Extending an Enterprise Integration Architecture: A Federated Approach using HP Process Manager.

Bill Martorano
Hewlett Packard Company
100 Mayfield Avenue MS 36U3
Mountain View, California 94043

(650) 691-7099
(916) 785-9278 fax

bill_martorano@hp.com

HP World Conference
August 2001

Contents

1 Abstract

Business Process Managers (BPM) are emerging in many middleware environments as a mechanism for defining and controlling complex business transactions. As integration brokers become a more prevalent technology for application and service integration, the capability for more effective and explicit process representation and automation is now recognized. Extensions to the business model beyond the typical boundaries of the application are also beginning to create a need for workflow tools with more powerful analytic and end-to-end process modeling capabilities. Extended capabilities, such as state management and explicit process definition lead to the need to link work management and process automation tools into the integration broker (middleware) environment.

Integration Brokers and Business Process Managers are two forms of middleware which when combined, provide extended capabilities to the business enterprise. Specifically, these tools may be used to alter the potential of application integration from simply managing transactions (the typical role of the application solution in today's world) to a more sophisticated perspective which supports a "Services Oriented Architecture". This perspective may involve definition and coordination of complex multi-step processes, which span application and enterprise boundaries. Successful exploitation of this type of capability to extend and enhance the business model, will allow businesses to gain competitive advantage in the E-Service marketplace.

A case study of HP's Enterprise Information Architecture (EIA) will describe use of HP's Process Manager (formerly HP Changengine) as a business process modeling and management tool. HP Process Manager has been tightly integrated into an asynchronous middleware environment based on BEA/eLink. The study will describe the architectural approach taken, also the empirical findings derived from use and experience of HP Process Manager in our mission critical enterprise integration broker environment.

2 Business Case

The predominant business case for integration of a Business Process Manager (BPM) into the Enterprise Information Architecture (EIA) was to enable HP businesses, to fundamentally *abstract*, *define* and ultimately *control* the services and processes existing outside the boundaries of the transactional application system domain.

Process definition has always been a difficult problem to model, given the mix of system and human interaction required. Typical process modeling at HP originated around the installation or creation of a new application system. While this model worked within the domain of the application system, it was not extensible beyond the technical and functional boundaries of the application system silo. Thus, over time, multiple (and uncoordinated) application system logical processes emerged in spaghetti fashion which when considered in the composite became quite complex to manage, or even to inventory!

Introduction of the Enterprise Information Architecture (EIA) began to enable the ability to model processes across application and service boundaries. As these intra-application and composite processes emerged, it quickly became apparent that workflow logic would be required to maintain the composite processes between systems. This workflow logic could be hard-coded directly into the middleware software layer. However, a hard-coding approach would effectively hide the logic of business process automation, as well as make it inflexible and difficult to change. Since lack of business process visibility was already a problem, this was clearly not the right approach.

It was also the desire of EIA's business sponsors, to create an environment in which business processes were *explicitly* defined and controlled. As a positive side effect, the approach of implementing a BPM would enforce a new degree of process rigor into the *development* of cross-functional, cross-organizational or extra-enterprise business models. Without specific modeling, automation and execution tools capability, the resulting processes would become a static inhibition to extension and expansion of these new business models.

Finally, it was recognized by business sponsors that middleware, as a "glue" between technology architectures, could be utilized to enable the integration and normalization of complex business processes in new ways, not just to overcome

existing business model bottlenecks and inconsistencies, but to also to provide positive new revenue opportunities.

3 Architectural Approach

Architecturally, EIA adopted a *federated* approach to middleware implementation. Philosophically, this was expressed by an architectural framework, which stressed a best-of-breed approach to the development and selection of software components. The principles behind this framework can be summarized as follows;

1. *Invest Strategicall*y for maximum business value.
2. Stress *Modularity and Reusability* of components.
3. Maintain an *Open Architecture*, by abstracting and shielding proprietary architectures, also by adopting industry standard approaches wherever possible.
4. Design for the *future*.
5. Build *resilience* into the architecture.
6. *Derive simplicity from complexity*.

These high-minded principles were taken into consideration when defining an implementation strategy to implement Business Process Management. Of primary consideration was the intersection of these values into an existing Integration Broker architecture, based on CORBA and BEA/eLink (Tuxedo) technologies. The Enterprise Information Architecture "*Integration Broker"* architecture currently implements the following legacy functionality.

- Queue Management (Message Oriented Middleware or "MOM") for asynchronous solutions.
- Connection management.
- Security management.
- Transaction management, both asynchronous and synchronous.
- Performance management.
- High-Availability

It was recognized early on that a **perfect** solution would probably not present itself in such a complex environment. Middleware technology is an emerging space. Most middleware products and vendors fall short in some form or fashion in terms of technological comprehensiveness. Also, it was acknowledged that

implementation of BPM within HP would intersect with an extremely complicated technical, semantic and operational environment. Because of HP's size and complexity, extreme diligence was taken to ensure that BPM technology did not become a future *legacy*, which might later impede future capabilities of the company. In particular, the "Plug-n-play" theme was approached with extreme caution. It was recognized that ease of implementation did not always translate to ease of support, or the extensibility of the solution.

Taking these elements into consideration, the EIA architect crafted the following design guidelines for developers. These guidelines recognized that *any particular implementation would express trade-offs, which* would need to be individually evaluated and balanced against the architectural framework.

3.1 Architectural Guidelines

- No Business Logic should reside in the Integration Broker technology.
- Functional or semantic translation should not occur within the Integration Broker.
- Granularity of events and meta-objects should map to the business model, NOT the application domain.
- Design up front to support a Request/Reply capability, even if Publish/Subscribe is the initial use model.
- Assume constant technological change (given the maturity of this technology).
- Design to be modular, replaceable, reusable!
- Think "City Planning" vs. “Application” architecture.
- Assume a federation of technologies will exist in any final solution, (not homogeneous implementations).
- Support a business-driven model, not a technology-driven model.
- Assume multiple users and multiple interface technologies.
- No one form of integration is "right" or "wrong."
- Design up-front for error handling.
- Producers/Consumer connections to the backbone should be able to be updated without disruption.

4 Implementation

Implementation of the Business Process Management (BPM) solution was accomplished by the installation of *HP Process Manager* (HPPM*),* as the controlling BPM. Integration of HP Process Manager to BEA/eLink was accomplished by purchase of an off-the-shelf **adapter** called *"Changengine Integrator"* (CE Integrator*).* A diagram representing the internals of this adapter design follows.

It is important to note that BEA/eLink is a composite product consisting of many software components. For purposes of integration with HPPM, *only the Tuxedo sub-component of eLink is required.* Coincidentally, only the Tuxedo sub-component was used in the EIA asynchronous solution, so this provided a smooth integration path.

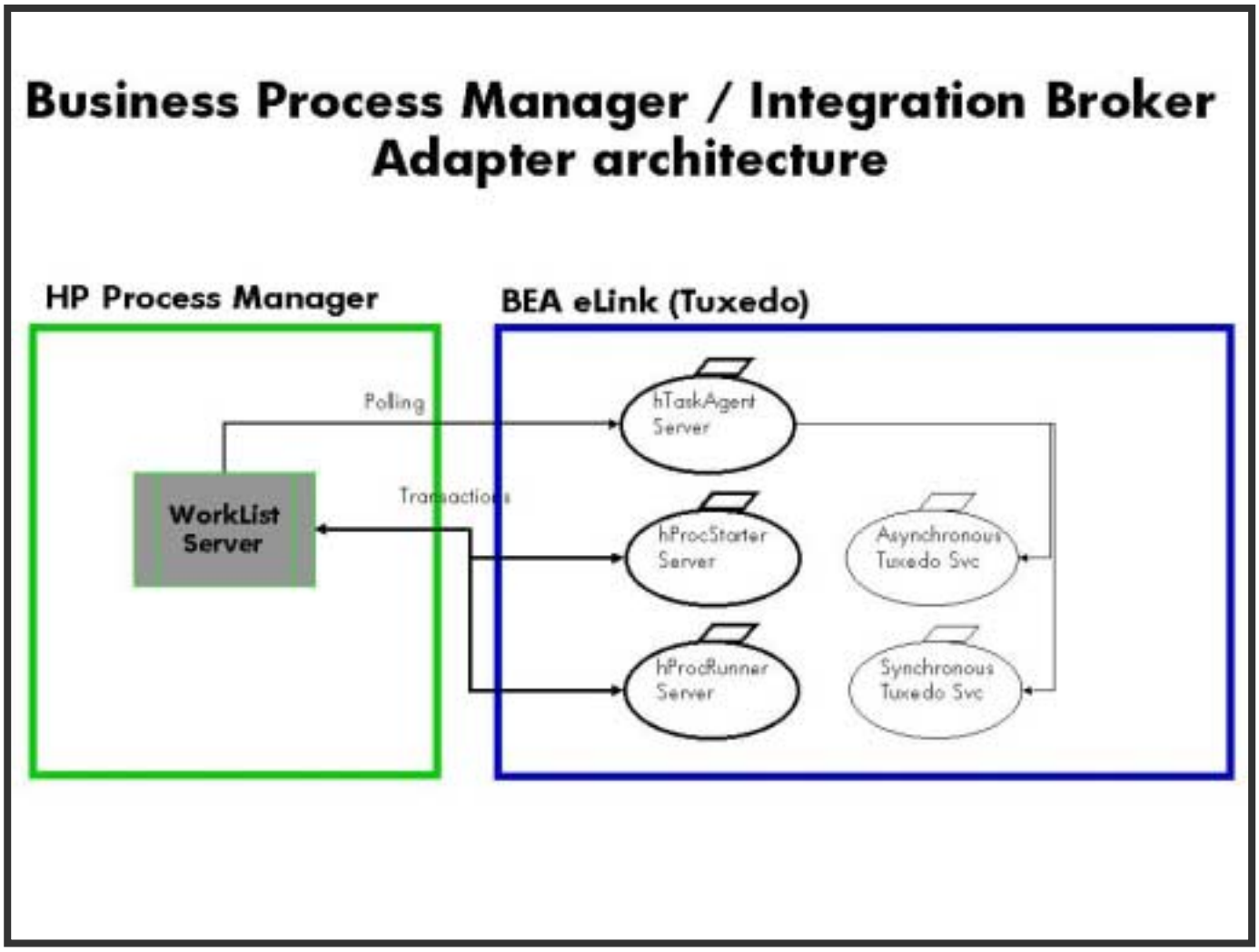
CE Integrator adapts to HPPM by means of exposing selected Tuxedo services, which are deployed on the BEA/eLink platform. These services function as invocation agent programs, which interact with the *Worklist Server* component of HPPM as follows;

- '**hTaskAgent**' allows HPPM to invoke **any** Tuxedo service (either synchronous or asynchronous) which is managed by BEA/eLink. This agent is invoked from within a task executing within the control of HPPM's Worklist Server.

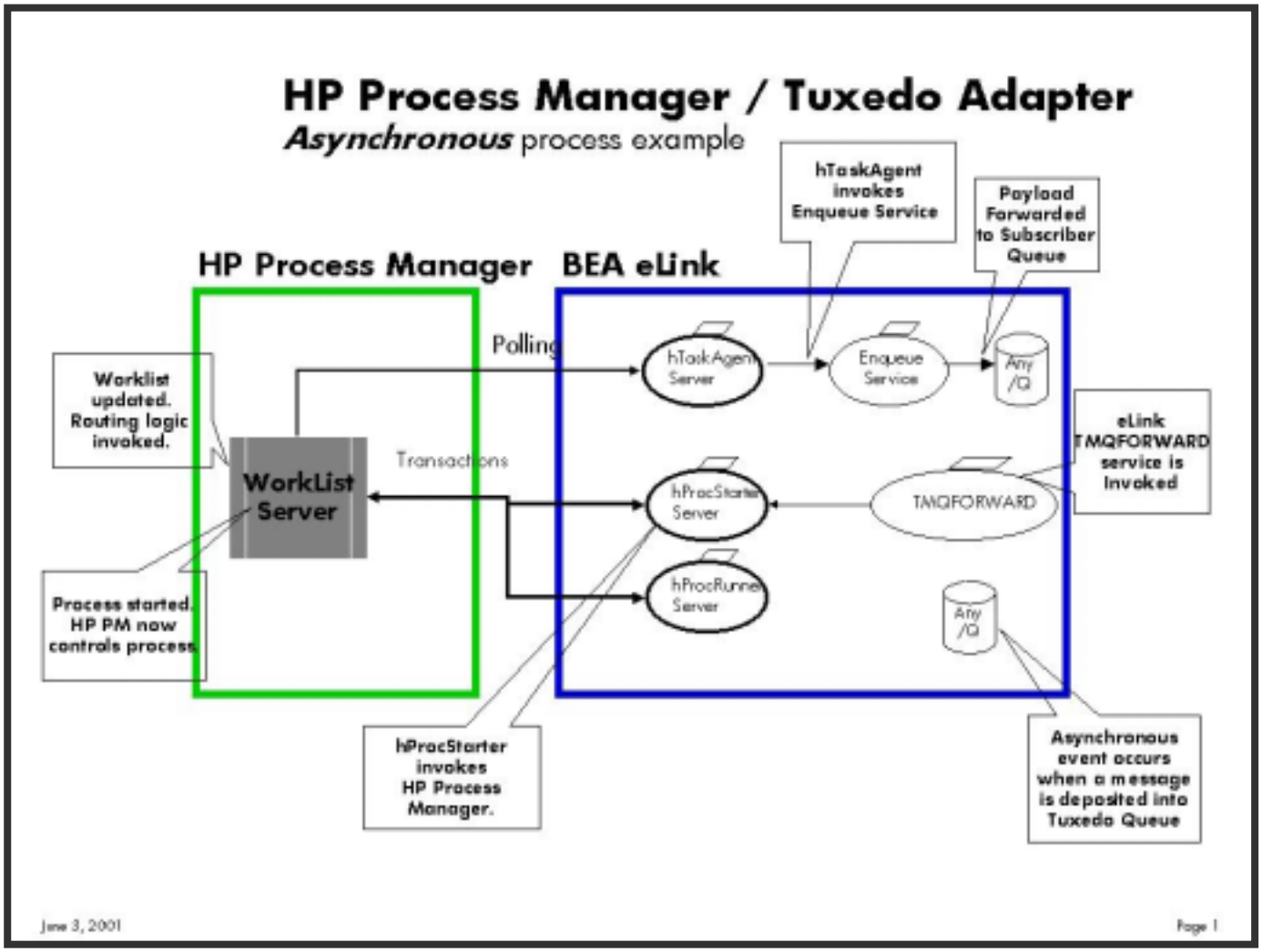
- '**hProcStarter**' allows any Tuxedo service to invoke and start an HPPM process in an *asynchronous* manner. Once the HPPM process is started, the Tuxedo agent does not wait (or block) for its completion in HPPM.

- '**hProcRunner**' allows any Tuxedo service to invoke and start an HPPM process in a *synchronous* manner. In this case the Tuxedo agent *will wait* (block) until the HPPM process has completed.

A pictorial view of this implementation is described.



In the EIA implementation of business process management, workflow was defined to control the routing of messages from producer application to consumer application; specifically, the passage of messages from incoming queue; to subscription queue. Within HPPM, process definition rules, including routing logic were defined; and executed using the 'hProcStarter' invocation mechanism. A pictorial description of this design follows.



5 Empirical Findings

Implementation of HPPM as a business process management tool was a success. However, performance considerations prohibited EIA from greatly expanding this architecture and solution design. Specifically, the CE Integrator (adapter) was identified as a performance bottleneck, so while process management was enabled, it performed too slow to keep up with the volume of messages passed through the Integration Broker. The use of HPPM was retained; however a scaled-back design approach was taken to "cache" routing logic into Tuxedo agents, rather than executing this logic for every transaction.

Encapsulating routing logic into Tuxedo agents violated the architectural principle of not encoding business logic into the Integration Broker. However poor, this was a necessary design tradeoff in order to use the Tuxedo adapter.

In the future, this design will be re-implemented against the HP Process Manager 5.0 architecture. Specifically, the adapter between HPPM and HP Bluestone (J2EE platform) will be leveraged. It is expected that transactional throughput performance will be much improved, and at some point this design will completely eclipse and replace the current Tuxedo adapter. HPPM 5.0 will also offer a native XML interface capacity, which coincides with the XML canonical data representation, used exclusively in EIA messages.

This re-design also corresponds to a new focus on web-service process control, which has recently become a priority for the EIA program. Thus, while routing logic is still a very important *internal* process to manage, it is clear that the real "bang for the buck" is in controlling higher level composite processes, which are deployed within the application server.

HPPM in isolation was a very reliable product. It offered a strong framework for business process modeling. By virtue of an open architecture, it also enabled various points of interface (e.g. http, API, CE Integrator to Tuxedo, soon J2EE).

6 Conclusion

Business process management, beyond being a "hot topic" of discussion in technical circles, adds real value to the business model. It is best positioned to target control of complex processes that span application and service domains. Unlike simpler

forms of automation, composite processes require a new degree of management, fail-over and recovery protection. Without automated business process management tools, engineering failover, high-availability and error recovery into these new and complex processes would be quite difficult.

State management across applications and services is another excellent role for a business process manager, especially as the control of state extends outside the boundaries of the internal enterprise and into the web services world. The use of automated business process modeling logically extends itself into the domain of *control of transactional conversations between business partners.*

Explicit automation of business processes is coming of age. EIA's early experiment into BPM has been viewed as a successful exercise in getting primal experience in this area. As could be expected with any new technology, there were bumps in the road; however the overall experience was positive and provided an important first-step experience, even if for the primary benefit of mainly technologists in this first implementation.