DEVELOPMENT OF EDUCATIONAL ONTOLOGY FOR C-PROGRAMMING

Sergey Sosnovsky, Tatiana Gavrilova

Abstract: Development of educational ontologies is a step towards creation of sharable and reusable adaptive educational systems. Ontology as a conceptual courseware structure may work as a mind tool for effective teaching and as a visual navigation interface to the learning objects. The paper discusses an approach to the practical ontology development and presents the designed ontology for teaching/learning C programming.

Keywords: Ontology Design, Knowledge, Educational Ontology, C Programming, Ontology Visualization.

ACM Classification Keywords: K.3.2 Computer and Information Science Education; D.2.11 Software Architectures: Languages (e.g., description, interconnection, definition)

Introduction

The intensity of modern technology development makes exceptional demands of the process of education. The speed of the knowledge deterioration increases steadily. According to the experts' reports the "half-value period" of a modern specialist is from 3 to 5 years. The number and the diversity of students grow up. Programs for life-long and distance education appear. Students differ in the learning goals, background, cultural aspects, which increase not only the volume of knowledge but also the ways, how it is taught. Different subjective views on the same knowledge for different groups of students may exist.

In these conditions a teacher as the main knowledge provider in the framework of modern education is overloaded. It becomes impossible for him/her alone to preserve the high quality of the knowledge taught. The solution is now obvious, knowledge should be created in the reusable and sharable form, in a way that once developed it could be used by anyone as a whole or partially.

Even greater need in making knowledge shareable and reusable is declared in the field of educational systems development. The knowledge base of a modern computer-based educational system should support the import and export of the knowledge in a standard format using standard protocols. Even for the domains where knowledge is pretty stable, like C Programming, such a perspective lead to the exceptional opportunity of using different systems from different developers in a common framework. The first step in this way is making the process of engineering of educational knowledge ontology-based.

The term of ontology emerged and became popular (even fashionable) during the last one and half decades. Though very yang it is yet a quite mature area of research. Ontological engineering inherits the practical and theoretical results of knowledge engineering, which has about forty yeas of history. According to one of the definitions "ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base" [Swartout et al., 1997]. It "defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary" [Neches et al., 1991].

In this paper we present the stepwise approach to ontology engineering and describe the experience of ontology developing for C-Programming. Developed knowledge structure is not just the hierarchy of the C language standard. It represents the application ontology designed for the purpose of education and accumulates the authors' experience of teaching several C-based programming courses. Next section gives details of the proposed algorithm for ontology development as well as a set of recommendations, which may be helpful in building a "beautiful ontology". Then in the section 3 we describe our domain, the motivation for the work presented here and, finally, the developed educational ontology for C programming. The summary and future work discussion conclude the paper in the section 4.

Stepwise Ontology Design

Generalizing our experience in developing different teaching ontologies for e-learning in the field of artificial intelligence and neurolinguistics [Gavrilova et al., 2004(a); Gavrilova, Voinov, 1998; Gavrilova et al., 1999; Gavrilova, 2003; Gavrilova, 2004(b)] we propose a 5-step algorithm that may be helpful for visual ontology design.

We put stress on visual representation as a powerful mind tool [Jonassen, 1996] in structuring process. Visual form influences both analyzing and synthesizing procedures in ontology development process.

Concise Algorithm for Ontology Design:

1. Glossary development: gather all the information relevant to described domain, select and verbalize all essential objects and concepts.
2. Laddering: define main levels of abstraction and define type of ontology (taxonomy, partonomy, genealogy, etc.). Reveal hierarchies among these concepts and represent them visually on defined levels.
3. Disintegration: try to detail “big” concepts into a set of “smaller” ones via top-down strategy.
4. Categorization: group similar concepts and create meta-concepts to generalize the groups via bottom-up structuring strategy.
5. Refinement: update the visual structure and exclude excessiveness, synonymy, and contradictions. Try to create beautiful ontology.

Some Precepts to Create Beautiful Ontology:

Conceptual balance (Harmony). It is a challenge to formulate the idea of well-balanced tree, but some tips may be helpful:

- One-level concepts should be linked with a “parent” concept by one type of relationship (is-a, has part, etc).
- The depth of the branches should be more or less equal (±2 nodes).
- The general outlay should be symmetrical.
- Try to avoid cross-links.

Clarity:

- Minimal number of concepts is the best tip according Ockham’s razor principle proposed by William of Ockham in the fourteenth century: "Pluralitas non est ponenda sine neccesitate", which translates as "entities should not be multiplied unnecessarily". The maximal number of branches and the number of levels should follow Miller’s number (7±2) [Miller, 1956].
- The type of relationship should be understandable if the name of relationship is missing.

C programming Ontology

Domain Description

During a number of years, we have been teaching C-based programming courses to undergraduate students of the School of Information Sciences at the University of Pittsburgh and artificial intelligence disciplines in Saint-Petersburg State Polytechnic University. Several adaptive computer-based systems have been developed for serving such learning activities as parameterized quizzes, interactive examples and social navigation [Brusilovsky et al., 2004(a); Brusilovsky et al., 2004(b); Brusilovsky et al., 2004(c)].

The natural development of such tools is an evolvement towards the distributed web-based architecture where applications share the common students’ profiles (student model) and the ontology of the domain (domain model). Some progress in this direction has been made [Brusilovsky, 2004]. Ontology of the domain as a framework for common knowledge base would allow our applications to “speak the same language”. Moreover applications from side developers can share our knowledge base and become the part of the architecture.

Another motivation to build the ontology of C programming is connected with the attempts to create more meaningful and effective teaching strategies as there is no predefined way to teach C. Different textbooks and different instructors (even when using the same textbook) may introduce C concepts, combine them into lectures and explain them one on the basis of another in very different orders. One teacher may believe that it is better to teach “while” before “if-else”, when another thinks visa versa. Not only the order of teaching concepts, but also the emphasis instructors' place on the different parts of the course and didactic paradigms they use could be different. Students may be required to learn first the structure of C program in details, or may be given “Hello World” example and immediately asked to code the similar program; the programming patterns for some courses (like algorithm design or data structures) might have much higher importance then for the introductory C course, etc.

The advantage of the ontology is that it attempts to unify different views on the domain. Selected parts of the ontology could be used for different sections of the course. The order, in which a teacher presents the material, is up to him/her while the basic hierarchical link structure is not violated.

Development of Educational C Ontology

We used the algorithm described above to create the ontology for teaching/learning C programming. Figure 1 demonstrates four top levels of the developed ontology. Lower levels trivially expand the hierarchy therefore we have hidden them. The main type of relationships is “has part”. That is why this is partonomy.

Naturally, the upper level central node is the C programming; second level represents the abstract meta-concepts, which combine more concrete entities. The major difficulties were to compose and to name these intermediate concepts. Figure 1 presents the fourth “release” of the design drafts.

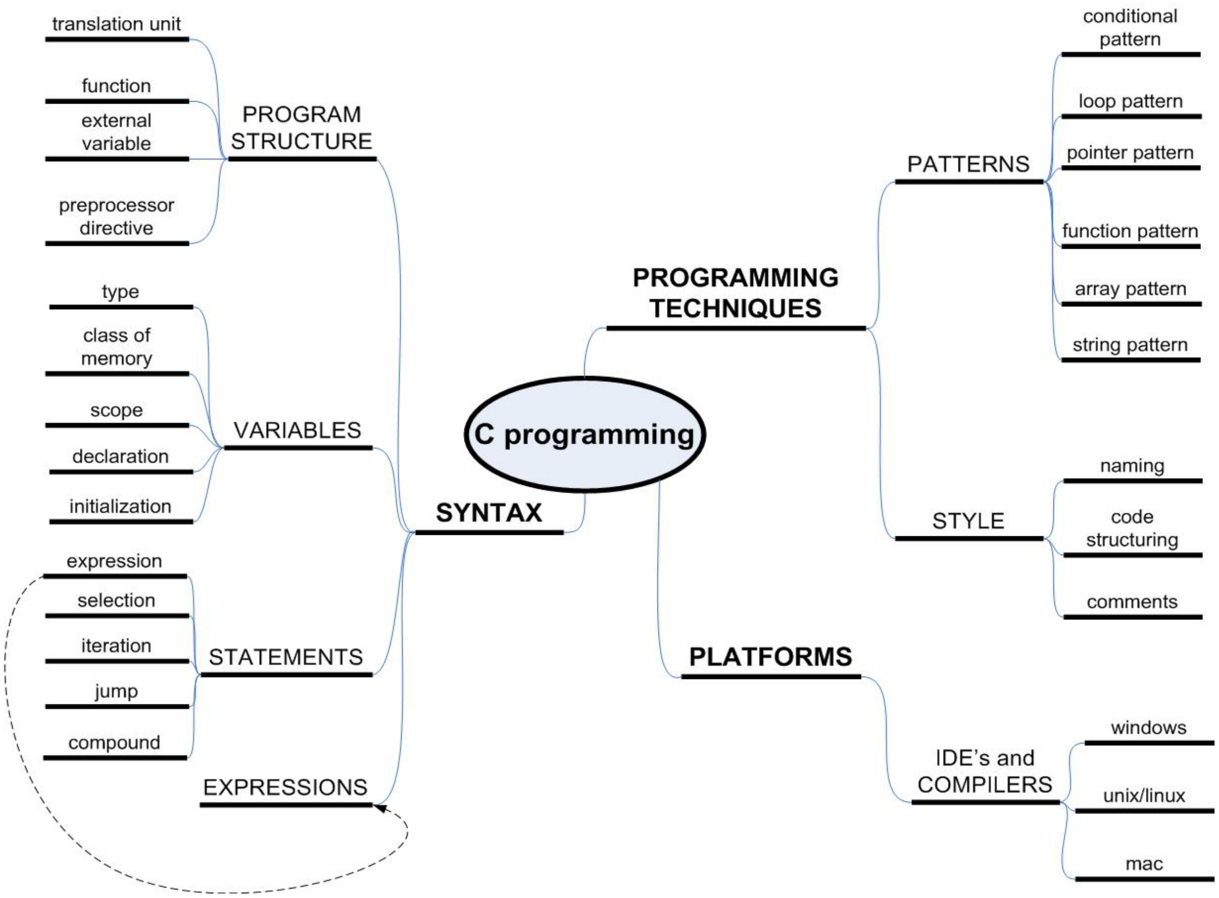

Figure 1. Top Levels of the Educaional Ontology of C Programming.

The concepts of the third level are the main parts of the material that students study. They combine very separate areas of C programming knowledge, where an emphasis needs to be placed. The entities of the third level in their turn are subdivided into programming topics as sub-concepts. These topics for some branches are already concrete enough to be the theme of the lecture or the section in a syllabus. However, as we mentioned before, this level is far from being the last one.

As we told already, the purpose of this ontology is use in education; therefore it attempts to reflect not simply the standard of the language, but all necessary knowledge that students need to learn, including helpful programming techniques and compiler usage. It does not mean that we necessary provide a system, which teaches students for example to work with compilers. However, this branch in the ontology let us to use it, say, for navigating them in entering the online compiler tutorials.

The association link between expression sub kind of statements and expressions as a section of the third level though adds some irregularity to our ontology, is needed because of the educational purpose. In C standard expression statement is a kind of statement. That is why expression is a sub concept of statements. However, from the point of view of teaching/learning C, expressions are totally different area then any other.

The expressive power of the ontology allows us to encode different relations between concepts (by concepts we mean here entities in the hierarchy, but not the knowledge elements of the lowest level). Besides the link topology representing the whole-part hierarchical relationships, the order of concepts in a group represent the interconnection between them and preferred sequence of their study, though the last one is rather a recommendation then a directive.

As the main source for knowledge elicitation on the stage of glossary development we used [Kernighan, Ritchie, 1988].

Summary and Future Work

The paper has proposed the stepwise algorithm for ontology development and implementation of this algorithm for creation of the educational ontology for C programming. Created ontology does not simply replicate the hierarchical structure of the C language standard, but reflects the authors' vision on what is important in studying C and accumulates their experience of teaching C-related programming courses. Following three subsections discuss the directions of the future research.

Ontology-based Common Domain Model

The developed ontology is going to be used for several computer-based educational systems as a domain knowledge representation model. The C programming as a domain for adaptive educational systems is "lucky" to be formal enough for its concepts possess grammatical structure. This is especially true for the sub kinds of the SYNTAX meta-concept (see figure 1). Traditionally, the extraction of grammatically meaningful structures from textual content and the determination of concepts on that basis is a task for the special class of programs called parsers. In our case, we have developed the parsing component with the help of the well-known UNIX utilities: lex and yacc. This component processed source code of a C program and generates a list of concepts used in the program [Sosnovsky et al., 2004]. This tool will help us to automatically index the content of our adaptive systems in terms of the concepts of the developed ontology. This leads us to the exceptional opportunities of implementing mutual adaptation across different educational application. As a result, the possible set of instructional strategies increases, since on every step instructional treatments from more applications are available.

Ontology Visualization

As we already mentioned above the ontology is not just a technical instrument but a powerful mind tool also. Ontology visualization and creation of a student interface for an educational system is one of the authors' primary goals. The hierarchical structure of the ontology makes it natural to create a navigable hypermedia interface on its basis. In 1995 Gaines and Shaw created the WebMap – system integrating concept maps (which can be to some extend considered as an ontology visualization technique) with WWW, making a first step in this direction [Gaines, Shaw, 1995]. It seems very natural to use hypermedia as an implementation framework for ontology; hence we can use different methods of hypermedia adaptation, which are well developed now [Brusilovsky, 2001].

Ontology Evaluation

One more direction of future research is the evaluation of the developed ontology, from both perspectives: as a knowledge base framework and as an interface framework. From the first point of view, we can evaluate its structural consistency as a domain knowledge representation mechanism. Also, the quality of defined concepts as assessment units might be evaluated.

From the second point of view, the quality of ontology-based interface is to be evaluated on the subjective and objective levels. Subjective evaluation could be done on the basis of questionnaires filled by students at the end of the course. To evaluate it objectively we are going to perform the statistical analysis of logs of students' work with the system to find: first, how does work with the system correlates with course performance, second, how reasonable student use the interface, i.e. do they follow our hints and suggestions, and third, if they do, how do they benefit from it, how reasonably the system adapt its behavior to the specific student.

Acknowledgements

The work reported in this paper is supported by NSF grant # 7525 *Individualized Exercises for Assessment and Self-Assessment of Programming Knowledge* and by grant 04-01-00466 RFBR.

Bibliography

[1] [Swartout et al., 1997] Swartout, B., Patil, R., Knight, K., Russ, T. Toward Distributed Use of Large-Scale Ontologies, Ontological Engineering. AAAI-97 Spring Symposium Series, 1997, 138-148.

[2] [Neches et al., 1991] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., Swartout, W. Enabling Technology for Knowledge Sharing. AI Magazine. Winter 1991, 36-56.

[3] [Gavrilova et al., 2004(a)] Gavrilova, T., Guian, F., Koshy, M. Ontological Tower of Babel. In proceedings of Second International Conference on Knowledge Economy and Development of Science, KEST 2004, Beijing, Tsinghua. University Press, 2004, 101-106.

[4] [Gavrilova, Voinov, 1998] Gavrilova, T., Voinov, A. Work in Progress: Visual Specification of Knowledge Bases. In A.P. del Pobil, J. Mira, M. Ali (Eds.) Lecture Notes in Artificial Intelligence 1416 "Tasks and Methods in Applied Artificial Intelligence", Springer, 1998, 717-726.

[5] [Gavrilova et al., 1999] Gavrilova, T., Voinov, A., Vasilyeva, E. Visual Knowledge Engineering as a Cognitive Tool. In Proceedings of International Conference on Artificial and Natural Networks IWANN'99, Benicassim, Spain, 1999, 123-128.

[6] [Gavrilova, 2003] Gavrilova, T. Teaching via Using Ontological Engineering. In Proceedings of XI International Conference "Powerful ICT for Teaching and Learning" PEG-2003, St.Petersburg, Russia, 2003, 23-26.

[7] [Gavrilova et al., 2004](b) Gavrilova, T., Kurochkin, M., Veremiev, V. Teaching Strategies and Ontologies for E-learning Information Theories and Applications, vol.11, N1, 2004, 61-65.

[8] [Jonassen, 1996] Jonassen, D. *Computers in the Classroom: Mindtools for Critical Thinking*, Englewood Cliffs, NJ: Prentice Hall, 1996.

[9] [Miller, 1956] Miller, G. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, 1956, vol. 63, pp. 81-97.

[10] [Brusilovsky et al., 2004(a)] Brusilovsky P., Sosnovsky S., Shcherbinina O. QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. In Janice Nall and Robby Robson (eds.) Proceedings of E-Learn 2004. Washington, DC, USA: AACE, 2004, 1806-1813.

[11] [Brusilovsky et al., 2004(b)] Brusilovsky P., Sosnovsky S., Yudelson M., An Adaptive E-Learning Service for Accessing Interactive Examples. In Janice Nall & Robby Robson (eds.) Proceedings of E-Learn 2004. Washington, DC, USA: AACE, 2004, 2556-2561.

[12] [Brusilovsky et al., 2004(c)] Brusilovsky, P., Chavan, G., Farzan, R. Social Adaptive Navigation Support for Open Corpus Electronic Textbooks. In: P.De Bra (ed.) Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH'2004), Eindhoven, The Netherlands, 2004.

[13] [Brusilovsky, 2004] Brusilovsky, P. A component-based distributed architecture for adaptive Web-based education. In: U. Hoppe, F. Vardejo and J. Kay (eds.) Proceedings of International Conference on Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies (AI-ED'2004), Sydney, Australia, July 20-24, 2004. Amsterdam: OIS Press, 2004, 386-388.

[14] [Kernighan, Ritchie, 1988] Kernighan, B., Ritchie, D. C Programming Language (2nd Edition). Prentice Hall PTR; 2 edition: 1988.

[15] [Sosnovsky et al., 2004] Sosnovsky S., Brusilovsky P., Yudelson M. Supporting Adaptive Hypermedia Authors with Automated Content Indexing. In Lora Aroyo and Carlo Tasso (eds.) Proceedings of AH2004 Workshops: Part II /Second Workshop on Authoring Adaptive and Adaptable Educational Hypermedia/, Eindhoven, The Netherlands, 2004, 380-389.

[16] [Gaines, Shaw, 1995] Gaines, B. R. and Shaw, M. L. G. Concept maps as hypermedia components. International Journal of HumanComputer Studies, 43(3), 1995, pp. 323--361.

[17] [Brusilovsky, 2001] Brusilovsky, P. Adaptive hypermedia. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 2001, 87-110.

Author's Information

Sergey Sosnovsky – University of Pittsburgh; 135, North Bellefield Street; Pittsburgh, PA, 15217, USA phone: +1 (412) 624-5513; e-mail: sas15@pitt.edu

Tatiana Gavrilova – Saint-Petersburg State Polytechnic University; Intelligent Computer Technologies Dept.; Politechnicheskaya, 29; Saint Petersburg - 195251, Russia; phone: + 7 (812) 550-40-73; e-mail: gavr@csa.ru

DEVELOPMENT OF PROCEDURES OF RECOGNITION OF OBJECTS WITH USAGE MULTISENSOR ONTOLOGY CONTROLLED INSTRUMENTAL COMPLEX

Alexander Palagin, Victor Peretyatko

***Abstract**: the ontological approach to structuring knowledge and the description of data domain of knowledge is considered. It is described tool ontology-controlled complex for research and developments of sensor systems. Some approaches to solution most frequently meeting tasks are considered for creation of the recognition procedures.*

***Keywords**: the tool complex, methods of recognition, ontology.*

***ACM Classification Keywords**: I.4.8 Scene Analysis – Object recognition; I.2.9 Robotics – Sensors*

Introduction

One of the ambitious purposes of the world civilization is construction of the knowledge-oriented society. In computer science, a main priority direction thus is intellectualization of computer resources and technologies, in particular creation knowledge-oriented ontology controlled intelligence systems for various assignments. Information technologies on their basis are composing components of all high technologies. Except for usage in spheres of socioeconomic activity (the most difficult) spheres of research and development activity which result are objects of new knowledge, engineering, high technologies are rather important. The majority of these applications of intelligent systems is related to the problem solving, recognition (identifications), the diversification of their settings and implementation which are extremely various. The present material is devoted to usage of the