Object-Oriented Workflow Technology in InConcert

Sunil K. Sarin
XSoft (A Division of Xerox)
4 Cambridge Center
Cambridge, MA 02142
E-Mail: sarin@xait.xerox.com
Telephone: 1 617 499 4426

Abstract

InConcert is an object-oriented client-server workflow management system. An overview is provided of the functionality of InConcert and how it is implemented on an underlying relational database management system. The process of developing a workflow solution is reviewed, with emphasis on extensibility, ease of integration, flexibility and adaptability, and reuse.

1. Introduction

Workflow management software allows organizations to model, execute, monitor, and dynamically control business processes that involve multiple human participants and multiple software systems. By providing the right information to the right users at the right time, workflow management enables increased levels of productivity, competitiveness, and customer service.

XSoft's InConcert is an application-independent workflow management software toolkit. An InConcert solution, for a given business application, involves configuring the InConcert software with relevant workflow process definitions, and integrating with desktop and other software tools used in the work steps that comprise these processes. Such a solution typically has a mix of "out of the box" InConcert applications and custom applications (built with the help of InConcert tools) working together in concert. A well-designed workflow solution not only helps an organization perform its current operations more efficiently and effectively, but is also flexible enough to support process reengineering and improvement over time in response to changing business conditions.

Section 2 below describes the core InConcert functionality and how it is implemented using an underlying relational database management system. Section 3 describes how InConcert's object-oriented software and tools are used to develop a workflow solution for a business application. Section 4 describes some future directions for further streamlining of solution development.

2. InConcert Functionality & Architecture

2.1 InConcert Object Model

A process in InConcert is a collaborative activity that consists of:

- **tasks**, which are the individual units of work that comprise the process;
- **roles**, which designate the participants (users or programs) that perform the tasks;
- **references** to documents or other data objects that are manipulated by tasks in the process.

Tasks in a process have a hierarchical structure, with as many levels of decomposition as needed for the particular application. Parallel execution of tasks in a process is implicit, except when explicit dependencies are specified between tasks to constrain the order of execution. Some tasks may be designated as conditional by associating a predicate whose evaluation may cause the task to be "skipped" and not offered to any participant for execution.

A process definition provides a reusable template of tasks, roles, and references for a category of business processes, e.g., to process an insurance claim or manage the design of a product. A process instance, created from a process definition, represents an actual process in execution. At the time of starting the process instance, or during the course of its execution, the roles (and references) in the process are bound to specific users (and document objects) for that particular process instance.

A key aspect of InConcert is its treatment of a process as a data object, not a program or script. A process and its related constituent objects are mutable, subject to
users' access rights. This is the key to allowing a process instance, which is based on an often idealized process definition, to adapt to the particular circumstances that arise during its execution. Exceptions or other unanticipated conditions can be dealt with by modifying the task structure of a process, changing role and reference assignments, and adding new roles and references.

All InConcert functionality is accessible via the InConcert API (application programming interface). The API presents a collection of C++ object classes (such as process, task, role, etc.) and operations on these classes. (The InConcert API is also accessible through a C language binding, and will soon be accessible from a variety of scripting languages via OLE Automation.) These operations include retrieving information (properties, attributes, relationships) about an object, updating an object (e.g., marking a task as completed), creating an object (e.g., a new process instance), and associative querying over collections of objects (e.g., all tasks that are available to be worked on by a given user).

The InConcert object model is extensible, in that applications can, via the InConcert API, define new subclasses of the predefined object classes and define new attributes of new or existing classes. A subclass of another class inherits all the attributes of the parent class, and may have additional attributes defined for it. Attributes of InConcert objects can be used for a variety of purposes:

- information displayed to and interpreted by users, e.g., descriptive text;
- information that determines conditional execution, e.g., an approval task may not need to be performed if the value of the dollar amount (an attribute of the process instance) is below a certain threshold;
- information interpreted by custom software (see Section 3.2) that integrates workflow with other system services, such as imaging and document management, engineering and product data management, project management, decision support, legacy systems, database and forms applications, and document editors and other interactive desktop tools.

InConcert also includes an extensible event and trigger model. A trigger specifies an action, which may be an electronic mail notification, creation of a new process instance, or invocation of an arbitrary application program, that is to be executed on occurrence of a system-defined or application-defined event.

2.2 InConcert Architecture

InConcert's client/server architecture has three "tiers":
1. InConcert client software.
2. InConcert server software.
3. Database, file, and mail services.

InConcert clients use the InConcert API to offer workflow functionality to users. These include the user interface and administrative client software that are part of the InConcert product, as well as custom applications that integrate InConcert functionality with other software used in the particular business environment. An InConcert client may initiate sessions with multiple InConcert servers in a distributed environment.

The client library that provides the InConcert API communicates with the InConcert server using remote procedure calls (RPC) that are transparent to the client program. The InConcert server isolates the client software from the particulars of how InConcert objects and their associated data are managed. These data can be stored in any of several popular relational DBMS products from which the customer can choose; InConcert client software is DBMS-independent and works with any implementation of the InConcert server. Relational tables are used to store all InConcert data except for document content files which are stored in a file system with locator information in the DBMS. Use of proven commercial DBMS software allows InConcert to take advantage of ongoing improvements in the robustness and scalability of this technology, as well as associated query and reporting (or "data warehouse" analysis) tools.

The InConcert server (which is multi-threaded in order to handle concurrent client requests) is responsible for accepting an RPC from an InConcert client and translating the call into the database accesses (SQL select, update, insert, and delete statements) necessary to achieve the desired semantics of the API operation invoked. All database accesses that comprise a given RPC are performed within a single database transaction, ensuring that the call is executed atomically and does not leave the database in an inconsistent state. This facility may be extended in the future to permit InConcert API calls to be executed within a larger distributed transaction under the control of a transaction processing monitor.

InConcert events and triggers are processed by a background component of the InConcert server. InConcert provides asynchronous trigger semantics [2], in that the execution of a trigger takes place in a transaction separate from the API call that caused the triggering event. Again, DBMS tables are used to
reliably record all events and triggered actions in order to guarantee eventual execution of triggered actions. The event log (a database table) can also be used for audit tracking and analysis of historical activity.

3. Building a Workflow Solution

Developing a workflow solution for a particular business application requires specifying the following kinds of objects:

- The business processes that will be modeled and enacted by the workflow system, and their constituent tasks (units of work).
- Users and groups, which are the entities to which tasks will be assigned (via roles).
- Documents or other data artifacts that will be manipulated from tasks.

In addition, user interfaces are needed for various activities which fall into the following categories:

- Process design (defining processes and their associated tasks, roles, and documents).
- Process management (initiating new processes, assigning work, monitoring progress, deleting completed processes).
- Task management (viewing tasks available for work, selecting tasks to work on, performing tasks by manipulating the associated data and applications).
- Administration of system objects and meta-data.

3.1 InConcert Application Suite

InConcert provides applications for each of the above activities, which are available to the solution developers and users. The InConcert administration tool, for managing users and groups, and object classes and attributes, is (for now) command-line oriented; the other applications are graphical interfaces with an object-oriented direct-manipulation metaphor.

The InConcert Process Designer module supports graphical layout of hierarchical process definitions or "maps", along with their associated role and document information. The user designing a process can create a new process definition (which may be based on an existing process definition) or can edit an existing process definition. Operations on the process structure include adding and deleting tasks (at any level in the task hierarchy of the process), adding and deleting dependencies, and specifying task properties such as durations and conditional execution. A new task can be added to a process as an "empty" task, or an existing task "template" (which may have nested subtasks under it) may be copied into the process; this supports reuse of previously-designed process subcomponents. The Process Designer also has an associated Task User Interface Designer module, which can be used to tailor the interface (see below) that is presented to the user performing a given task.

The InConcert Process Manager module supports initiation of a new process instance (either from the library of available process definitions, or as an ad hoc process with dynamic routing options), as well as management of process instances of interest to the given user (including but not limited to the processes this user initiated). The status of a process instance can be viewed, e.g., showing the progress of tasks against deadlines. Role and document assignments for the tasks in a process instance can be made. In addition, the task structure of a process instance can be modified dynamically (if the user has the appropriate authorization), via the same operations and user interface as the Process Designer module.

The InConcert Task Organizer module allows a user to view and work on the tasks assigned to them. The view can be customized to show attributes of interest to the user, and can be sorted, e.g., to show higher-priority or more urgent tasks first. On selecting a task to work on, the user is presented with the task user interface for that task. This form-oriented interface shows task status and other attribute information, a task checklist and help or instructions, and the documents associated with the task context. The user can edit the attributes (if appropriate), mark checklist items, and view or edit the associated documents. Opening a document (to view or edit) from the task user interface will launch the application (e.g., a word processor or spreadsheet) associated with the document’s file type.

A user working on a given task can view the associated process instance, to better understand the overall context for their task and the progress of related tasks. The user may also be permitted to make ad hoc changes to the process, such as:

- Delegation of the task to another user.
- Requesting rework of a previously-executed task.
- Other arbitrary changes to the task structure (as in the Process Designer, above) of the process.

In addition to the above modules, the InConcert application suite includes support modules for document management and reporting on processes and workgroups.

The InConcert applications offer a variety of capabilities for tailoring and for integrating with off-the-shelf desktop applications:

- Custom attributes, defined by the administrator(s) for the particular installation, can be viewed and updated by the user.
3.2 Custom Applications

The InConcert "out-of-the-box" application suite may be sufficient for a particular business problem if all of the work to be performed can be accomplished via the built-in capabilities (with tailoring) of the InConcert task user interface and launching of document applications. This is adequate for many prototyping and pilot exercises, but in a live production environment the work to be performed very frequently involves viewing and manipulating information in existing databases or document or image repositories, interacting with communications services such as mail, fax, or EDI, and so on.

For each kind of business task, the solution developer needs to provide an activity-based application that combines workflow (InConcert) operations with operations on other services; these operations should be packaged and offered to the user using familiar business terminology rather than the jargon of the underlying technologies. InConcert's extensible attribute mechanism supports this integration, by providing a mechanism for associating external data identifiers and locator information (e.g., a database key, or the URL for a World-Wide Web document) with processes, tasks, and documents. Thus, the application logic for a task may involve retrieving attribute values from InConcert and using these values to manipulate external databases and services, or obtaining such information from an external service and storing it as attribute values in InConcert for future use.

Such activity-based interfaces can be developed using any of a variety of application-building tools (popular ones include Visual Basic and PowerBuilder); the solution designer will tie together operations on the InConcert API and on other services in order to produce a coherent user interface or form for each particular business task. This interface can then be associated with a task (in a process definition or instance) using the same mechanism that InConcert provides for registering document types and applications to launch. Then, when an end-user (in the Task Organizer) selects a task of this type to work on, InConcert will launch the custom user interface for that task (and pass the necessary information) instead of InConcert's default task user interface form.

Not all tasks in a process necessarily require a human being to perform the work. For tasks that are to be automated, the solution developer writes an "agent" program to do the necessary manipulations of the InConcert API and the other services needed in performing the task, and registers the agent with InConcert so that it is invoked automatically when an instance of the task is ready to be worked on.

Other InConcert user interfaces (in addition to the task user interface) can be replaced by custom applications. Some organizations develop their own look and feel, and viewing and sorting options, for presenting tasks to the user (instead of the InConcert Task Organizer). Or, users may prefer to receive and manage assigned tasks (along with other work) via their mail inbox or a personal information manager. In other cases, the burden of selecting the "next" task to work on is taken away from the user and is performed by a background application that automatically "pushes" tasks to the user one by one according to application-defined criteria. In all such cases, the interactive or automated custom application makes use of the InConcert API (and other services) to accomplish its goal.

The InConcert Process Manager can also be replaced by a custom application. Some customers prefer to perform their process management activities via a project management system, and this can be (and has been) done using the extension capabilities of the project management system and the InConcert API. In some cases, initiation of a process does not require human interaction. A custom application can be written, for example, that watches for the arrival of a document or message or form and then invokes the InConcert API to start a process instance and attach the received data to make it accessible from tasks in the process.

Management reports on process status and history can be defined using any SQL-based reporting or analysis tool against InConcert's database views. InConcert provides a collection of standard report definitions, and these can be modified or additional reports defined, e.g., to correlate InConcert data with other database tables.

The InConcert Process Designer is rarely replaced by a custom application. In the future, with the development of Workflow Management Coalition standards for process interchange, customers will be able to design processes with, say, a BPR (business...
process reengineering) tool of their choice and then transfer the process definitions to a workflow engine (such as InConcert) for deployment and enactment.

In summary, a typical workflow solution consists of a combination of InConcert applications and custom applications. A common approach is to build a custom activity-based application (or automated agent application) for each kind of business task, use the InConcert Process Designer to tie together these tasks, and use the InConcert Task Organizer (from which the task user interfaces will be launched) to deliver work to users. A variety of other combinations are possible, and InConcert’s object-oriented interfaces and tools provide the power and flexibility to construct a solution that meets the needs of the business; in addition, modular separation of the components of the solution makes it easy to extend and adapt the solution as the nature of the work and of the processes evolves over time.

4. Future Directions

InConcert’s open architecture, reliance on industry-standard technology, and object-oriented approach to workflow have been used successfully in a variety of business contexts. InConcert supports the development of integrated workflow solutions that are flexible and easy to adapt to changes, and that exhibit a high degree of reuse. Future work is intended to further increase the productivity of both solution developers and end-users by extending the object-oriented approach in several ways:

- Enabling more rapid application development, by providing scripting language support (e.g., via OLE Automation) for the InConcert API, by providing industry-standard user interface components (e.g., custom controls) that can be embedded in other applications, and by providing the ability to embed separately-developed components (representing external data or services) in the InConcert applications.

- Supporting libraries of processes and process components, including version and configuration control, and extending InConcert’s inheritance capabilities to hierarchies of process definitions (similar to the Process Handbook [1]) in order to support development of more specific processes as incremental refinements to more generic ones.

- Supporting workflow processes that involve distributed participants. This will include both inter-organization distributed workflow, which will be based on the Workflow Management Coalition’s specifications for workflow interoperability [3], and intra-organization distributed workflow. The latter will involve replication of process data across multiple cooperating InConcert servers, in a way that is both object-oriented and dynamic. The unit of replication will be a process instance and its associated data (as opposed to, say, a slice of a database table), and the locations where such a unit is replicated will be implicitly defined by the locations of the participants assigned to tasks in the process. In addition, concurrency control over process data will be based on the concept of each task in a process being “owned” at any given time by one location (to prevent users at different locations from working on the same task), thereby permitting users at different locations to work on different tasks in the same process.

References