Taking the drudgery out of software development

November 24 2009

(PhysOrg.com) -- Software developers will no longer have to reinvent the wheel when writing new programs and applications thanks to a clever new set of tools and a central repository of 'building blocks'.

Traditionally, developers have had to write software programs from scratch, whether or not something similar has been done in the past for a different application or circumstance.

This has meant skilled people spending a lot of time doing repetitive, boring work before being able to turn out new software for a specific requirement.

But a much more efficient way of doing things has been demonstrated by European researchers who have developed an automated way of

searching a central software repository to extract software 'artefacts' from existing systems for use in new systems.

The EU-funded ReDSeeDS project was established with the aim of initially researching the requirements for the components of an automated searchable repository, and then producing a software toolkit to cope with the requirements.

The researchers had to come up with a requirement specification language, which allows developers to use a single user interface to frame their queries; query technologies which automate the query process once the requirements have been entered; and a repository technology which searches the repository and provides answers to queries.

Automation is the key

According to project coordinator Michal Smialek there have been attempts made in the past to build standards for the type of software repository envisaged and then developed by the project. But the problem was they were not automated and so a lot of work was still involved.

"The big difference with our platform is that it allows you to simply sketch out the requirements of your proposed new system and then these are automatically compared with the requirements and capabilities of existing systems. The results are displayed to you with the differences and similarities between the old and new systems highlighted," he says.

This allows a developer to pick and choose what artefacts he can take from existing systems and slot them into the new system.

Of blueprints and codes

"In this context, by artefact we mean a software artefact which has been constructed on a computer by a software developer," Smialek notes. "This can be any kind of model or document or program which is the result of a software project," he says.

"In a project, you may produce several artefacts which are design blueprints and then an artefact which is the code that tells the system how to work. The final program is also an artefact which is served by the other artefacts - that is the design and the code."

So when it comes to matching up the requirements of a new system with what has been done in previous systems, it is a question of automatically matching up the design blueprints and then seeing which bits of code can be re-used.

Smialek points out it may not be as simple as stripping out the relevant points of functionality from one system and pasting them straight into the new system - some adaption work might need to be done. Even so, it is still a great deal quicker and more efficient than starting from scratch.

"When you have a similar problem to that solved previously you put in the design and the code, and of course you might have to adapt it slightly to your new problem, but the majority of work was already done on the previous project," he says.

Any size of repository

He gives as an example different types of registration. "Somebody logging onto a website might have to press a button which causes a form to pop up which must be filled in. You press enter and the system checks the validity of the data and registers it in the memory.

"This type of functionality can be used between different types of

program domains so doing something as different as registering a computer in a warehouse could have the same logic as registering a user in an online system - so much of the same system design and code could be copied," he says.

In practical terms, a repository might just be in one company, or widely shared. The project has set up an online open-source repository with its own server, and project members from seven different countries have contributed software and trialled the system.

Once fully validated, the next step is to try to commercialise it. A software tools company, which was not part of the project, has expressed interest in incorporating the system into its product suite, reveals Smialek.

"What it will do as a commercial product is to reduce considerably the amount of work required to develop a new [software](#) application, and that means the ability to develop more and larger systems using the same human resources, which is bound to have a wide appeal," he concludes.

More information: [ReDSeeDS project](#)

Provided by [ICT Results](#)