Software Architectures Change Propagation Tool (SACPT)\textsuperscript{1}

W. Abdelmoez, M. Shereshevsky, R. Gunanala, H. H. Ammar\textsuperscript{2}
Lane Department of Computer Science,
West Virginia University
Morgantown WV 26506

Bo Yu, S. Bogazzi, M. Korkmaz, A. Mili
College of Computer Science,
New Jersey Institute of Technology
Newark NJ 07102

\section*{Abstract}

To assist the analysis software architectures, we have developed a web-based CASE tool that automates the steps of capturing, modeling, and inspection of software architectures, in order to derive and display the change propagation probabilities.

\section*{1. Functional Description}

The process of deriving our proposed metrics through the CASE tool includes 5 steps that are centered on the analyst uploading the software architecture’s metadata, configuring the resulting model, and finally validating this model before it is analyzed by the tool’s metrics layer.

First, through a web-interface, we enable analysts to upload architectural artifacts for the software architecture they wish to analyze. The CASE tool subsequently parses this input for metadata consisting of components, data types, and connectors, and generates a corresponding MOF-based model in a centralized repository, as shown in Figure 1. After the initial model has been generated, the analyst is further presented with various configuration options for the model. These options include equating components with files, classes and custom-defining components by aggregating files or classes. In addition, the tool will post all the data types that it does not recognize and asks the user to submit an estimate data type. With the model configured and refined by the analyst, the tool can now display the change propagation matrix.

\section*{2. Structural Description}

The CASE tool’s structure can be partitioned into three logical layers, shown in Figure 2. First, a metamodel layer captures, models, and stores a software architecture’s component, data types, and connector information. Second, a metrics layer inspects the generated model and applies our algorithms to derive the corresponding metrics. Finally, a web-based user-interface layer displays the metric results to the analyst. Each of these layers is implemented as a series of software component libraries written in C# on the Microsoft .NET\textsuperscript{TM} platform. In addition, the meta-model layer is consistent with the OMG Meta-Object Facility (MOF\textsuperscript{TM}).

Currently the CASE tool supports two different input formats, Java Understand \cite{1} and UML \cite{2}, to produce the change propagation matrix. In the future we are planning on adding more input format types that our CASE tool supports. The CASE tool is available online at the following URL: http://www.ccs.njit.edu/swarch/tools.htm

\section*{3. References}

\textsuperscript{1} Java Understand http://www.scitools.com/uj.html.
\textsuperscript{2} UML Revision Task Force. http://www.uml.org/

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{SACPT Metamodel (CML/Mof Metamodel)}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{SACPT Architecture}
\end{figure}