering, 2000. **33**(2): p. 103-117.
 12. Gomez-Perez, A. and V.R. Benjamins. *Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods*. in *IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*. 1999. Stockholm, Sweden, p. 1-1 - 1-15.
 13. Gruber, T.R., *Toward Principles For The Design Of Ontologies Used For Knowledge Sharing*. 1993, Stanford University.
 14. Kende, R., *Knowledge Modelling in Support of Knowledge Management*. Lecture Notes in Artificial Intelligence, 2001. **2070**: p. 107-112.
 15. Naumenko, A. and A. Wegmann. *A Metamodel for the Unified Modeling Language*. in *UML 2002*. 2002. Dresden, Germany.: Springer, Berlin, p. 2-17.
 16. Cuenca, J. and M. Molina, *The Role Of Knowledge Modelling Techniques In Software Development: A General Approach Based On A Knowledge Management Tool*. Int. Journal of Human-Computer Studies, 2000. **52**: p. 385-421.
 17. Juristo, N., *Guest editor'*. Knowledge Based System, 1998. **11**(2): p. 77-85.
 18. Muller, P.-A., Studer, P., and Bezivin, J. *Platform Independent Web Application Modeling*. In *The Sixth International Conference On The Unified Modeling Language (UML 2003)*. 2003: Springer, p. 220-233.
 19. OMG, *Unified Modeling Language specification (version 1.4)*. 2001.
 20. Clark, T., Evans, A., Sammut, P., and Williams, J. *Metamodelling for Model-Driven Development (draft): To be published*. <http://albinixactium.com>. 2005.
 21. Abdullah, M.S., Evans, A., Paige, R., Benest, I., and Kimble, C. *Modelling Knowledge Based Systems Using the eXecutable Modelling Framework (XMF)*, in *Cybernetic and Intelligent Systems (CIS) 2004*. IEEE Press, p. 1053-1059.
 22. OMG, *Requirements for UML Profile*. 1999, Object Management Group: Framingham, MA, U.S.A. p. 8.
 23. Perez-Martinez, J.E., *Heavyweight Extensions To The UML Metamodel To Describe The C3 Architectural Style*. ACM SIGSOFT Software Engineering Notes, 2003. **28**(3), p. 5-5.
 24. Friedman-Hill, E., *Jess in Action: Rule-Based System in Java*. 2003, Manning.