



Carnot - First Impression

Author: Dieter E. Jenz

Quick Guide

What it is	Workflow Management System
You would use it for...	Modeling of private business processes and as an execution platform for automated and human processes
... when ...	you are fully committed to a J2EE-based strategy
Distinctive features	Fully exploits J2EE capabilities (e.g. transactions)
Relationships with other Carnot products	None
Alternative Products	None (as far as purely J2EE-based WfMS are concerned)
What to do	Evaluate
Vendor URL	http://www.carnot.ag

Overview

Carnot AG, a Germany based company with headquarters in Frankfurt and a subsidiary in the U.S., was founded in 2000. The company is privately held with backing from a venture capitalist firm. Entering the market very late, Carnot AG fully committed itself to the J2EE platform, and thus chose to address a particular market segment only: companies that have standardized on J2EE.

Research shows that the number of companies that have successfully deployed a Business Process Management System (BPMS) at the corporate level is still small. Process control logic is as yet buried in applications. However, awareness is rapidly increasing and as software architectures evolve, process control logic is moved into a layer of its own. BPMS have already proven suitable for accommodating this extra layer. In addition, many IT departments have already standardized on J2EE or are in the process of doing so. Carnot AG directs its Carnot product just at the intersection of these two mainstream movements.

In 2001, Carnot AG released the first version of Carnot, which it terms a Workflow Management System. Version 2 appeared on the stage in March, 2002, with some notable enhancements, such as Web Services support.

Carnot AG sells its product directly (in Germany and the U.S.) and through partners. At present, Carnot AG is actively seeking partnerships with ISVs in several countries.



Carnot Concepts and Architecture

Carnot cleanly separates the modeling environment and the execution environment. Before a process model can be instantiated and executed, it needs to be deployed in the run-time environment.

The Carnot execution environment consists of the Audit Trail database and one or more EJB servers. Hence, Carnot requires at least one EJB 1.1 or EJB 2.0 compliant application server, such as BEA WebLogic Server or JBoss, and a DBMS. Since Carnot runs with various application server products, companies are in the position to save on license fees, for example by using open source JBoss in the development environment and WebLogic Server in the production environment. In a similar way, companies can use Cloudscape as a low-cost solution in the development environment and Oracle8 for production. Since the database is accessed through a JDBC 2.0 compliant driver, the list of DBMS tested with Carnot may grow over time.

Relying on an application server as container for the execution of business activity implementations provides for scalability. Many application server products implement sophisticated load balancing and fail-over capabilities, which the Carnot process engine can exploit.

Process Modeling

The modeling environment features a graphical workflow editor. Contrasting many other business process modeling tools, Carnot builds on a clean meta model, which defines associations between activity, role, data, and application. Carnot supports workflow model versioning, allowing for the distinction of public, private and released version.

Business process modeling focuses on providing process models that have undergone simulation, analysis and optimization. It has then been shown, that the process model is implementable. Carnot provides basic support for this workflow through its process modeling tool. There are no dedicated simulation and optimization facilities. A dedicated process engine would be used to deploy process models, execute them and optimize them based on statistical information. When optimization is completed, the process model would be deployed in the production environment.

Carnot stores workflow models using a WfMC compliant XML dialect. Different types of activities are supported, such as manual and application activities. For application type activities, there is a 1:1 correspondence of activity and activity implementation (application), which must represent a logical unit of work (transaction). As such, Carnot is closely related to implementation. Other tools, such as ARIS, do not enforce such relationship, requiring a later refinement of process models as process design progresses.

Activities can be associated with different types of applications: Session Beans (run in the application server), Visible Java Beans and Java Server Pages (JSPs), Host Terminal Screens, and external applications.

Integration of external application systems (e.g. SAP, Siebel) is possible through adapters based on the JCA (Java Connectivity Architecture). The application calls a function of the common client interface of a JCA resource adapter.



Another option for integration of external applications is provided through Web Services integration, which significantly improves technical interoperability in that a Web Service implementation can reside anywhere and can be implemented in any language. However, due to current limitations in technology and standards, Web Services should only be used when specific criteria are met (security is not an issue, service is not transactional, etc.).

Deployment

A process model version must be deployed before it is available for instantiation and execution. The designer can debug process models before deployment, albeit with some limitations, since debugging occurs in the modeling space (hence, actual process participants are unknown).

A process model is automatically validated before deployment in the execution environment. Deployment of applications needs to be done separately.

Process execution

Carnot's process engine provides flow control over business processes. It is implemented as a set of Enterprise Java Beans (EJBs). Hence, the process engine can run in any EJB-compliant application server.

Carnot provides two types of desktops: The Carnot Execution Desktop and the Carnot Web Desktop. The Execution Desktop provides an execution environment for Visual Java Beans, which interoperate with backend components via RMI/IIOP. In contrast, the Web Desktop is an execution environment for JSP front ends.

The Audit Trail database logs various kinds of information during process execution, which can be used for status monitoring of process instances and activities.

Administration

Carnot provides an administration desktop and command-line tools for the administrator to perform administrative functions. The administration desktop also provides monitoring and analysis functions that let an administrator gain insight into executing process instances, the current status of activity instances, and so on.

The administration tool accesses the Audit Trail database in order to determine valuable information, such as processing times and idle times of process instances and activities.

Summary

Carnot is a modern, well-architected BPMS, which exploits Java technology to the full. Companies will find Carnot to be a suitable solution for intra-enterprise process



integration in a J2EE environment. As of today, Carnot AG has not begun to tackle inter-enterprise integration, which would result in implementing ebXML specifications. The opportunity to choose among application servers and DBMS is appealing. Carnot AG does not force clients into a vendor-homogeneous platform, like some competitors, who forge tight links between the BPMS/WfMS and the application server.

Unlike most competing BPMS product suites, Carnot relies on the JTA architecture, which provides for transaction consistency and integrity. The two-phase commit protocol ensures that both the state of the process instance and the state of the activity implementation (application) are in sync. However, crossing the boundaries of the J2EE universe (for example by integrating a non-Java activity implementation with transaction properties), will result in compromising on consistency and end-to-end data integrity.

In the current version, Carnot does not provide support for the exchange of XML-encoded business documents between process engine and activity implementations. Also, Carnot does not contain a rules engine. It would be a definite advantage for users to be able to change business rules dynamically. From a development point of view, a rules engine would open up a way to move business rules out of application logic.

Carnot's pricing model is CPU based with a typical license fee of EUR20,000 (approx. \$18,000). This brings Carnot to a very favorable position compared to the competition.

Overall, Carnot is a cleanly architected BPMS without the burden of pre-J2EE days. The homogeneity of the J2EE environment turns out to be a huge advantage. Yet it can integrate with legacy applications through adapters, albeit losing transaction consistency and end-to-end data integrity, just as competing products.

Clearly, Carnot is a product to evaluate. However, a technically superior solution makes only one side of the coin. Carnot's success to a great extent depends on whether Carnot AG is successful in finding the right partners that promote Carnot to their clients.

Although this information is believed to be accurate at the time of publication, Jenz & Partner GmbH cannot and does not warrant the accuracy, completeness or suitability of this information or that the information is correct. Jenz & Partner GmbH and/or Jenz & Partner GmbH analysts and consultants cannot be held liable for any damages caused or alleged to be caused directly or indirectly by decisions made using any Jenz & Partner GmbH material.

Names appearing in this document that are registered trademarks are not mentioned as being so, nor is the trademark symbol inserted with each mention of these registered trademarks. Product names and company names mentioned in this publication are the property of their respective owners. In no way does Jenz & Partner GmbH have the intention of infringing on any registered trademark.