Probabilistic Metamodel Merging

Robert Lagerström[1], Moustafa Chenine[1], Pontus Johnson[1], and Ulrik Franke[1]

[1] The Royal Institute of Technology, Industrial information and control systems, Osquldas v.12, 10044 Stockholm, Sweden
{robertl, moustafac, pj101, ulrikf}@ics.kth.se

Abstract. This paper proposes the use Bayesian networks for the automatic merging of metamodels. The proposed Bayesian networks calculate the probability that a merge of two metamodel elements is suitable, thus suggesting what to merge.

Keywords: Metamodel merging, Bayesian networks.

1 Introduction

In recent years, both researchers and practitioners have discovered the emerging possibilities of using metamodels and ontologies. As a consequence, a large quantity of metamodels and ontologies now exists within all sorts of different applications. The distributed environment of metamodel and ontology development has led to large overlaps within and between metamodels. Since these diverse metamodels often describe similar aspects of systems, developers and users would gain large benefits if metamodels could easily be merged and aligned with each other. Therefore, integration and merging of metamodels and ontologies has received an increasing interest lately [1][2].

This paper proposes an approach to metamodel merging where a probabilistic inference engine is employed to evaluate candidate metamodel concepts suitable for merging. The proposed approach uses Bayesian networks to assess the probability that a merge of two elements is suitable. The Bayesian network evaluates a merge based on various syntactic and semantic characteristics of the candidate concepts, such as the similarity of names and associations to other concepts.

The concepts described in this paper all pertain to a certain type of metamodels called abstract models. Abstract models have previously been proposed as a notation for describing and analyzing enterprise systems [3]. The models represent the architectures of these systems as well as other expert knowledge and empirical observations that can be instantiated and used for analysis. An abstract model contains classes and class associations, augmented with attributes and attribute associations. A UML description of abstract models can be seen in Fig. 1.

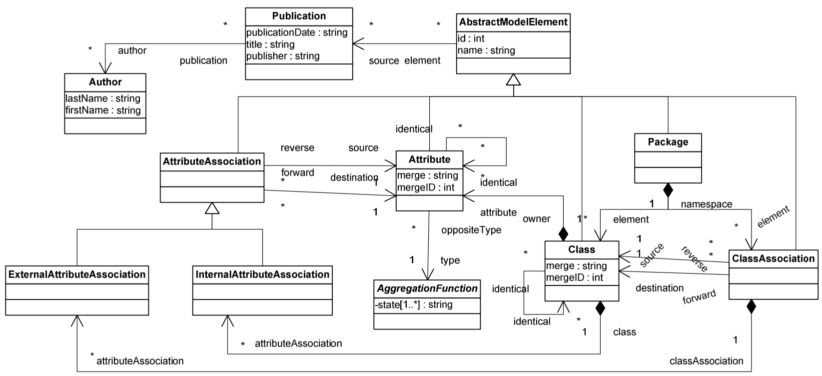

Fig. 1. UML description of abstract models.

2 Merging Metamodels using Bayesian Networks

A Bayesian network, $B=(G, P)$, is a representation of a joint probability distribution, where $G=(V, E)$ is a directed acyclic graph consisting of vertices, V, and edges, E. The vertices denote a domain of random variables $X_1,\ldots, X_n$, also called chance nodes. Each chance node, X_i, may take on a value x_i from the finite domain $Val(X_i)$. The edges denote causal dependencies between the nodes, i.e. how the nodes relate to each other. The second component, P, of the network B, describes a conditional probability distribution for each chance node, $P(X_i)$, given its parents $Pa(X_i)$ in G. More comprehensive treatment on Bayesian networks can be found in e.g. Jensen [4].

When merging abstract models, there are two concerns. Firstly, do any of the classes in the source models represent the same concepts? Secondly, when two classes have been merged, do any of the attributes in the merged class represent the same concepts? If these two concerns are correctly handled, then all associations separately holding in the source models will also be correctly transferred into the target model. Therefore, two Bayesian networks were developed; one describing class merges and one describing attribute merges, c.f. Fig. 2.

The nodes in the class merge network have the following scales: Class Merge = {Yes, No}, Class Similarity Association = {Yes, No}, Class Names = {Identical, Similar, Dissimilar}, Class References = {SamePublication, SameAuthor, DifferentAuthors}, Class Attributes = {All, Some, None}, and Class Associations = {All, Some, None}.

The nodes in the attribute merge network have the following scales: Attribute Merge = {Yes, No}, Attribute Similarity Association = {Yes, No}, Attribute Names = {Identical, Similar, Dissimilar}, Attribute References = {SamePublication, SameAuthor, DifferentAuthors}, and Attribute Associations = {All, Some, None}.

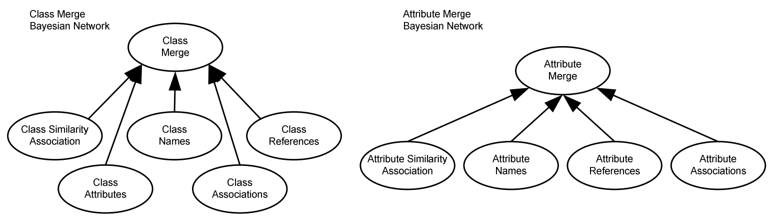

Fig. 2. Bayesian networks representing class and attribute merge.

To illustrate the application of the Class Merge network c.f. the Bayesian network screenshot from GeNIe [5] in Fig. 3.

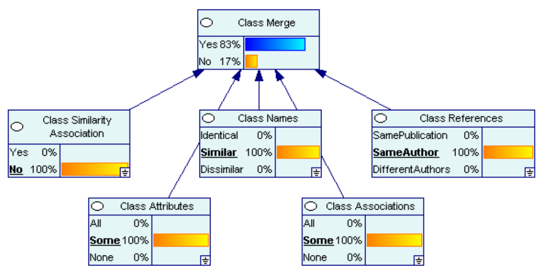

Fig. 3. The Bayesian network for class merges with example values.

Assume that a package, i.e. a set of abstract models, contains the abstract models presented in Fig. 4.

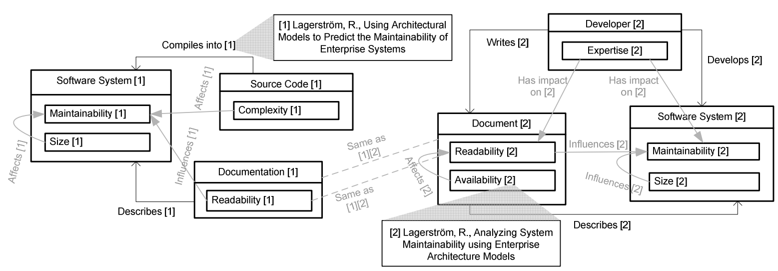

Fig. 4. Abstract models to be tested for possible merges.

The method starts by comparing the pairs of classes from the models presented in Fig. 4. Then, the classes with the highest probability are merged. In this example, the class pair *Documentation* and *Document* received the probability P = 100 %, and are therefore merged.

The next step is to compare all pairs of attributes in the merged class *Documentation*. Then, the attributes with the highest probability are merged. In this example, the pair *Readability* and *Readability* received the probability P = 100 %, and are therefore merged.

It all starts over from the beginning by comparing all pairs of classes and iterates until no classes receive probabilities over a predefined merging threshold. The resulting model in this example is presented in Fig. 5.

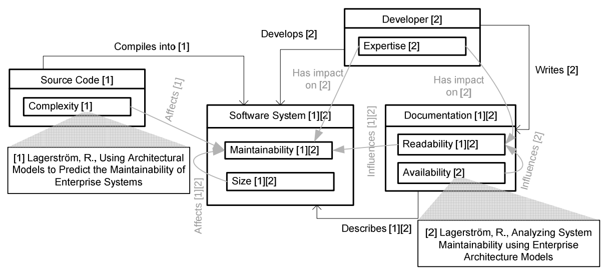

Fig. 5. The resulting abstract model after employing the proposed Bayesian networks.

4 Conclusions

This paper addresses the issue of metamodel merging, using the probabilistic framework of Bayesian networks. It was shown that Bayesian networks can be used to guide the merging of metamodels, by considering some key features of the classes and attributes at hand: basically their names, references, and associations. With this information it is possible to discern the probability that the concepts are sufficiently similar to be merged.

References

1. Noy, N., Musen, M.A.: The PROMPT suite: interactive tools for ontology merging and mapping. Internationl Journal of Human-Computer Studies, Vol. 59, No 6, p. 983-1024 (2003)
2. Doan, A., Madhaven, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In the proceedings of the 11th International Conference on World Wide Web (2002)
3. Johnson, P., Johansson, E., Sommestad, T., Ullberg, J.: A Tool for Enterprise Architecture Analysis, In Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC), Vol. 11, Annapolis USA (2007)
4. Jensen, F.: Bayesian Networks and Decision Graphs. Springer-Verlag (2001)
5. GeNIe: Decision System Laboratories, About GeNIe and SMILE, University of Pittsburgh, http://genie.sis.pitt.edu/about.html (2007)