



Implementing SOA in a “Business-Critical” Payments Environment: Drs. Patrick Maes

About the Author:

Drs. Patrick Maes brings in over 15 years of IT experience in various areas of Investment Banking. As Principal - Transaction Banking & Investment Banking in HCL Technologies, Drs. Maes spearheads various initiatives in HCL related to SOA in payments and trade execution.

In his previous role, Drs. Maes was **Global Head of Derivatives Technology and Market Trust, Head of IT in Rabobank** and **CTO WIT Capital**. Drs. Maes holds a doctoral degree in Computer Science and a master’s degree in Artificial Intelligence and in Applied Economics from reputed universities.

About HCL:

About HCL Technologies

HCL Technologies is one of India's leading global IT Services companies, providing software-led IT solutions, remote infrastructure management services and BPO. Having made a foray into the global IT landscape in 1999 after its IPO, HCL Technologies focuses on Transformational Outsourcing, working with clients in areas that impact and re-define the core of their business. The company leverages an extensive global offshore infrastructure and its global network of offices in 17 countries to deliver solutions across select verticals including Financial Services, Retail & Consumer, Life Sciences & Healthcare, Hi-Tech & Manufacturing, Telecom and Media & Entertainment (M&E). For the quarter ended 31st March 2007, HCL Technologies, along with its subsidiaries had last twelve months (LTM) revenue of US \$ 1.27 billion and employed 40,149 professionals. For more information, please visit www.hcltech.com

About HCL Enterprise

HCL Enterprise is a leading Global Technology and IT enterprise that comprises two companies listed in India - HCL Technologies & HCL Infosystems. The 3-decade-old enterprise, founded in 1976, is one of India's original IT garage start-ups. Its range of offerings span Product Engineering, Custom & Package Applications, BPO, IT Infrastructure Services, IT Hardware, Systems Integration, and distribution of ICT products. The HCL team comprises approximately 45,000 professionals of diverse nationalities, who operate from 17 countries including 360 points of presence in India. HCL has global partnerships with several leading Fortune 1000 firms, including leading IT and Technology firms. For more information, please visit www.hcl.in

HCL America Inc,

100 Galleria Parkway, Suite 1545 – Atlanta,
GA 30339 Tel: 770-857-2625

Confidential Information

Information contained within this document is confidential and proprietary to HCL and should not be disclosed to anyone other than the recipients and reviewers of this document. However, in the event of award, this information may be disclosed to and will be used on behalf of and according to the interests of the recipients to whom it is addressed.



Implementing SOA in a “Business-Critical” Payments Environment

The Payments Market

The banking payments market is going through stormy times. There is the fast amount of regulatory changes happening in Europe and the US, requiring faster and more real-time payment processing. Competition is increasing by innovative payment offerings and new modes of business models, while clients are asking for reduced costs and strict SLA compliance. These trends are forcing specialization by means of global transaction banks and consolidation of execution by using outsourcing, in-sourcing and white labeling.

Why a single Payment Architecture?

In many banks payments were always perceived as a necessary extension of a primarily lender-oriented client relationship. This has changed and many corporate clients are nowadays measuring their banks performance on its transaction capabilities. Offering clients real-time and scheduled payments execution with advanced cash and liquidity management capabilities will be an important differentiator. The reality is often different. Current payment infrastructures are often composed of many components with weak point-to-point interfaces, a duplication of functionality and a silodriven approach. For instance, if a payment misses its cut-off time for a certain payment channel, there is no automatic functionality to re-route this payment to another channel, hence an unhappy customer.

Case: Faster Payments in the UK New regulatory requirements as e.g. Faster Payments in the UK require a fundamental re-think of the payments architecture. Target date for Faster Payments is November 2007 with May 2007 as date for inter-bank testing. The new payment system will allow clients to execute electronic transfers in real-time (with 15 seconds round-trip). This will replace a large amount of the current standing orders currently executed within the 3 days BACS cycle, where banks are making money by debiting the payer's account on day 1 and crediting the beneficiary account on day 3. Moving millions of payments from a current batch-oriented to a real-time architecture does require more than a clean-up exercise. And this is only the beginning, as SEPA and other initiatives are changing



the payments landscape dramatically. There are different approaches to solve this problem. Often banks create a Payment Council without having real authority over the different payment silos and budgets. This approach rarely results in a solution and often only can document the problem.

An alternative is to create a Single Payment Database. This gives added-value in management and reporting purposes, but does not reduce cost and duplication.

A third alternative is to use Middleware to create a Payment Hub. This will reduce the number of interfaces, but again does not reduce duplication of functionality, nor does it improve the current processes.

A last alternative is to implement a Single Payment Architecture. This approach will allow the bank to identify duplication and to define a step-by-step migration to common utilities based upon open standards. This needs a pragmatic approach that will split each payment process or silo in three main elements of functionality: origination, order management and payment execution.

By separating these three layers, it is possible to isolate the diverse set of origination channels (e.g. email, fax, EDI, Internet, Phone, etc.) from the payment execution and processing systems.

This enables the bank to introduce the following rationalizations and added-values early in the process without impacting current systems:

- Improve STP rates by validation on input, automated enrichment and repair, minimizing exception processing;
- Enable customers to repair & enrich by providing transaction visibility;
- Provide automated investigations capabilities;
- Standardize business processes around a common Payment Hub, enabling a shared Payment Service Centre irrespective of origination source, delivery channel and type of payment;
- Enable business operations managers to have visibility and full information of the status and performance of payments expected;
- Reduce risk and costs by removing duplicated functions, processes and legacy IT systems, replacing them with open systems and "best-of breed" solutions;
- Allow a greater proportion of future IT spend on innovation, rather than maintenance;

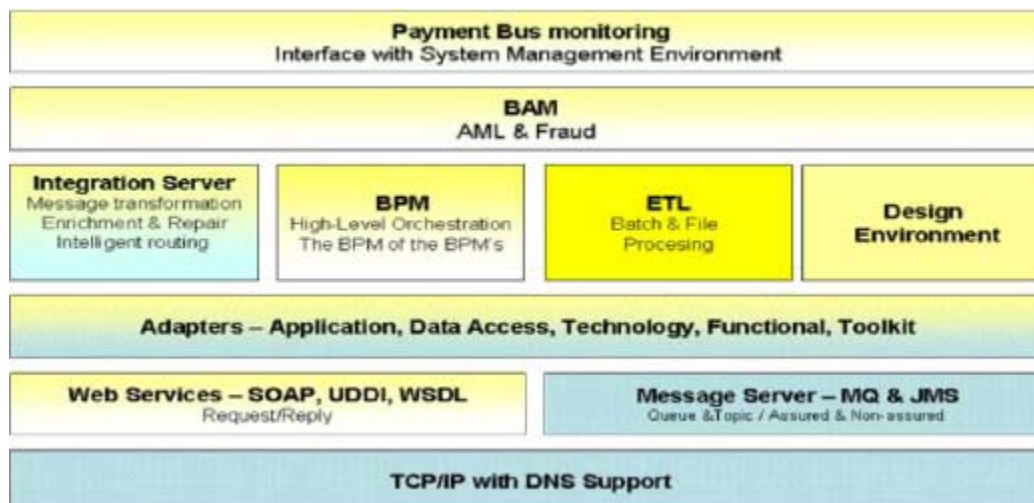


A crucial component is the Payment ESB (Enterprise Service Bus) based upon a SOA. By using a service oriented approach, we can de-couple and re-use existing application and middleware components in an unobtrusive way (the components are not aware of this), providing a low-risk transition from the current process and systems environments to the new architecture through a series of steps where each adds business value. This evolutionary approach should be complemented with a strong implementation methodology with an absolute focus on customer service and payment system resilience.

The payment Bus

As with all "new" technologies, there are many different definitions of ESB and SOA. ESB and SOA can be best defined as conceptual models that are aligned with the requirements of an overall Payment Bus.

The following picture illustrates the required components of a Payment Bus.



The basic layer is the Transport Layer where we define standard communication protocols. Above this we have two communication stacks: one is asynchronous using all the functionality of Message Oriented Middleware, the other support a synchronous paradigm using Web services as standard.

Application components are interfaced using adapter technology. The solution offers a rich set of pre-built adapters for interfacing applications, data storages,



transactional systems and external gateways. An adapter development toolkit enables us to build adapters efficiently.

The next layer of “functional” components comprises a number of servers. The Integration Server, containing the run-time environment of the ESB is responsible for all message transformation, repair & enrichment and routing.

The BPM Server offers business process execution, including human workflow. Finally, the ETL server, providing file and batch processing on pre-defined schedules. Above these components we have the BAM server (Business Activity Monitoring), which uses an event-based interface to capture relevant data from the bus and which can apply intelligent rules to publish monitoring info on a user-defined dashboard (e.g. for AML monitoring). All these elements need some design and configuration, so we expect a rich design and development environment.

The complete environment is managed via a central Bus Administration and Monitoring Server that provides load balancing, performance optimization, resilience, etc. using business (not system!) processes as input and having an interface to a, for example Tivoli environment (e.g. for alerting) for system management. All static data, business rules, messages & events, audit trails, logs and transactional data are stored in repositories using standard DBMS technology.

Most Banks already have a rich set of middleware components. Therefore it is best advised to use this conceptual model to validate the re-use of existing middleware and functional components.

The model should support multiple implementation technologies such as pure EAI solutions (SeeBeyond, webMethods, etc.), application server based solutions (e.g. WebSphere WAS and Message Broker), boutique solutions (e.g. CapeClear) and a hybrid approach where the ESB is built upon a set of best-of-breed solutions from multiple vendors.



ESB and SOA is not a message bus, an integration server, a BPM engine or a web service architecture, but a conceptual architecture where:

- Components implemented in different technologies, co-exist using a pre-defined set of adapters minimizing the development effort;
- Orchestration is done via multiple layers of BPM engines
- Business objects (e.g. payment instruction) have internally a standard (XML) representation and carry their own state (not only syntax, also semantics);
- Reliable messaging, transactional integrity and authenticated security is implemented across organizational boundaries;
- On-demand scalability and continuous availability exists.

I see the next phase in Service Oriented Architectures, where centralized workflow processing is replaced by "intelligent objects" acting upon facts and events. This architecture allows implementing complex business logic without having to invest in a centralized workflow component and consequently build new business functionality on existing architectures.