AwareSoft Pty Ltd
ABN 11 099 495 468, Suite 715,
1C Burdett Str., Hornsby NSW, 2077, Australia
Tel: +61 2 8076 7413, Fax: +61 2 8076 7418
www.awareim.com

Aware IM

Technical Whitepaper

Conceptual Framework

Aware IM is built on an assumption that most data management systems have many common aspects. Essentially any data management system requires data to be entered and stored somewhere (usually in a database), it also requires the data to be found, retrieved and edited and it offers some user interface to perform all these operations. Most systems also perform processing of this data according to certain rules (this is often referred to as "business logic" of a system). Apart from these main generic aspects most systems also need to take care of security issues, manage documents, generate reports, communicate with other systems etc.

The cornerstones of the Aware IM approach to the implementation of the above mentioned aspects of a data management system are listed below.

Business objects are carriers of data

In Aware IM the world of a data management system consists of business objects. Business objects encapsulate the data that needs to be entered, retrieved, edited and processed.

In Aware IM there is no data that exists outside of a business object.

In a way this model very closely reflects the real world. Business objects exist in every business – customers, accounts, orders, payments etc. The data that a business object encapsulates is represented as attributes of a business object. For example, an order may have the placement date, customer, line items, shipment address, shipment date, shipment number etc.

Business objects form relationships

Business objects can be linked with other related objects. For example, an order object can be related to an order line item object (an order may contain multiple order line items and an order line item belongs to a particular order).

Business rules are carriers of business logic

In Aware IM data processing (or business logic) is encapsulated in business rules. A rule specifies one or more actions that should be executed when the rule conditions are met. Conditions are optional and if none are specified actions are executed unconditionally. In other words a rule states what should happen and when.

Here are some examples of rules:

If Account.Holder.Age < 16 Then REPORT ERROR 'Account holder must be 16 years old or over'

LineItem.Total = LineItem.Price * LineItem.Quantity

If Reservation.Status WAS CHANGED TO 'Offered' Then SEND ReservaitonOfferEmail TO Reservation.Member

REQUEST SERVICE ProcessPayment OF PaymentProcessingSystem

If Fee.Status='Applied' Then PROTECT Fee.Amount FROM ALL

Actions of business rules can perform a variety of tasks. Most importantly they can create and modify business objects, i.e. perform data processing. They can also perform calculations, create or print documents, display information, exchange data with other software, etc.

Business rules are triggered when there is a change in data

Rules are triggered when certain events happen inside the system. Most importantly, rules are triggered when there is a change in the system's data, i.e. when business objects are created or modified.

When an object is modified (for example, the value of an object attribute changes) Aware IM evaluates all rules that depend on the changes executing the actions of those rules for which the conditions are satisfied. Note that:

Execution of actions may have a ripple effect – Aware IM tracks all changes made by the actions during rule execution and continuously re-considers all rules that depend on the data changed by the actions.

Not only does Aware IM consider all rules that depend on the modified object directly, but it also considers rules that depend on the changes indirectly, i.e. through objects that are related to the modified object.

Rule execution continues until there are no more actions to execute. The change to the system's data therefore moves the system from one logically consistent stable state to another logically consistent stable state through a period of instability when relevant rules are executed.

Processes are Links between User Interface and Business Logic

The initial trigger that leads to the creation or modification of a business object is almost always some external request to the system usually from a user who sends this request via the User Interface. In many cases this request starts a process, which creates or modifies a business object(s).

A process usually consists of actions that create a business object or find the existing object and modify it. This triggers evaluation of business rules. The rules start a chain reaction of other modifications until the system settles down. When this happens the process communicates the results back to the user.

Therefore a process represents a link between the User Interface and business rules – it triggers business rules in response to the request from the User Interface by performing initial creation or modification of a business object and then communicates the result back to the User Interface.

For example, a process transferring funds between two accounts can be made up of the following rules:

Display funds transfer form: ENTER NEW FundsTransfer

Display receipt: DISPLAY DOCUMENT TransferReceipt

Display client details: VIEW Customer

Note that the process is very small and is mainly concerned with allowing the user to enter necessary details and presenting the results of the operation to the user. All the business logic behind the funds transfer operation (such as checking whether there are sufficient funds in the source account, calculation of fees for the transfer operation etc) is expressed in business rules related to the object FundsTransfer (they are not shown here). Aware IM evaluates these rules as soon as the user enters the data for the FundsTransfer object (between "Display funds transfer form" and "Display Receipt" operations).

Although a process is primarily used as a link between the user interface and business logic, it can also be used as a container of un-ordered business rules. It is possible to call such a process from other processes or business rules and let the rule engine evaluate the rules and invoke the appropriate actions. Thus it is possible to combine the data-oriented approach to business rules (when rules are triggered by the change of data) with the service-oriented approach when the rule engine is explicitly invoked from some place in the system.

Major Technical Features

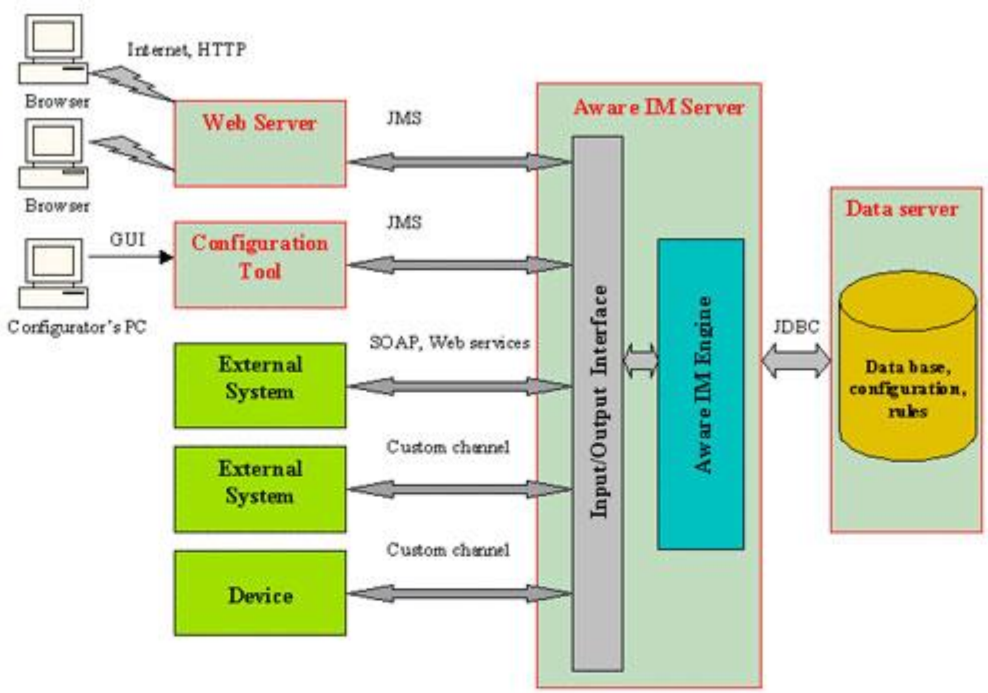
The following section lists some of the major technical features of Aware IM

- Aware IM is written in the Java programming language and is therefore independent of the operating system used.
- Aware IM works with any servlets container such as Tomcat.
- Aware IM is a multi-user system, which allows any number of concurrent users both in the Operation and Configuration modes.
- Aware IM runs over Internet/intranet and requires HTML 5.0 compatible Internet browser as a user interface agent in the Operation Mode. No software download or installation is required on the client side.
- Aware IM works on a single computer but is scalable across multiple clustered computers.
- Aware IM uses JDBC to communicate with an SQL-compliant database management software and is therefore independent of the specific database used.
- Aware IM allows users to configure fully functional systems quickly without having to know the details of the database technology. Specifically:
    - Database tables are created automatically based on definitions of business objects and their attributes.
    - Database tables are altered automatically if definitions of business objects and/or their attributes change.
    - Relationships between database tables are created and maintained automatically based on definitions of relationships between business objects (this is much simpler than the standard approach of linking database tables).
    - Database transactions are maintained automatically.
    - Database SQL queries are automatically constructed from definitions of Aware IM queries. These definitions are at a higher level of abstraction than SQL and

include, in particular, links between objects, which are automatically converted into database table joins.

- Aware IM employs non-procedural rule-based approach to data processing (business logic) instead of scripting or programming. At the heart of this approach is Rete algorithm-based rule engine.
- Aware IM supports long operations that can span hours, days and months. The state of a long running operation is saved in the database and is restored whenever the reply is received.
- Aware IM automatically generates forms based on the definitions of business objects and adjusts the forms when the definitions of objects change.
- Aware IM automatically generates web services based on the definitions of Aware IM services. Web services may be called by any external software system.
- Aware IM offers a rich set of API that allow programmers to plug in custom extensions written in the Java and Javascript programming languages in those circumstances when a certain feature of the target system cannot be implemented using the built-in capabilities.

Technical Architecture



Aware IM architecture diagram

Aware IM Server

Aware IM Server is at the heart of the IM architecture. It fulfills all external configuration and operational requests – saves and retrieves configuration information, executes processes, evaluates business rules, saves and retrieves operational data.

Data Server

In Aware IM the storage and retrieval of data (both configuration and operational – such as business spaces, business space versions, instances of business objects etc) is managed by a database management system. Aware IM can work with such popular SQL-compliant databases as MySQL, Microsoft SQL Server, Oracle and Cloudscape/Derby. The database management system is often deployed on a separate machine called the Data Server. The Aware IM Server communicates with the Data Server by sending data storage and retrieval requests via JDBC. It is also possible to deploy the Aware IM Server and the Data Server on the same machine or even embed the database management system into the Aware IM Server.

Web Server

Web Server is a client of the Aware IM Server. It handles HTTP requests from clients in the Operation Mode. The requests are sent from browsers over the Internet. The Web Server redirects the requests to the Aware IM Server and sends the replies back to the browser client in an HTML form. The Web Server runs within the servlets container such as Tomcat or Weblogic. The Web Server and the Aware IM Server can be deployed on a single machine or on separate machines.

Configuration Tool

The Configuration Tool is another client of the Aware IM Server. It offers a Graphical User Interface (based on Java Swing) that allows configuring Aware IM-based systems. Once a system has been configured the Configuration Tool sends the resulting configuration to the Aware IM Server, which stores it in the database. Just like the Web Server, the Configuration Tool and the Aware IM Server can be deployed on a single machine or on separate machines.

Channels

Channels connect the Aware IM Server with its clients. All requests to the Aware IM Server come through channels. There are a number of built-in channels that connect the Aware IM Server with some well-known clients. For example, the Configuration Tool and the Web Server are connected with the Aware IM Server through the built-in JMS (Java Messaging Service) channel. Other clients, such as external software systems or hardware devices can connect to the Aware IM Server via a built-in SOAP/Web services channel, that offers industry standard web services mechanism to communicate between systems. In addition, custom channels can be plugged into the Aware IM Server to connect with external software systems and hardware devices via non-standard mechanisms.

On the Aware IM Server side all channels plug into the Input/Output Interface, which provides the standard delivery mechanism of the external requests to the main engine of the server.

Features

The following features characterize Aware IM architecture:

Use of robust modern technologies
Aware IM is based on the plethora of Java technologies such as J2EE application server, JDBC, JMS, JSP/servlets technology, web services etc. These are robust technologies that have been proven in the industry.

Client-Server
Aware IM is a client-server system. Clients of the Aware IM Server can be end users running Internet browsers, configurators running the Configuration Tool, other software systems or devices.

Multi-user
Aware IM is a multi-user system. There are no limitations on a number of users both in the Operation Mode and in the Configuration Mode. The Aware IM Server manages registered users and guests and handles every client request in a separate thread.

Scalability
Aware IM architecture has been designed to be fully scalable. When a workload of the Aware IM Server increases additional machines running copies of the Aware IM Server can be deployed. The extra work can thus be offloaded to additional servers. It is also possible to cluster Web Servers and deploy one Web Server per machine.

Extendibility
Aware IM architecture has been designed to be very flexible. It offers a rich set of API that allows to easily plug-in new components, such as custom channels and processes.

Power
The architectural features described above make Aware IM a very powerful platform that can be used to solve a broad spectrum of data management tasks.