Read in Notes Mode

Enterprise Integration with Workflow Management

Fred A. Cummins

November 1,1999

1

Asynchronous messaging, using message brokers, has emerged as a commonly accepted approach to integration of independently developed systems, particularly ERP and legacy systems. The acceptance of this approach is evidenced by the emergence of products to support this form of integration, and was reinforced by presentations and discussions at the recently formed Enterprise Application Integration work group of the Object Management Group (OMG).

This presentation describes this approach and extends it to incorporate workflow management.

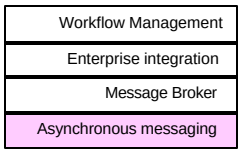
Enterprise Integration Architecture

Workflow Management	Business process execution, organization-based routing, work lists, monitoring
Loose-coupling integration	System adapters and business records transformation
Message Broker	Physical routing, tracking and resource management
Asynchronous messaging	Message queue management (store & forward)

This diagram depicts levels in an asynchronous messaging architecture to support enterprise application integration. The bottom layer provides basic asynchronous messaging such as that provided by IBM's MQ Series. I will discuss each of these levels in more detail.

Asynchronous Messaging

- Store-and-forward communication of a message
- Holds the message in a queue until the recipient formally accepts it
- Assures the message is recoverable and is received once and only once
- Provides a mechanism for communicating business transactions between systems



The bottom layer represents point-to-point, store-and-forward communication.

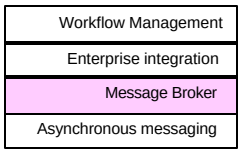
The sender of a message places a message in a queue for transmission. The messaging service communicates the message to a destination queue, and the receiver removes the message from the destination queue at a time determined by the receiver.

The messaging service provides assurance that the message is recoverable, and that it is received once and only once. Mechanisms at the sending and receiving queues provide for submission and acceptance of messages in a transactional context.

The net result is a mechanism for communicating business transactions between independent systems which can be applied between systems in an enterprise or between enterprises.

Message Broker

- Manage messaging from many sources to many recipients
- Route messages based on message parameters and recipient queue identification
- Provide a shared point of message distribution and control
- Manage network routing for performance and component failures
- Support programmatic subscription



Message brokers are at the next level of this architecture. A message broker receives messages from many sources, through point-to-point communications, and selectively forwards them to many destinations.

Messages are forwarded based on parameters and recipient message queue identities.

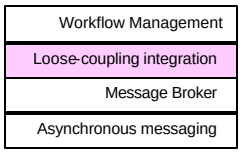
The message broker provides a shared point for message distribution an control. When a type of message is needed for a new purpose, the message broker can easily be updated to add the new destination with no effect on the source or sources. In addition, the sources of particular information can change without affecting the recipients.

The message broker also provides the potential for re-routing of messages to balance the workload for performance and respond to outages for uninterrupted service.

Finally, the message broker can provide a programmatic mechanism to subscribe as a recipient of particular message types. This can be used to support ad hoc reporting as well as various forms of analysis of system activity.

Loose-Coupling Integration

- Transformation of exchanged records to meet application requirements
- Shared tools and facilities for interfaces and transformation
- Routing based on record type and content
- Routing to physical queue locations based on logical addresses
- Ad hoc addition and removal of recipients of selected messages



The next level of the architecture incorporates transformation facilities. Such facilities apply rules to the transformation of a message from the format provided by a sender to a receiver's required format.

Generally, this consists of a shared transformation service to perform the transformations in a generalized way using easily expressed transformation rules. Transformations may be primarily conversions of record and data element formats, or they could also accomplish filtering by only forwarding a subset of the elements of a message type. Not all transformation activity may be accomplished within the transformation service; application adapters may be used to originate and receive messages in a specified format.

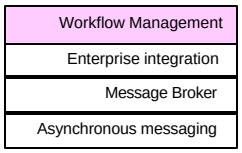
The transformation facility can expose or perform computations on selected elements of the records being exchanged to support selective routing.

Transformation rules can be used to route messages to logical destinations or distribution lists.

The facility can also support the ad hoc addition and removal of recipients of selected messages, particularly selection based on message content.

Workflow Management

- Automation of business processes for workflow among people and machines
- Routing rules driven by record content and organizational roles
- Multiple individuals serving multiple work lists
- Monitoring and control of individual business transactions



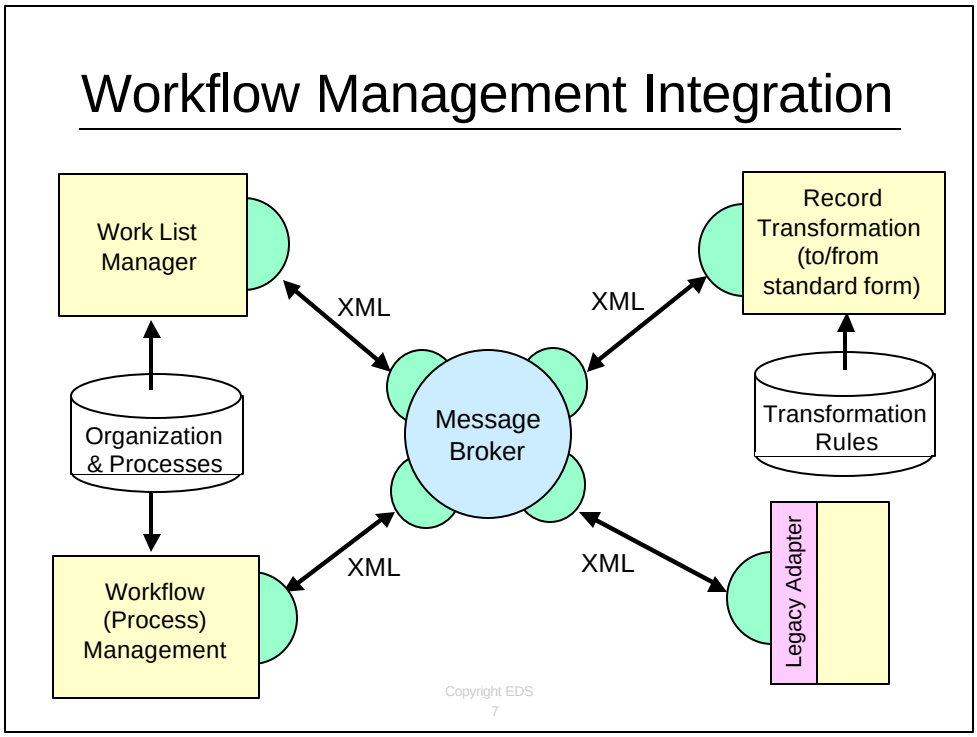
Workflow management adds important new capabilities to an asynchronous messaging architecture. The movement of messages between applications is essentially a primitive form of workflow. Workflow management can make this more visible and manageable.

Workflow management provides automation of business processes involving both computer applications and people. The processes are expressed in a form that can be interpreted and modified by business people.

Routing of workflow records can be based on record content, business rules and organizational relationships. This provides much more powerful and dynamic capabilities.

The desirable implementation would separate work list management from workflow management. This allows individuals to receive and act on requests in a single work list while the requests may be directed to the individual from several different workflow management systems.

The movement of work through the processes can be monitored and analyzed so that individual problems or process bottlenecks can be detected and corrective action can be taken.



This diagram illustrates the integration of the elements I have discussed.

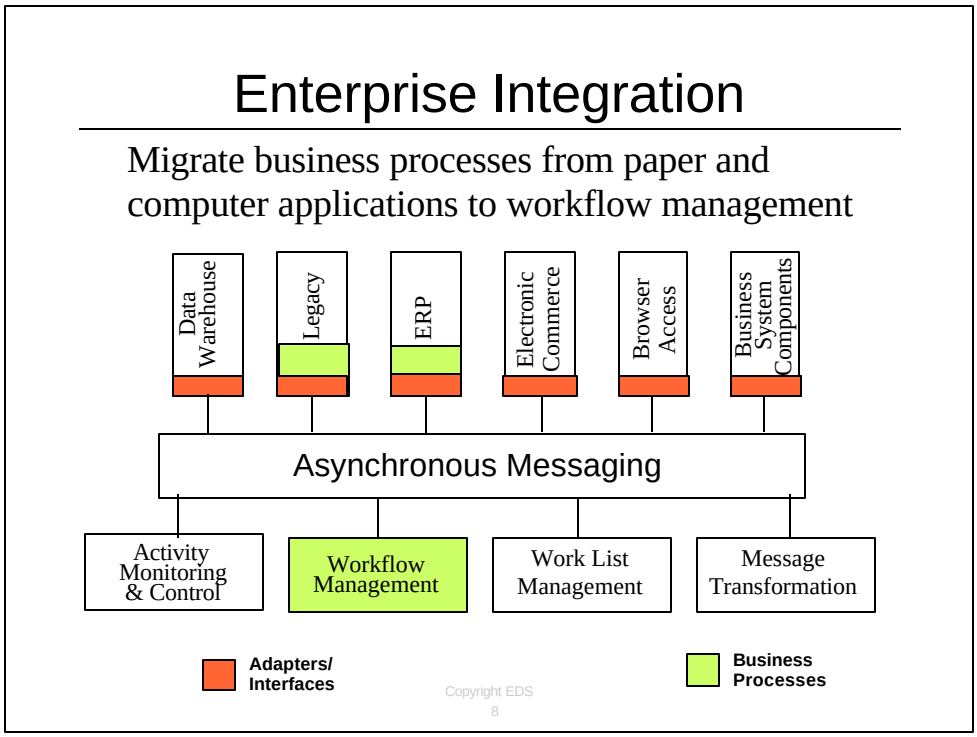
Asynchronous messaging is the basic “glue” that provides the point-to-point communication.

The message broker provides the many-to-many distribution of messages. In a large-scale system, there would be many, federated message brokers. There would also likely be gateways or specialized adapters to allow for incorporation of different message queue and message broker products.

In general, message content should be communicated in XML format to provide consistent, flexible access to message content. However, the architecture is capable of accommodating diverse formats with some penalty in flexibility.

Records would be routed to the record transformation facility. In general, there should be a standard form for each type of information and, as necessary, incoming messages should be converted to the standard form, and then, as necessary, converted to the form required by various recipients.

The workflow management system may be a recipient of records from other systems, causing the initiation of workflow, or it may route records to applications for processing. It would also route records to work lists for action by people.



Adoption of this architecture opens the door to an important opportunity for evolution of business applications.

Most current applications contain embedded business processes. These business processes should be migrated out of applications and into workflow management so that they are visible and can be monitored and changed to meet business needs.

As this migration occurs, the large, monolithic applications can be partitioned into smaller functional units, integrated with workflow management. This will enable a marketplace in these smaller applications, increasing competition and improving the flexibility of enterprise systems. New applications can be plugged into a specific functional area without creating a major upheaval and years of effort to transform related areas.

EDS

www.eds.com