

From ADEPT to AristaFlow BPM Suite: A Research Vision has become Reality

Peter Dadam¹, Manfred Reichert², Stefanie Rinderle-Ma¹, Andreas Lanz¹,
Rüdiger Pryss¹, Michael Predeschly¹, Jens Kolb¹, Linh Thao Ly¹,
Martin Jurisch², Ulrich Kreher², and Kevin Göser²

¹Institute of Databases and Information Systems, University of Ulm, Germany

²AristaFlow GmbH, Ulm, Germany

1 Introduction

During the last decade we have developed the ADEPT next generation process management technology. Its features and its different prototype versions attracted a number of companies. However, an enterprise cannot base the implementation of its process-aware information system (PAIS) on an experimental prototype, especially if maintenance and further development are not assured. At the beginning of 2008, therefore, we founded a spin-off as joint venture with industrial partners to transfer ADEPT into an industrial-strength product version called *AristaFlow BPM Suite*, and to provide maintenance support for it. The product version is now available for academic and industrial use.

The work done in the ADEPT project on ad-hoc deviations at the process instance level and process schema evolution has been documented in many research papers, and is therefore rather well-known. Much less known, however, is another fundamental aspect of the ADEPT project which significantly influenced and guided our research work, namely *ease of use*. Although this may sound like the typical lip service, we consider ease of use as being fundamental for the broad usage of process management technology in different domains. It needs not only be achieved for end users, but should be provided to process implementers and application developers as well. Obviously, ease of use does not come for free; i.e., somebody has “to pay the price”. Supporting ad hoc changes at the process instance level, for example, requires a profound understanding of basic PAIS concepts as well as deep knowledge about PAIS internals. If such system-near knowledge is required for process administrators or application programmers, however, the battle will be lost before it will have begun.

We all know: “There ain’t no such thing as a free lunch.” Regarding the user groups for which ease of use shall be achieved, however, we can observe that one party is missing: the implementers of the fundamental PAIS technology. When developing ADEPT we have had one shining example in mind which has enabled ease of use by hiding complexity beneath the surface: relational database technology. Our basic belief was that we would be able to achieve a similar effect for PAIS if we were able to develop the adequate underlying theory. One of our basic challenges was to develop a technology which supports “correctness by construction“ during process composition and which guarantees correctness in the context of dynamic process changes. This was probably the most influential challenge for our research activities.

It had also significant impact on the development of the AristaFlow BPM Suite. In particular, we had to hide the inherent complexity of process-orientation (especially in conjunction with process flexibility) as far as possible from system administrators and application programmers; i.e., we have to perform all complex things “beneath the surface” in the process management system.

2 Ease of Use Aspects

To speed up process implementation, AristaFlow pursues the idea of process composition in a “plug & play” style supported by comprehensive correctness checks. These checks are accomplished in such a way that runtime errors during process execution can be excluded to a large extent. As prerequisite, implicit data flow and other dependencies among application services, being relevant for their execution order, have to be made known to AristaFlow to be incorporated in the correctness checks. AristaFlow provides an intuitive graphical editor to process implementers, and it applies a *correctness by construction principle* by providing at any time only those operations to the user which allow to transform a structurally correct process schema into another one; i.e., change operations are enabled or disabled according to which region in the process graph is marked for applying the respective operation. Deficiencies not prohibited by this approach (e.g., concerning data flow) are checked on-the-fly and are reported continuously in the problem window of the *Process Template Editor*.

Another goal was to make the assignment of application functions to process steps as simple as possible; i.e., a process implementer should not need to know details about the implementation of these application functions. However, this should not be achieved by undermining the correctness by construction principle. Both goals have been achieved. All kinds of executables, that may be associated with process activities, are first registered in the *Activity Repository* as activity templates. An activity template provides all information to the *Process Template Editor* about mandatory or optional input and output parameters, as well as information about data dependencies to other activity templates. The process implementer just drags and drops an activity template from the *Activity Repository Browser* window of the *Process Template Editor* onto the desired location in the process graph.

A developer who wants to provide a new application function must implement a corresponding activity template and add it to the *Activity Repository*. This way it becomes available and accessible within the *Process Template Editor* during process modelling. To simplify implementation of such activity templates, we support several levels of abstraction. At the lowest one, we provide an *Execution Environment* for each kind of supported basic operation; e.g., AristaFlow offers execution environments for SQL statements, web services, EXE files, BeanShell scripts, basic file operations, and system-generated forms. Based on them one can rather easily develop customized activity templates for specific purposes like, for example, retrieving a set of tuples from the database which satisfy a certain predicate.

Enabling ease of use for end users is mainly the task of application developers. They decide how “manual” process activities interact with the end user. They also

decide whether the standard workflow client is used or whether a dedicated one shall be provided. An important prerequisite for realizing adapted user interfaces is to provide the appropriate methods to the application developer; e.g., to enable him to realize end user interactions in the context of ad-hoc changes. To implement clients with such capabilities, the application developer can make use of the powerful system functions provided by the AristaFlow API. To move, for example, an activity to another position within the process graph is rather simple to implement: After having identified an activity *x* to be moved, the application receives a list of activities *after* which *x* could be inserted. Having selected one or more activities, it receives a list of activities *before* which *x* could be completed. After having selected one or more activities from that list, the AristaFlow system will execute the operation “insert between node sets”. That’s it!

3 The AristaFlow Community Platform

Due to its “correctness by construction” principle, AristaFlow is ideally suited to teach the implementation of PAIS because one can very quickly compose robustly executable processes. AristaFlow supports also rapid prototyping of PAIS. One can at first, for example, only model the control and data flow among activities, and not assign any “executables” to them. Nevertheless, the process becomes already test-wise executable, because the *AristaFlow TestClient* will automatically assign forms to them so that one gets already a rather realistic impression how the final process will look like. Also the idea of service-oriented process development is very easy to communicate due to the plug & play style outlined above as well as the provided repository services.

We, therefore, have established a community platform to support the utilization of the AristaFlow BPM suite in higher level education and research projects. The intention is that users (including ourselves) help other users in using the system in these areas but providing, e.g., sample processes, auxiliary activity templates, organization models, auxiliary tools, and share experiences.

For an extended version of this paper see [1]. For further information please visit the following web sites:

- www.AristaFlow-Forum.de – for the community platform
- www.AristaFlow.com – for obtaining the AristaFlow BPM Suite
- www.uni-ulm.de/dbis – for information on our research activities

References

- [1] P. Dadam and M. Reichert: The ADEPT Project: A Decade of Research and Development for Robust and Flexible Process Support - Challenges and Achievements. Springer, Computer Science - Research and Development, Vol. 23, No. 2, pp. 81-97, 2009