

The COTS Spot

Automating Design Search

Robert C. Seacord

In my last column, I discussed the relationship of design and search, namely that the design of systems based on commercial off-the-shelf (COTS) components consists of a search for compatible *ensembles* of commercial components that come closest to meeting system objectives[1]. In this column, I discuss a technique and tool for automating this process that, of course, incorporates search technology.

Broad Landscape

The commercial software market offers a broad landscape of software standards, technologies, and products. While it is sometimes easy to discern and relate the features of this landscape, other times components are hopelessly commingled. COTS-based system-development processes, such as the use of model problems [see <http://interactive.sei.cmu.edu/news@sei/columns/the_cots_spot/2001/2q01/cots-spot-2q01.htm>], can be used to establish compatibility among defined component interactions. Evaluation techniques, such as risk-misfit, can be used to establish preference of one solution over another. [For a presentation about risk-misfit, see <http://www.sei.cmu.edu/cbs/cbs_slides/99symposium/056pr.pdf>.] However, neither of these processes guarantees that the entire design space is searched, primarily because these are labor- and knowledge-intensive processes (with both labor and knowledge often in short supply).

Automation

We at the SEI have made a number of attempts to solve this problem. One effort, a collaboration with the National Institute of Standards and Technology's Manufacturing Engineering Laboratory, attempted to automate knowledge collection. The resulting prototype, Agora [link to <http://www.sei.cmu.edu/publications/documents/98.reports/98tr011/98tr011abstract.html>], was successful in automating the collection of component interface information using a combination of search technology and introspection [2]. However, Agora was limited to gathering information maintained at runtime by the component model. In no case did this include semantic descriptions of the component or component methods, and in many cases the amount of useful information that could be gathered was nearly nil.

Having probed the limitations of automated knowledge collection, we decided to adopt another approach, namely automating the evaluation of ensembles formed from qualified

component information. In this case, knowledge about components, in the form of component specifications, is provided by component developers, system integrators, and other software engineering professionals. The knowledge-based automated component ensemble evaluation (K-BACEE) tool can then reference this information in evaluating the compatibility of component ensembles [3].

K-BACEE

K-BACEE helps automate the search for compatible ensembles of commercial components that come closest to meeting system objectives. This, of course, requires that the system integrator provide a statement of the system objectives. To this end, K-BACEE accepts a *system manifest* that, simply stated, provides a list of components that are required by the system. This is more than a simple software-requirements specification because it assumes a decomposition of functionality among components. The system manifest can also be complemented by a description of the interactions allowed among the components and additional constraints on both individual components and the overall system. The extended manifest, in this case, is very close to an architectural description of the system.

K-BACEE searches the repository of component specifications for components that match the system objectives described in the manifest. These components are then grouped into possible ensembles and evaluated for compatibility. Compatibility is assessed using a knowledge base of integration rules.

As part of the K-BACEE effort, the SEI is fostering a community of system integrators and component developers to assist with knowledge-base development. In support of this community-building effort, a birds-of-a-feather session is planned for the International Conference on COTS-Based Software Systems (ICCBSS) conference (see <http://www.sel.iit.nrc.ca/iccbss/>) in Orlando, FL, in February 2002. For more information on the K-BACEE community effort please see the K-BACEE Web site at <http://www.sei.cmu.edu/cbs/k-bacee/>.

Summary

While K-BACEE cannot possibly guarantee that the entire design space is explored, it does allow a computer to do what computers do best—evaluate a large amount of information quickly in an unbiased manner. When used in this manner, K-BACEE can be considered a “discovery aid” that allows the system integrator to consider a broader range of possible component ensembles than is otherwise possible.

An initial K-BACEE prototype has been developed and continues to be improved. Community assistance is actively being solicited and incorporated in the ongoing development effort.

References

[1] Wallnau, K., Hissam, S., & Seacord, R. *Building Systems from Commercial Components*. Addison-Wesley, June 2001, ISBN: 0201700646.

[2] Seacord, R., Hissam, S., & Wallnau, K., "Agora: A Search Engine for Software Components." *IEEE Internet Computing* 2:6, November/December 1998, pgs. 62-70.

[3] Seacord, R., Mundie, D., & Boonsiri, S. "K-BACEE: Knowledge-Based Automated Component Ensemble Evaluation," published in proceedings of the 2001 Workshop on Component-Based Software Engineering, held in conjunction with the 27th Euromicro Conference, Warsaw, Poland, September 4-6, 2001, IEEE Computer Society.

About the Author

Robert C. Seacord is a senior member of the technical staff at the SEI and an eclectic technologist. He is coauthor of the book *Building Systems from Commercial Components* as well as more than 30 papers on component-based software engineering, Web-based system design, legacy system modernization, component repositories and search engines, security, and user interface design and development.