



White Paper

SynchSource
Open Framework Architecture
Four Part Harmony

I. Voices

Most modern transaction architectures separate presentation, business logic, and data access into layers with well-defined interfaces between each layer. Such architectures provide independence of function and afford strong reuse of components.

The Open Framework provides a four-tier server architecture. The Open Framework has the same two lower layers – Business Objects and Data Access – as are present in three-tiered architectures. However, the presentation tier is divided into a Process Management layer and rendering facility called the Foreplane. The Foreplane components interact with the Business Object layer to create renderings in a variety of modes: HTML, XML, SOAP, JSON, CSV and so on. A well-defined interface to the Business Object layer allows new Foreplane classes to be quickly constructed as interfaces, presentation facilities and standards evolve.

A user presentation can be thought of as a set of building blocks or Presentation Units. The Presentation Units are dynamically assembled into a User Presentation based on a variety of factors such as the user's authority when viewing a particular subject, the state of a process the user is involved with, the language and preferences of the user, the policies and preferences of the user's company and services or facilities available. Here, the term User Presentation refers not only to HTML messages for a user's browser, but also to structured messages in XML, JSON or other formats delivered to various user agents such as mobile devices or companion applications.

In the Open Framework, each User Presentation is part of a Process. A Process might be composed of a single User Presentation or a network of User Presentations that guide a user through a series of steps involving branching or alternate routes. A Process might lead a user to presentations constructed for the specific user's needs after progressive drill downs, selections or questions.

The Process Management Facility dynamically adapts a Process as a user interacts with the data. The Process Definitions are stored in database tables that represent a directed graph. A user is guided through a Process based on their authority, the customer configuration and the state of the process (State-based Workflow). The Process Management facility is composed of a Process Definition Tool, Process Definitions, Supervisory Services and a Workflow Enactment Service. This discussion pertains mainly to the Workflow Enactment Service.

A unique aspect of the Open Framework implementation of the Process Management Facility is the ability to have different customers operate with different Process Definitions. This is a requirement in a service operation where clients of various sizes and industries operate in a single database and code line.

The combination of a Process Management Facility that controls dynamic Presentation Units is ideal for business applications whether processes are simple or sophisticated.

Voices:

- Process Management Facility
- Foreplane Classes
- Business Objects
- Data Access Facility

II. A Familiar Melody

Consider a relatively simple business process such as a Merit Increase. This Process might be initiated by several individuals: a direct manager, a manager who is an executive or an HR Generalist.

A scenario:

Fred is preparing a Merit Increase for one of his subordinates, Sally. Fred believes that Sally is a key individual and a top performer. When Fred enters the Merit Increase Process, the system determines that his company has established salary ranges and increase guidelines. The system therefore presents the salary ranges for Sally's position, the range utilization calculation and the guidelines that apply.

Fred has available Sally's compensation history and performance history. Fred's company has established both organizational and personal objectives. Links are available that will open page sections for him to review that information.

Fred decides that Sally's performance is so good that he is going to try for a raise that exceeds the standard guidelines. The system informs him that this will require executive approval and shows him that Pamela is the line executive with that authority. He has a text window to write his argument for the increase and he can attach documentation.

When Pamela clicks in her in-box on the Merit Increase for Sally, the presentation shows that this is an exceptional case and she is presented with salary range and guideline analysis as well as the performance information for Fred and his organization. She also sees a graph of the range utilization for Fred's organization and comparable organizations. Satisfied with her review of the information, she approves the change.

When Roger, the HR Generalist, picks the transaction from his in-box, he reviews the material and decides to do a lateral range utilization analysis for Sally's grade and region. The graph shows that salaries are trending above the midpoint. Suspicious that the Western Region is out of synch with the market and reminded that his company has purchased a salary survey service, he selects a survey and reviews the data. He decides that a change is in order and prepares a change recommendation. He sends a note back to Pamela and Fred with his conclusions.

Even though this Process is composed of only a few steps – Entry Form, Approval Form and Confirmation – the Presentation Units prepared for each user were appropriate for their authority as it related to the subject. One size does not fit all. If Fred's company had not decided to use salary ranges and guidelines, such information would be incorrect or misleading. If the company had decided on another workflow model the transaction would have been routed differently. If the system had not reminded Roger that salary surveys were available, he might have missed an opportunity to make his company competitive for key talent. Had not the system let Fred know that he could propose an over guideline increase, he might have lost Sally and ended up spending much more than her increase to recruit a replacement.

Fred's Merit Increase form was dynamically constructed based on: his direct manager relationship to Sally, his companies decision that Sally's grade should be governed by ranges and guidelines, and the companies selection of a process model that provided for escalation.

III. Instruments

Just as one would not consider having a drum play a part scored for a woodwind, the Open Framework uses different tools for each of the four layers.

Process Management

Process Definitions are stored in a relational database. The Process Definition Tool is a suite of web pages that provide for both entry and analysis of processes. The Supervisory Services are web pages that expose the control information stored in the relational database and allow processes to be changed. Some Supervisory Services are primarily for administrative users, while portions such as re-routing and path selection are integrated into the broader user experience. The Workflow Enactment Service is Java code that is distributed into common components of the Foreplane and control classes. This service continuously monitors the transactions and provides dynamic adaptation of processes. The state of processes is maintained in the database.

Internal messaging is used to support processes that operate independently from user interactions. Such process as time-based Notifications, deferred messaging and mass processing such as Open Enrollment preparation are managed by a set of message queues and process tables.

Foreplane Classes

HTML generation, XML message preparation and the preparation of other communication formats such as JSON are done by Java classes. The Foreplane classes have a broad set of facilities available to them for transaction control, database access and process control, but the primary interface to the Business Objects is a dynamic interface. This dynamic interface is at the heart of the flexibility and adaptability of the Open Framework.

Because different Foreplane classes can work with any Business Object, once a Business Object is defined, it can be accessed by Web Browsers, other user agents such as mobile devices, AJAX facilities or intra-system messages.

Business Objects

Business Object construction uses an Object Factory Design Pattern that allows the system to dynamically build objects based on declarations in XML, compiled views and schema definitions from databases or web services.

Because the interface between the Foreplane Classes and Business Objects is dynamic, a Business Object can be adapted in real time to a user's authority, the company's options, the system configuration or software versions. The view of a Business Object can be different for an administrator than for an employee without having a different version of the code or complex logic built into each user interface component.

Business Objects are defined using a Declarative Programming paradigm. The declarations are done in XML. In Declarative Programming, one specifies what rather than how. A declaration is made, for example, that a particular constraint – one legal name for an employee – should be applied to the data. Constraints can be defined either as database views that return a prescribed number of rows or can be written in Java. The developer does not say when or how this constraint should be applied. Enforcement is left to the framework.

Declarative systems enforce consistency and eliminate side effects – a declaration will be applied in the same fashion for all objects. However, Declarative Programming implementations are limited by the vocabulary of the declarations. This limitation is addressed in the Open Framework by extending the Factory Design Pattern using Java subclasses.

These subclass delegates are determined at run-time to provide a dynamic extension capability not unlike what is found in dynamic languages like Ruby and Python. In this layer, procedural logic can be used to implement complex business logic, validation or post-processing.

Data Access Layer

The Data Access Layer in the Open Framework implements a Persistent Object-Relational Mapping (ORM) library that is functionally similar to ORMs such as Hibernate, iBATIS and the Java Persistence API. Like these frameworks, the TRDM library provides session construction, transaction control and data mapping. The TRDM Library fully implements the SynchSource Time Relational Data Model providing for managing different history types, corrections and proposals. Like other ORMs, the TRDM Library uses XML declarations to structure data. However, unlike most other ORMs, the TRDM Library implements a dynamic API and derives data definitions, in part, from the database schema. The library supports direct SQL as does iBATIS and also SQL generation like Hibernate. Data protection mechanisms such as service-based, multi-client controls span both access models.

The implementation of the Time Relational Data Model in the object persistence layer ensures that data is uniformly managed across the applications. The dynamic API allows the same code to run on different database versions with a single code base. But just as important, the dynamic API allows data objects to have different configurations for different users.

The dynamic mapping of the TRDM Library encourages programmers to develop database access in database views. In most online systems, performance is governed by the number of accesses to the database and the length of the queries. Having a programmer or ORM individually retrieve the components of a presentation rather than have the components assembled by a view or multiple joins on the database, tends to increase the number of accesses to the database. Storing a pre-compiled view reduces the query time. Having the database optimizer build a query plan rather than the programmer is the essential reason for the declarative nature of the SQL language.

The TRDM supports data facilities such as Web Services. In the networked environment, the dynamic interface has the advantage of being able to adjust to changes in the structure of networked data without programming.

IV. From Harmony to Symphony

In the simple three-step Merit Increase process above, there might easily be ten different ways to construct the Merit Increase and Approval presentations. The SynchSource application presently has 722 Processes, 2,389 Process Steps and over 8,800 Presentations. The number of each category could easily double as the application suite matures. Managing a highly flexible and dynamic application suite is the reason for the Open Framework.

Other than the integration of Process Management layer, a significant difference between the Open Framework and many other frameworks is the dynamic nature of the Business Objects. Rather than a fixed data layer that, in the case of the SynchSource application, would require managing over 10,000 units of program code (perhaps growing to twice or four times that number if multiple versions are required), the dynamic operation of the Business Objects greatly simplifies system management. A Business Object can support a variety of presentations and multiple versions without additional code.

Successful competition relies, in part, on how rapidly a product suite can evolve. Client companies change their focus over very short time frames. In the midst of a recession, the focus is on how to reduce staff without hurting competitiveness. As a recovery gets underway, the focus is on recruiting top talent and improving productivity. The SynchSource Open Framework and the applications built on that framework make service companies more competitive by providing an agile environment that can rapidly adjust to clients needs.

Related Material:

[Definition and comparison of serial and state-based workflows.](https://msmvps.com/blogs/theproblemsolver/archive/2006/10/11/Sequential-versus-State-workflows.aspx)

<https://msmvps.com/blogs/theproblemsolver/archive/2006/10/11/Sequential-versus-State-workflows.aspx>

[Workflow Management Coalition Specification](http://www.wfmc.org/standards/docs/tc003v11.pdf)

<http://www.wfmc.org/standards/docs/tc003v11.pdf>

[Object Factory Design Pattern](http://en.wikipedia.org/wiki/Factory_method_pattern)

http://en.wikipedia.org/wiki/Factory_method_pattern

[Declarative Programming](http://en.wikipedia.org/wiki/Declarative_programming)

http://en.wikipedia.org/wiki/Declarative_programming

[Comparison of Object Relational Mapping Systems](http://www.javaworld.com/javaworld/jw-07-2008/jw-07-orm-comparison.html)

<http://www.javaworld.com/javaworld/jw-07-2008/jw-07-orm-comparison.html>

[Discussion of Query Plans](http://en.wikipedia.org/wiki/Query_plan)

http://en.wikipedia.org/wiki/Query_plan

About SynchSource

SynchSource Inc. has developed a suite of open source HR, workforce management, benefits, and payroll solutions that are rapidly establishing the company as a leader in the service bureau software marketplace. SynchSource's fully integrated platform provides Human Resource Outsourcers (HROs), Professional Employment Organizations (PEOs), Payroll Outsourcers, Benefits Administration Outsourcers, and other service providers with a SaaS delivery model designed to empower their clients, accommodate unique requirements without customizations, reduce overall administrative support costs, and provide more value-added services at improved margins. The patented time-relational data model and the patented workflow and security engine are at the core of the SynchSource platform.

SynchSource Inc.
2201 Broadway, Suite #701
Oakland, CA 94612

Phone: +1 866-989-1145
email: info@synchsource.com
www.synchsource.com

