An Ontological Framework for Adaptive Feedback to Support Students while Programming

Pedro J. Muñoz-Merino[1], Abelardo Pardo[1], Maren Scheffel[2], Katja Niemann[2], Martin Wolpers[2], Derick Leony[1], and Carlos Delgado Kloos[1]

[1] Universidad Carlos III de Madrid,
Avda. de la Universidad, 30, E-28911, Leganés (Madrid), Spain
{pedmume, abel, dleony, cdk}@it.uc3m.es
[2] Fraunhofer-Institute for Applied Information Technology FIT,
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{maren.sheffel, katja.niemann, martin.wolpers}@fit.fraunhofer.de

Abstract. This paper presents a global framework based on ontologies to generate effective feedback for students getting error messages while learning to program. The proposed framework includes several ontologies: one of the domain, one of possible mistake types and one of the causes of these mistakes. They are connected by an intermediary ontology. The feedback is adaptive and depends on contents or students profiles (derived from interaction data) and is supported by a fifth ontology. Possible compilation errors in an introductory C programming course are presented as an example of this framework.

Keywords: ontology, adaptation, feedback, education

1 Introduction

Students often run into problems while trying to solve a programming task. Making mistakes and getting error messages is part of their learning process. In those cases, they need to review course materials or look for information in Internet forums to try to find the specific information that can help them solve their problem. Searching for a solution can be time consuming and does not always imply a better learning experience. If a system can generate effective feedback that guides students to the solution, the learning process can be more effective. In addition, depending on each student, the generated feedback can be adaptive based on their previous interactions.

The use of an ontology framework that models the subject domain, the possible mistakes and their connection, can be a base for generating adaptive effective feedback. In this paper, we present an ontology framework for the generation of such adaptive feedback for students. The framework is illustrated with a specific example about the generation of adaptive feedback for students getting different compilation errors in a C programming course.

There are other systems that provide student feedback using semantic web technologies. Jovanovic et al. [1] and Jeremic et al. [2] give feedback based on parameters such as students' evaluations while feedback depending on the students' knowledge level is also possible (e.g. [3]). All three, however, are not focused on the content's relationships of the domain ontology. Moreover, Kazi et al. [4] provide feedback for a medical environment based on the appearance of several words in a free text and compare it with a domain ontology, but this work does not connect predefined mistakes with a domain ontology and does not implement adaptation. The importance of compilation errors has been addressed in [5] where different statistics about them are provided.

2 Overview of the Ontological Framework

The proposed ontological framework is composed of five ontologies: (1) A subject domain ontology. This ontology describes the different concepts to be taught in the course and includes all the concepts related to the possible errors. (2) An error ontology. This ontology describes the different errors a student can get. (3) A cause of error ontology. This ontology describes the different possible causes for each possible error. (4) An ontology that relates causes of error to the subject domain concepts. (5) An ontology that matches different possible adaptations depending on students profiles and the corresponding generated feedbacks in relationship with the subject concepts.

3 An Example in an Introductory C Programming Course

A part of the subject ontology that models part of the C programming concepts related to variable types and variable declaration can be seen in figure 1. Boxes represent classes while arrays represent inheritance or a specific property. Figure 2 shows part of the ontology that describes several compilation errors. The selected errors are the more common ones on the course, as analyzed in [6]. Figure 3 shows part of an ontology describing the different possible causes for the *Variable_undeclared* error.

An ontology that relates causes of errors with the different concepts involved in the course will enable further insights for students getting such errors. For example, a *shouldbereviewed* property can be part of this ontology to determine the different concepts that are related to a cause of error and that should be looked at again. In this way, the *extern* concept can for example be reviewed when the E*xtern_needed* error is present. Finally, an adaptation ontology must be defined to adapt the feedback to the different students. The ontology must contain information about the students' profiles that will have different feedback, the exact delivery of the feedback, and its presentation.

This adaptive feedback can be generated automatically based on the presented ontologies and the students' interactions with the system, i.e. their usage data, that is being logged as CAM (Contextualized Attention Metadata) [7]. Ontologies are suited for describing data models that may change over time as they can be adapted easily, e.g. to include a student's learning curve. The decisions about how to make the adaptation, and thus building the ontology, can be made based on the analysis of the students' previous interaction data. A detailed description about what events are logged for the C programming course and how the students' key actions are detected can be found in [6]. Based on these CAM it is possible to see which error messages the students get and how often, what they did before the compilation and how they react to the error message.

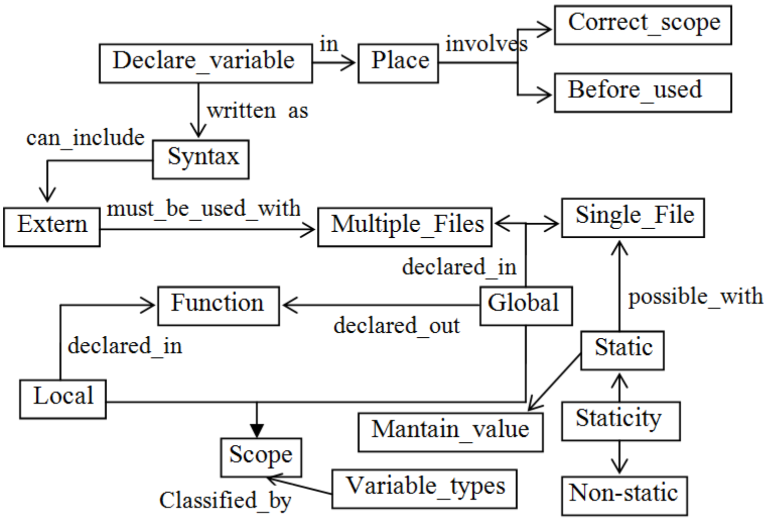

Fig. 1. Part of a subject domain ontology for C programming

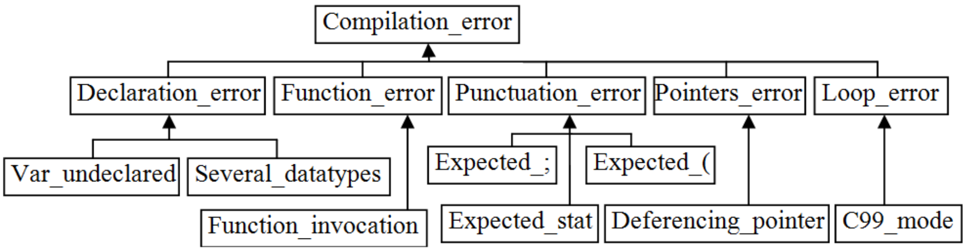

Fig. 2. Part of an ontology about students' compilation errors

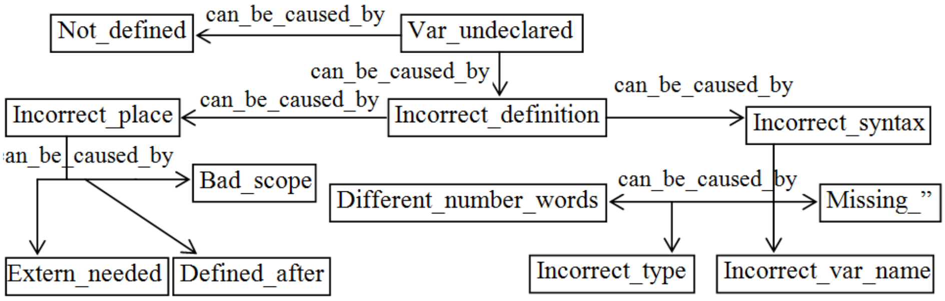

Fig. 3. Ontology about the causes of the *variable undeclared* error

The CAM data in combination with the ontologies can be used to personalize the system's feedback for each student. For example, for some students, the system can start a dialogue with them about all possible causes of the *Variable_undeclared* error (using the ontology of figure 3), or give information not only about the *Extern* concept but about those that are connected to it as well, depending on whether this is a common error for this student or not. Many other aspects, such as the probability for a certain cause of error for each student based on his previous mistakes, can be taken into account as well.

4 Conclusion

In this paper we presented an ontological framework for generating adaptive feedback for students' mistakes, illustrating it on the basis of a C programming course. The feedback is based on the subject topic information and its relationship with students' mistakes and their causes. As future work, we plan to develop the corresponding ontologies for the entire C programming course and implement a complete system that allows adaptation based on this ontological framework. The framework can be easily generalized for other learning domains in which there are different topics, students' errors, and their related causes. The same five ontologies presented in section 2 would have to be defined for new learning domains.

References

1. Jovanovic, J., Gasevic, D., Brooks, C., Devedzic, V., Hatala, M., Eap, T., Richards, G., Using Semantic Web Technologies for the Analysis of Learning Content, IEEE Internet Computing, vol. 11, no. 5, 45—53 (2007)
2. Jeremic, Z., Jovanovic, J., Gasevic, D.: Semantic Web Technologies for the Integration of Learning Tools and Context-Aware Educational Services. In: Bernstein, A. et al (eds.) ISWC-2009. LNCS, vol. 5823, pp. 860--875. Springer, Heidelberg (2009)
3. Muñoz-Merino, P.J., Delgado Kloos, C.: An Architecture for Combining Semantic Web Techniques with Intelligent Tutoring Systems. In: Woolf, B.P, Aımeur, E., Nkambou, R., Lajoie, S.P.(eds.), ITS 2008. LNCS, vol. 5091, pp. 540–550. Springer, Heidelberg (2008)
4. Kazi H, Haddawy P, Suebnukarn S.: Leveraging a domain ontology to increase the quality of feedback in an intelligent tutoring system. In: Proceedings of the 10th International Conf. on Intelligent Tutoring Systems; 2010 Jun 14-18; Pittsburgh, USA. New York: Springer.
5. Blikstein, P.: Using learning analytics to assess students' behavior in open-ended programming tasks. 1st International Conf. on Learning Analytics and Knowledge (2011)
6. Scheffel, M., Niemann, K., Pardo, A., Leony, D., Friedrich, M., Schmidt, K., Wolpers, M., Delgado Kloos, C.: Usage Pattern Recognition in Student Activities. In: Delgado Kloos et al (eds.) ECTEL-2011. LNCS, To be published, Springer Heidelberg (2011)
7. Schmitz, H.-C., Kirschenmann, U., Niemann, K., Wolpers, M.: Contextualized Attention Metadata. In: Roda, C. (ed.): Human Attention in Digital Environments.Cambridge University Press (2011)