Students Knowledge Base

Bálint FARKAS, Michal HOLUB, Juraj KOLLÁR, Vojtech VILLARIS, Martin VIRIK*

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
fiit-tp9@googlegroups.com

Our faculty collects various information reflecting abilities of its students like subjects they study or topics of their theses. When the school receives an offer for cooperation on some research or the teacher needs specific students for a project, he faces a problem of finding suitable candidates. The goal of our project is to automate the extraction of student's knowledge from his abilities and skills. To fulfil this goal we decided to build a system that will create and maintain a students' knowledge base.

We developed a method for deducing student's knowledge from information about his abilities and skills. This method works with the model of the user and weighting of his skills inspired by [2]. The user model will contain information provided by the user himself by filling out an electronic form as well as information gathered automatically from the Academic Information System (AIS), which is a system used on the whole university. Our solution could therefore be extended to include students not only from our faculty. Currently teachers evaluate students by giving them grades for subjects they study. We will collect the grades automatically from AIS; therefore no further evaluation from teachers will be needed.

Information in the user model is in the form of facts from which the knowledge is subsequently extracted. We proposed eight basic facts like school subject, training, project, publication, etc. This list could be further extended if needed. Facts are linked to knowledge through keywords. This link has its weight reflecting the importance of the keyword to knowledge and fact which it connects. This aspect of our method tells us whether a student has particular knowledge or not.

The second aspect of our method tells us how to compare students with the same knowledge. Each fact defines its own attributes. Some of them are informative (like the ID of the school subject) but some of them also influence the amount of knowledge the student has (like the number of credits awarded for this subject). In our method we assign each fact a formula which computes the degree to which the fact contributes to the knowledge (e.g. number of credits divided by the grade).

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Valéria Šimáková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

In our work we propose a system in which the user searching for particular students defines the knowledge they should have. The user defines keywords composing this knowledge and their percentage weight. Our system then searches through each student's facts for these keywords and when it finds a match it computes the output for this fact. The output is then added to the total amount of knowledge the student has. System subsequently shows sorted list of students with this knowledge.

For implementation of our system we use modern technologies based on JEE. Interesting is the way of how we record facts about students. Instead of enumerating a fixed number of facts during design phase, we allow the user with special role to define which facts with which attributes should our system collect. User can do this at run-time without the need for modifying the source code or database schema. This is done by the concept of abstraction of types and attributes [1]. It relies on defining a meta-level which represents the type of a fact and attributes of this fact. Particular facts and values of their attributes then represent instances of these meta-level entities. Thanks to this concept the user can define virtually any type of fact.

We have also developed special visual components for open-source Java framework Tapestry [3] which we use for presentation layer. These components can display defined facts intelligently, so that for example string attribute automatically renders as text field, date attribute renders as date picker, Boolean attribute as checkbox, etc. This also automatically executes validation, e.g. entering letters for number attribute is not allowed.

In the future work we would like to focus on connecting our system with AIS so that we could obtain real data about students. We would like to evaluate our system on sample group of students who will provide real facts. We will then let some teachers to define knowledge and search among these students.

Main contribution of our approach is in weighted connection of facts to knowledge and in possibility of dynamic definition of new fact types. The results of prototyping show that our method will work and that we are able to implement the type abstraction as described. This gives our system a high degree of configurability.

Acknowledgement: This work was partially supported by the Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava.

References

[1] Šešera, Ľ., Mičovský, A., Červeň, J.: *Architektúra softvérových systémov: Analytické a dátové vzory*. Vydavateľstvo STU v Bratislave, (2000). 179 p. (in Slovak).

[2] Šimún, M., Andrejko, A., Bieliková, M.: Inicializácia a aktualizácia modelu používateľa pri hľadaní pracovných ponúk na webe. In Mikulecký, P., Dvorský, J., Krátký, M., eds.: *Znalosti 2007*. Ostrava, (2007), pp. 109-120. (in Slovak).

[3] Tong, K. I.: *Enjoying Web Development with Tapestry*. 3rd edition. TipTec Development, (2006).