

## Predicting the effects of modifications in software

*Large software systems are usually in service over decades. Studies show that 75-95% of overall costs fall on maintenance and further development. Austria's national insurance system operates numerous large software systems. All requests for modifications of these software systems converge at the system managers.*

### OVERVIEW

Before the implementation of a modification request, possible effects on the software system need to be detected. The Upper Austrian Regional Health Insurance (OÖ GKK) is responsible for maintaining one of the largest software systems in the national insurance system. In cooperation with Software Competence Center Hagenberg, a procedure and a prototype were developed to improve prediction of the effects of modifications. In the medium range, costs will be saved by avoiding failures. Generally it is the job of testers to detect errors caused by modifications of the software. Many of these errors, however, could be avoided if an effect analysis is conducted systematically before implementation.

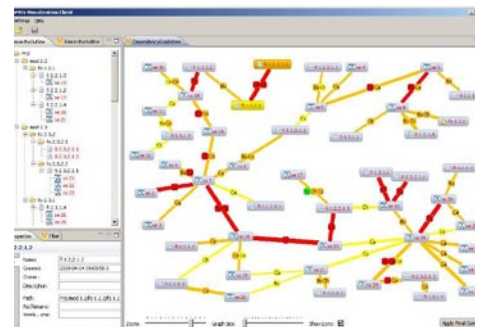
### CHALLENGE

The research field Dependency Analysis studies the identification of components and dependencies in software systems. Manual maintenance of dependencies is no longer possible for comprehensive software applications due to their complexity; therefore researchers have developed approaches that fully automatically identify components and dependencies. OÖ GKK has already recognized the potential of effect analysis and together with SCCH has developed a

procedure to conduct effect analysis systematically and efficiently. For OÖ GKK, it was important to have a fully automatic procedure that extracts the necessary data for effect analysis from existing data sources. The central data source is the version management system. Here all important artifacts (components, database scripts, documents, etc.) are registered that are delivered with a release. Through the application of special algorithms, the history of the registered entries reveals modification patterns. These logical dependencies are enriched with data from log files. Footprint analysis of the log files provides the invocation hierarchy of program components.

### SOLUTION

This innovative approach applies metrics to all dependencies. These metrics enable analysts and developers to compute all potentially affected artifacts based on a defined set of artifacts. Here they can define thresholds for individual metrics in order to tune the computed results to the context of the modifications. The approach also provides for evaluation of the quality of the analysis results after the implementation of modifications by checking to what extent the computation algorithm of the tool and the results for the users match the actually modified artifacts.



### CONTACT

Dr. Wolfgang Beer  
Area Manager Software Engineering  
and Technology

Tel.: +43 7236 3343 858  
Fax: +43 7236 3343 888  
E-Mail: [wolfgang.beer@scch.at](mailto:wolfgang.beer@scch.at)  
Web: <http://www.scch.at>

Software Competence Center  
Hagenberg GmbH  
Softwarepark 21  
4232 Hagenberg