Java Based Component Expert Framework for Automatic Application Builder

AUTHORS

Lukasz Dutka[1] and Jacek Kitowski[1,2]
[1] Institute of Computer Science, AGH-UST, Cracow/Poland
[2] Academic Computer Centre CYFRONET-AGH, Cracow/Poland

ABSTRACT

Automatic Application Builder (AAB) for automatic service selection for scientific and business grid enabled workflows requires Component Expert Architecture as a framework for service selection in a knowledge based environment. However, most of the contemporary state-of-the-art knowledge driven solutions are written in Java programming language, and thus it implies that AAB framework should be written in that language. This work disscusses a comparison of a Java prototype implementation of CEA elements to the C++ one, previously used in CrossGrid project.

OBJECTIVES

The complexity of modern grids force to introduce automatic or semi-automatic tools for assisting in workflow construction

The expected solution should be scalable and expandable

Ontologies for the decision support

User requirements and preferences are important for automatic decision support

AAB refines previously constructed abstract workflows making them ready do be executed by the workflow execution facility.

Parcial results of workflow execution are important for selection services – AAB can be called during workflow execution too, not only before it.

AUTOMATIC APPLICATION BUILDER

The AAB module will be responsible for filling the workflow skeleton (workflow skeleton is a special form of workflow, which contains classes of grid services instead of real ones) obtained in the 'Service Selection' step. In the next step the skeleton is filled up with the selected realization of services (or selected components); this selection is based on AAB, i.e., the selection defines physical instances (marked by small letters a_1, a_2, ... a_n, b_1, b_2, ... b_n) of each class of services or class of components (marked by A, B, C, ...). The selection is performed using expert systems and knowledge base approach.

AAB is the connector filling the gap between the steps: 'Workflow Definition' and 'Workflow Execution'. The AAB module, using Component-Expert Architecture taking into account many factors, decides which grid service is the best. The most important factors taken into account are: detailed specialization of every potential grid service, all important information in external knowledge databases, internal knowledge written in rules and some context stated in call-environment.
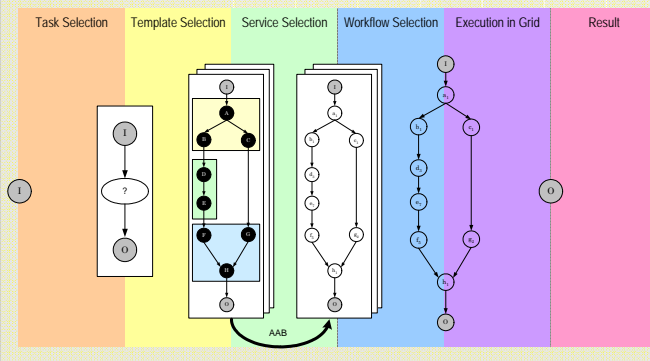

COMPONENT-EXPERT ARCHITECTURE
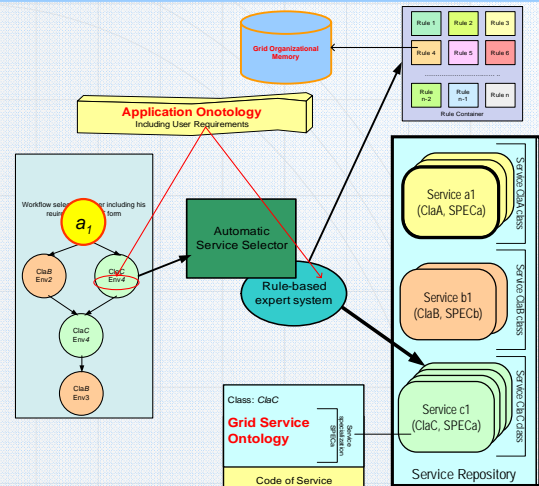
AAB applies Component Expert Architecture in the following way:

Abstract workflow consisted of service classes needs to be filled up by the real instances of the respective services. AAB processes each node trying to find the best service instance already registered in Service Repository.

Automatic Service Selector obtains request to find the best service instance for a specific service class and for a specific context described in ontological way. The context includes current status of workflow as well as user requirements.

Automatic Service Selector using built-in Rule Based Expert System, exploiting Grid Organizational Memory decides which instance of all registered service instances is the best in that context. It is important to remind that all services have their Grid Service Ontology. The ontologies describe service specialization.

Then the best instance of the service replaces node in the abstract workflow.

The process is repeated for all elements of the workflow. Sometimes it is not possible to decide which service is the best for a specific node in abstract workflows, since it is important to take into account results of execution other services. Thus, the decision must be postpone to the better moment.

PERFORMANCE COMPARISON
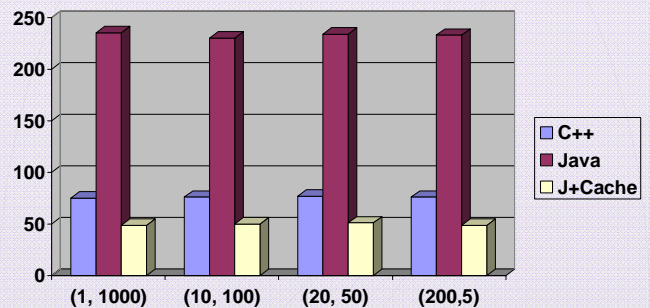
The presented in the chart above comparison of execution 1000 selection queries processed by Component-Expert Framework implemented in C++, Java as well as Java with cache option, shows that well implemented Java Component Expert Framework can be as good as the C++ implementation. The experiment was carried on by running 1000 queries with different concurrent strategies. The pairs (x, y) in the chart means that we ran x concurrent queries repeated y times.

CONCLUSIONS

The application of Component-Expert Architecture to the selection of grid services allows in easy way developing flexible and self adoptable solutions which are able using newly registered components (grid services) without software recompilation and any other programmers' effort.

The discussed within this work experimental results of the comparison Java implementation to C++ one allows us claiming that Java environment is good enough to implement the CEA approach and since AAB tool is deeply rooted in knowledge processing the final Java implementation will be much simpler then C++ one since of plenty ready-made solutions for ontology and knowledge processing available for the Java environment.