



ANTIPODES

WORKFLOW.CUBES



ANTIPODES **WORKFLOW.CUBES**

WORKFLOW DEFINITION: A WORKFLOW CONSISTS OF PROCESS LOGIC AND ROUTING RULES. THE PROCESS LOGIC DEFINES THE SEQUENCE OF TASKS AND THE ROUTING RULES THAT MUST BE FOLLOWED, AS WELL AS THE DEADLINES AND OTHER BUSINESS RULES IMPLEMENTED BY THE WORKFLOW ENGINE.

ANTIPODES WORKFLOW.CUBES IS A SYSTEM FOR VISUAL MODELING AND AUTOMATION OF BUSINESS PROCEDURES IN A COMPANY.



ANTIPODES MEDIA

Sofia 1172, Bulgaria
Dianabad, G.M. Dimitrov Blvd. bl.60 entr. 4, fl.4 ap. 83
tel: +359 2 962 69 40, fax: +359 2 962 69 41
e-mail: info@antipodes.bg, web: <http://www.antipodes.bg>



Why implement a Workflow system?

By implementing a “common standard” throughout the company you can establish a more efficient style of work. Visual control and optimization of the procedures and reuse of predefined procedures will help you implement the “best practice” model for your business.

Advantages of Antipodes Workflow.Cubes

- The design of processes and procedures with different complexity is done in the Workflow Architect using a Graphical User Interface (GUI).
- Visual control helps managers better understand the business logic of the company processes.
- Simple Java snippets can be used when the need to define more complex procedures occurs.
- Introduction of “roles” makes changes to the business easier to manage, by separating the business logic from company structure.
- Storing of predefined procedures and parts of procedures helps in the reuse of process definitions and makes implementation quicker and easier.

Technical Description and Architecture

The development of Antipodes WorkFlow.Cubes follows and extends the OMG specification comprising particular problems and enhancements.

WorkFlow.Cubes is entirely developed in Java and may run on every Java enabled computer (Java 1.4 Standard Edition or higher).

Basic Modules

WF Container - this is actually the server that makes the communication with the clients. It receives requests and transmits the responses. It also creates a common platform for all the rest of the modules.

WfDBManager - a module that handles the connection to the database - supported are all databases that have a JDBC 2.0 compliant driver.

WF Definitions Manager - this is the module that manages the definitions of the processes. It interacts with the WF Architect, where the definition of the processes is done. XML input is also supported to make possible interaction with other systems in the future.

WF Configuration Manager - reads all the needed configurations and sets the work of the server accordingly.

WF Instances Manager - a module that serves the purpose of creation and management of new processes based on the available definitions.

WF Workers Manager - a module that serves for the dynamic instantiation of the new business components, named "Worker".

WF Process Manager - this module takes care of the correct tasks assignment, the process behavior and the behavior of all tasks that compose the process itself. It interacts with the clients (Web, Microsoft Outlook, etc.), where tasks are displayed. It also keeps track of the execution phase of every process.

WF Unit - an interface that serves to describe a definite task. The WF Unit interface is a part of the WF Process. It handles the question "Who should do the task". The performer functionality is not limited only to "a role or a single user", but also to runtime allocation. WF Unit also supports execution of subprocesses within a process.

WF Worker - an interface containing the business logic of a task that must be implemented. It handles the question "What should be done" when a task is started. It is an interface with open Architecture, which everyone can implement to build custom activities no matter how complex they can be.

Integration

Currently WorkFlow.Cubes is fully integrated with Teamwork.Cubes. Additional modules for integration of WorkFlow.Cubes with Outlook are in a final stage of development process. Antipodes WorkFlow.Cubes is capsulated and designed to be compatible with every client that may process a correct command to the Workflow server in order to receive response to its request. This might be a command for initiation of new process, command for processing data to a definite task, command for termination of a process or task, etc.

WF Server File structure

The following is a list of directories and their content:

bin - script that starts, terminates and restarts the server;

etc - configuration file

ext - packages containing Workers

lib - contains the server and all additional libraries necessary for its proper functioning (e.g. JDBC)

logs - contains history log

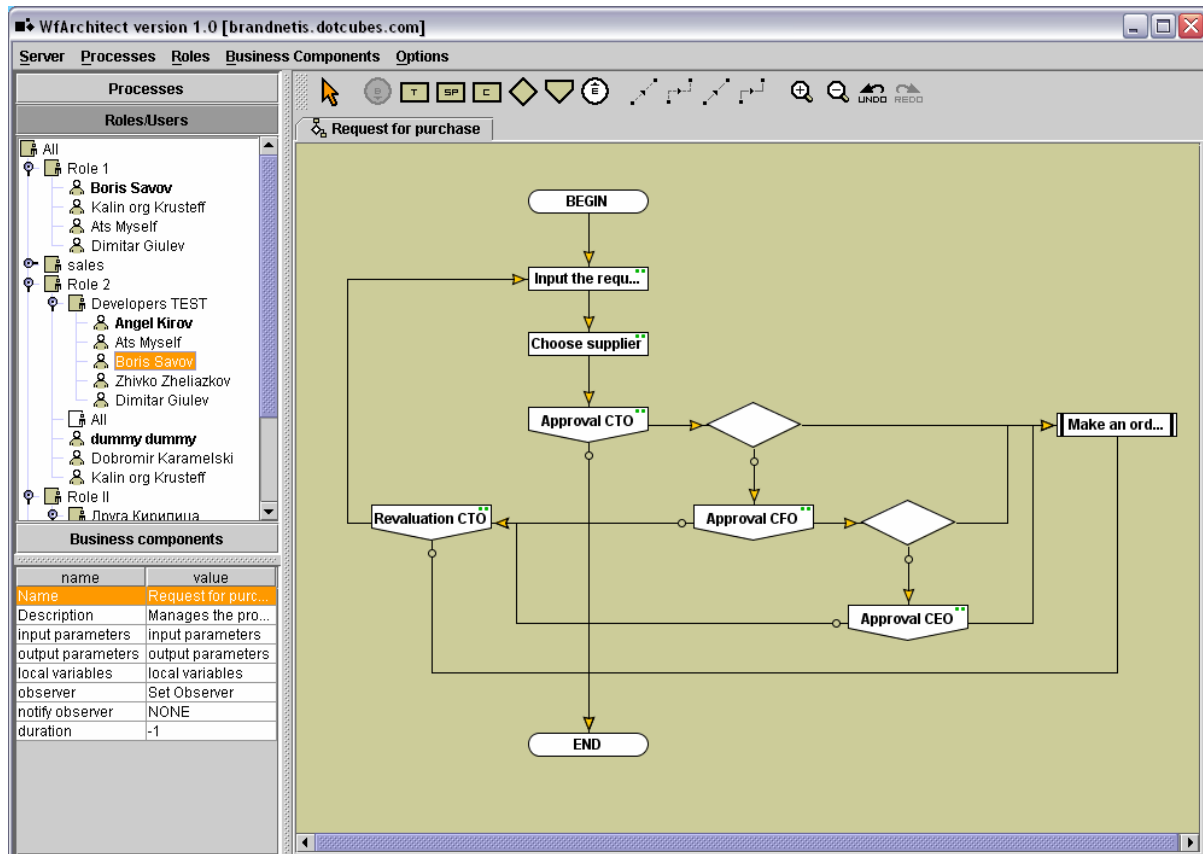
WF Architect

Workflow Architect is a Graphic User Interface (GUI) for building block diagrams to visually depict processes. WF Architect is Java based application that can work on any platform.

The WF Architect Interface is user-friendly and easy to manage for creating procedures.

The program communicates with the server via TCP using XML for the application layer of TCP/IP communication protocol. The system is designed to allow simultaneous interaction among a number of administrators. Thus parts of a process or even different procedures can be built by many users at the same time.

When no server access is available, work can be done in offline mode. You can revise the procedures wherever you are and upload them on the server when connected. This has been implemented through a special option for synchronization with the server status, taking snapshots of the processes, users, groups, Workers, forms available on the server.



The completed procedures are loaded on the server, which automatically assigns tasks according to the process definitions.

WF Worker

WF Worker is an interface containing simple and clear methods. Each Java class implementing this interface may participate in the process as a definite task. The WF Process and the WF Unit set the moment and the user that start the task and the WF Worker sets the action that is expected to be taken by this task.

The WF Worker may invoke native methods and code so that existing modules, programs or drivers for devices may be added. In this way we can reuse existing code.

In order to work properly with the Workers, special constant or variable parameters can be assigned to the task and the processes. The purpose is to start the process with some parameters (if necessary), and make the communication between the Workers possible through these parameters. Assuming that, the process can be defined as a program, containing multiple procedures with incoming and outgoing parameters.

Performer assignment

Due to the specific of the WF Process, these procedures can be assigned to a definite user or a definite role. The rules defining the user to whom the task must be assigned are described in the WF Architect. There is another special interface WF RoleToUser that indicates the user who must take the task, when a definite role and definite parameters are assigned. This interface can be implemented according to the client's preferences. The class used to indicate the user according to the role is defined in the configuration file. Run-time assignment is also supported.

The Process

The process is described graphically as a block diagram. The following containers for Workers are available:

BEGIN - indicates the start of the process. There is only one "BEGIN" per process. The user or the role that have the rights to start the process are assigned to this container. Time events can also be set to start automatically the process at a given period of time.

END - this is the end of the process. Multiple "ENDS" may exist in one process. When the process ends all active tasks are terminated.

TASK - task is the main container used in the design of processes. It supports assignment of roles/users, incoming and outgoing parameters and WF Workers.

SUBPROCESS - a container that depicts a whole process (subprocess), which can be included within the definition of another process. The Subprocess' incoming and outgoing parameters set in the definition of the subprocess are automatically transferred to incoming and outgoing parameters of the container.

CODE - this is a container that holds Java source code. The Java code executes when the task is started. It may serve to manage the process, start specific programs, change or determine the parameters' values.

CONDITION - it is similar as the CODE but returns TRUE or FALSE at the end. TRUE or FALSE serve to branch out the process in two directions.

USER DECISION - it is the same as TASK with added functionality of human decision or code, returning TRUE or FALSE at its end. TRUE or FALSE serve to branch out the process in two directions.

Connections between tasks

The relations between the separate tasks are described by vectors that connect the tasks. There are two different types of vectors:

Ordinary vector - indicates the tasks that should start when the previous task has ended.

Prerequisite vector - this is a compulsory relation. It means that the task may start only if all previous tasks leading to it with a prerequisite vector are completed.

In order to branch out the process, there should be an appointed direction in which the process will continue if TRUE or FALSE are returned from the CONDITION and USER DECISION containers. For that purpose another parameter of the vectors connecting the tasks is assigned.



INTEGRATION BETWEEN WF SERVER AND TEAMWORK.CUBES

Integration between WF Server and Teamwork.Cubes

Antipodes Teamwork.Cubes (www.dotcubes.com/_teamwork/) contains additional tab, showing the current tasks that should be completed, as well as the list of processes that the user may start. If a certain task has exceeded the time-limit, the user is alerted. The user may track the stage of accomplishment of all processes started by him, including information about which user has implemented a definite task and which task is completed. If a user is set to be an observer of processes he may track it as well. The observer may also terminate the observed process.

The screenshot shows the Teamwork.Cubes web interface in a Netscape browser window. The browser address bar shows the URL: <http://brandnetis.dotcubes.com/7f406a13847117321>. The page header includes a welcome message for Boris Savov, a log out link, and the date Tuesday, 20 May 2003. The navigation menu includes links for MY..., FORUMS, OPTIONS, HETMC CAT, HOME, ADDRESSES, MAIL BOX, CALENDAR, TASKS, NOTES, FILES, FAVORITES, and GROUPS. The current page is the Task list.

The main content area displays a task list with the following details:

- Due date: before May 20, 2003
- Show all tasks regardless of the date
- Show done as well
- Currently showing 1 - 1 from 1 items
- Task Title: [Input the request](#)
- Due date: No Date
- %Acc.: 0%
- Task created on: 05/20/03
- Task Description: Fill in all the fields for the form for buying new equipment

The right sidebar contains a calendar for MAY 2003, a "Due today" section with a warning for "May 20, 2003, Input the request", and a "Your Roles" section listing roles and processes.

Process	Task Title	Task created on	Task Description
Request for purchase(455)	Input the request	05/20/03	Fill in all the fields for the form for buying new equipment

A special module in Teamwork.Cubes generates a visual representation of the relevant WF Worker. There is ability to generate a form over a DB table and that form may become a part of the WF Process. These forms can be an adequate part of the process from which can be exported parameters to the Workflow, being a base for decision making.

ANTIPODES

WORKFLOW.CUBES



ANTIPODES MEDIA

Sofia 1172, Bulgaria

Dianabad, G.M. Dimitrov Blvd. bl.60 entr. 4, fl.4 ap. 83

tel: +359 2 962 69 40, fax: +359 2 962 69 41

e-mail: info@antipodes.bg, web: <http://www.antipodes.bg>