

## COLLABORATIVE BUSINESS PROCESS MANAGEMENT THROUGH HARMONIZED MESSAGING<sup>1</sup>

**Shazia Sadiq, Maria Orlowska**  
School of Information Technology and Electrical Engineering  
The University of Queensland, AUSTRALIA  
{shazia, maria}@itee.uq.edu.au

**Wasim Sadiq, Karsten Shulz**  
SAP Research Centre Brisbane, AUSTRALIA  
{wasim.sad, ka.shulz}@sap.com

---

**Abstract:** *Workflow systems in specific and Business Process Management Technologies in general, have significantly contributed to overcoming some of the integration problems of intra and inter enterprise process control and monitoring. However, the complexity of the interactions between heterogeneous and autonomous systems within the enterprise and often between trading partners in ever increasing. A number of initiatives and proposals are underway to provide solutions for process specification and communication. However, the focus is often on defining APIs and interfaces rather than the semantics of the underlying interactions. On the other hand messaging technologies are well positioned as the key enabling technology for facilitating these interactions. Although messaging systems have traditionally had a different agenda dominated by scalability and performance, we see a great potential in the enhancement of current messaging infrastructure, in its new role in facilitating complex, long running interactions for dynamic and collaborative processes operating in decentralized environments like the web. In this paper, we primarily present a vision for a technology aimed at providing a level of harmonization to multiple messages to form a single custom definable backbone. We will provide the foundation framework for the harmonized messaging technology and identify fundamental issues for the management of such complex interactions.*

**Keywords:** *business process management, workflow systems, messaging middleware, enterprise application integration.*

---

---

<sup>1</sup> A preliminary version of this paper appeared in *Proceedings of the 6th International Conference on Enterprise Information Systems, ICEIS2004, Porto, Portugal, April 4 -17, 2004.*

## 1. INTRODUCTION

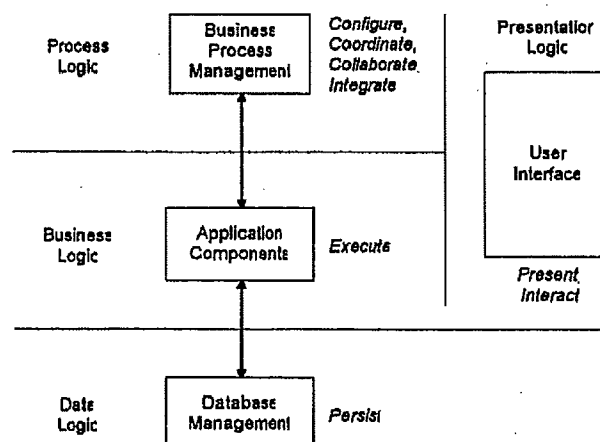
The last two decades have witnessed a tremendous growth in data storing, processing and communication capabilities. This growth has made a profound impact on how organizations deal with day-to-day operations and develop future business strategies. Commonly available networking and expansion of the access to the Internet have changed the way we reason about system architectures, with integration becoming an obvious and preferred option.

The research efforts and development paths pursued by many academic groups and industry vendors, targeting heterogeneous system integration, have not been easy and have not always delivered effective and practical results which could make a real impact on how the future solutions are to be constructed. Many lessons have been learnt from these research outcomes. They outline the clear boundaries of feasibility when dealing with building new applications out of existing and useful/deployable components (Colomb & Orlowska, 1994). These conclusions are not only related to the technological aspects of integrated structures, such as the middleware, but also to semantic issues of terms used across multiple systems. In particular, the need for a complete and extensible ontology that expresses the basic concepts that are common across a variety of domains, became apparent, forming a new research direction over the last few years (see e.g. the obase series of conferences)

Business process management technologies are considered as one of the key success stories in providing process control and addressing complex integration requirements. However, the expectation of what this technology must deliver is a moving target. What was true for workflow systems is no longer acceptable in the dynamic and cross organizational requirements for management of collaborative processes. Whereas the success of coordinative processes depends upon the conformance to the prescribed control flow, the success of the collaborative process depends upon the ability to detect and react to changing conditions. Our focus in this paper is primarily on the collaborative process. The complexities of managing the collaborative process especially becomes apparent in the B2B context. However, to position the discussion appropriately we first present some background on process enablement, and its evolving role in enterprise systems.

The remaining paper is structured as follows. In section 2, we will present the proposal for the Harmonized Messaging Technology, discussing the essential requirements, foundation principles, and the technology framework. Section 3 discusses related work, and section 4 will present the summary and outlook for this work.

### 1.1 PROCESS ENABLED ENTERPRISE SYSTEMS



**Figure 1: Building Blocks of Process-Enabled Enterprise Systems**

In Figure 1, we show building blocks of process-enabled enterprise systems. Just as the DBMS provided a means of abstracting application logic from data logic, the WFMS provided a means of abstracting coordinative process logic from application logic.

Every generation has provided additional functionality through supporting systems. Workflows Management Systems (WFMS) delivered effectively in the area of process enforcement, offering a clear separation of business process logic from component applications involved in process execution, thereby responding to the well-established need for application integration. It is an observed phenomenon that a new IT solution often triggers additional, and even more advanced user requirements, which probably would not be discovered if the current systems functionality would not be so widely available. This pattern can be clearly observed in the context of workflows technology evolution.

Although, workflow technology has delivered a great deal of productivity improvements, it has been mainly for pre-defined static and repetitive business processes, that required basic level of coordination between human performers and some application components. More recently business process management (BPM) has been used as a broader term to reflect the fact that a business process may or may not involve human participants and may also cross organizational boundaries through messaging infrastructures. The role of business process management systems must now be extended to provide additional functionality to support configurable, coordinative and collaborative processes and a much more sophisticated level of integration. This need arises from expanding business requirements for cross-organizational process communication. Examples of such processes can be found:

- in globalization of many manufacturing companies where different product parts are developed at different locations by different organizations,
- in a new wave of e-commerce applications where a great deal of outsourcing is the norm,
- in financial services with emerging subsidiary agencies sharing work practice,
- and recently, in new non-traditional application domains for business process technology such as e-learning, where cross-organizational units offer new educational services that would greatly benefit from integration.
- Only an integration technology that offers rapid and easy integration procedures, requiring only minimal IT expert intervention, can be successful at multiple and diverse, geographically spread e-business environments. The great challenge for IT specialists now is to find a functionally rich and technically feasible balanced solution for this overall complex problem of integration taking into account technological and semantic limitations.

## *1.2. ROLE OF MESSAGING*

Collaborative business process technologies are firmly positioned as an industry hot spot due to the increasing demands from the business sector for effective management of outsourced business activities and ability to control cross-enterprise processes. It is well known that this demand brings with it complex integration requirements that span interoperability across multi-platform systems, to semantic differences in business terminology.

Our vision of a new integration platform that could provide universal sub-process connectivity has roots in principles of messaging systems. Messaging infrastructures

provide advantages over method call based integration due to their ability to provide loose coupling.

Message oriented middleware is known to tackle some key issues of cross enterprise data exchange, without violating individual system autonomy. We explain this further by introducing a BPM architecture that utilizes the concept of a **BPM Object**. In this architecture, application components are exposed as BPM objects, which may have public method interface and/or messaging interface. This concept allows us to use the same BPM Object as an interaction bridge between application components and BPM technologies whether we want to make method calls to application components or let messages derive the interaction.

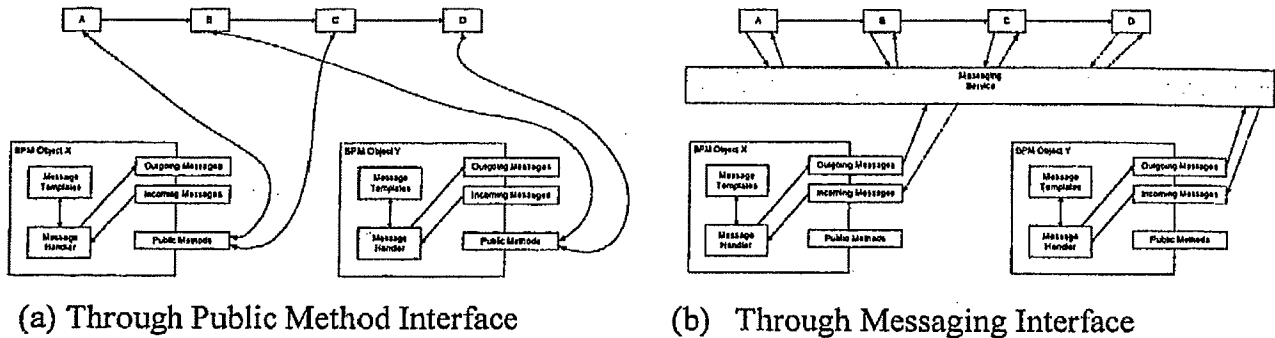


Figure 2: Invocation of BPM Objects

As with any common representation model to which multiple systems conform, communication through a common messaging service to multiple application components reduces the need for dealing with multiple public method calls (Figure 2). Recent developments in Service-Oriented Architectures (SOA) also build upon message based service interactions.

In this paper, we assume the availability of a messaging service as described above for BPM objects, to establish communication between application components. It is irrespective of whether the communication takes place between application components within the same organization (A2A) or across organizations (B2B).

However, the messaging features that are imperative to the success of private and public process communication in the web context are somewhat lacking in current messaging technologies. We aim to extend and integrate messaging paradigms and use them as key enabling technology for business process management.

## 2. HARMONIZED MESSAGING TECHNOLOGY

The provision of a level of harmonization to multiple messages often originating from different sources can form a single custom definable backbone of newly formed message streams. Such a technology could present a new and simple way of enterprise application integration/communication with substantial degree of outsourcing capabilities, B2B connectivity and collaborative business process management. We call this next generation of messaging: "Harmonized Messaging Technology" (HMT).

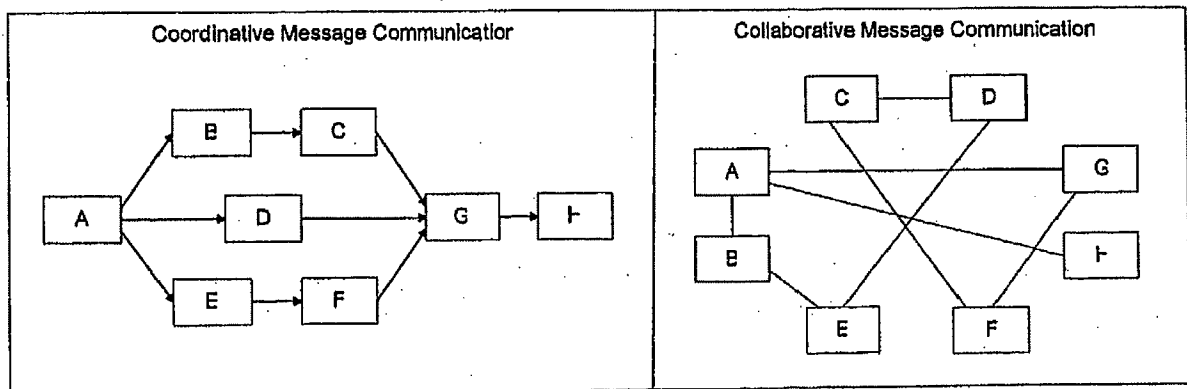
This research problem, as any integration problem where heterogeneous sources provide components, is very challenging. Any solutions offered must not only deal with data integration, but also structure of hidden sub-processes, providing interfaces to external partners without violating privacy or security rules and at the same time offering new functionality to all involved.

Even shallow inspection of the problem indicates some serious difficulty. To better position this complex problem, we base our framework on a set of assumptions that define the scope of the problem addressed in this paper.

- Our first assumption is that business process activities are mostly automated as is typical of B2B environments and web service based architectures.
- Our second assumption is that we are aware about the existence of the components. The issue of dynamic search and discovery of services is beyond the scope of this paper.

Under these assumptions and as further motivation for HMT consider a simple scenario where multiple business partners are engaged in a common process. A merchant organization places orders to two separate manufacturers. Order delivery by shipment partner needs to be synchronized within and between the two orders. That is, shipment is to take place not only when the entire quantity of one order has arrived, but is to wait for the arrival of the second order items as well. Furthermore a number of exceptions may arise dynamically, such as cancellation of some or all items of an order, delays in agreed delivery time due to transportation factors, etc. Orders and shipment requests can be seen as electronic messages routed through a messaging middleware. However, current messaging functionality (publish/subscribe, point-to-point) does not directly meet such advanced, yet much needed requirements. Instead these requirements are catered by direct communication between application components wherein, the process logic is embedded thereby compromising the process level control and monitoring.

WFMSs provide an effective means of coordinating business activities with well-defined dependency relations (sequence, choice, fork etc). We can use underlying coordination principles of WFMSs to satisfy coordinative communication requirements. However, workflow systems do not possess effective means to deal with collaborative form of message communication.



**Figure 3: Coordinative and Collaborative Message Communication Approaches**

The difference between coordinative and collaborative messaging requirements is depicted in Figure 3. In coordinative messaging paradigms the order of message communication between participants is defined using workflow like constructs. In collaborative communication paradigm the exchange of messages between participants is depicted by linking them up with communication channels. We propose to merge coordinative and collaborative messaging paradigms to effectively satisfy and manage a wide variety of complex business process requirements. We believe that this merger will serve the essential requirements of current business process management systems.

## **2.1. ASPECTS OF MESSAGING**

In this section we will establish the need and motivation behind harmonized messaging by presenting a number of cases where specialized messaging functions are required. These cases are intended to present a definition of what is meant by *harmonization*. Some of these cases can be (partially) met by existing messaging and workflow solutions. However, it will become clear that achieving a combined and extended functionality, within a single technology, is what current business interactions demand, and thus constitutes the objective of harmonized messaging.

There are several aspects of messaging which impact on, and define the scope for message harmonization. We identify below seven aspects of harmonized messaging, as a list of our minimum requirements. However, there is no restriction on extending this list, if business semantics warrant. The power of the proposed technology lies in providing a *generic specification* mechanism for the rules and constraints which describe these interactions.

### **Coordination**

Messages often represent a step in a business transaction or process. Coordinating the flow of messages can take the form of most, if not all, activity coordination structures in workflow/process management. HMT can facilitate coordination through multi-step complex routing specifications. For example:

- Wait for message from A and B to arrive before sending to C.

### **Temporal**

Temporal constraints represent a critical aspect of business events. Time driven messages may depend on absolute time e.g. 2.00 PM on Friday, as well as relative time e.g. every 4 hours. Example of time driven messaging can be:

- Keep collecting messages from A and B until a specific time and then send them to C.
- Wait to send message to B until 3 hours after receiving message from A.

### **Correlation**

Messages from a single (or even multiple) senders may be linked in terms of the content they carry. Correlation can include associating or relating a new message with a previously received message, for example:

- Associating or relating a new message with a previously received message, for example multiple items of a single purchase order.
- Invalidating a previously received message.

### **Batching**

The need for batching messages is clear from the above. Batching or grouping may be required due to message coordination, correlation or time dependencies. The definition of the batch may thus encompass many properties, for example:

- Deliver all messages on a given topic from a given sender at a given time, rather than one a time as they arrive to the message server.

### **Filtering**

This is essentially sending messages to interested parties based on message contents (content based routing). However, advanced filtering may be required, which takes into consideration a combination of conditions such as content, time, sender and others. For example

- Send a message to either B or C depending on contextual condition.

### **Transformation**

Message transformation may be required for conformance to formatting restrictions, or for ensuring that recipients are sent relevant data only. For example

- Extract essential data on date/time for a shipment order and send a FYI message to the customer

### **Composition**

Composition basically entails extraction of relevant data from one or more incoming messages and composing together a system generated message. The aspect of harmonized messaging is also illustrated further in Figure 3. For example

- Extract relevant data from A, B and C respectively and compose message D

Obviously the above functions are to be provided over and above traditional messaging functions like queuing and publishing/subscribing. Furthermore, it is important to note that coordination requirements are but a part of the overall set of requirements. Thus, where typical workflow modelling constructs such as sequence and synchronization may be applicable, they cannot satisfy the larger scope of requirements necessary for harmonized messages. Interestingly, workflow constructs can then be considered as a special case. A language to support the specification of harmonization requirements must additionally at least provide time related, content dependent and existential conditions. Considerations into the expressability and limitations of such a language, is a major research challenge, and remains an open question at this time.

## **2.2. FOUNDATION PRINCIPLE**

We base the foundation of HMT on a very simply principle, which we name **REPS**. Essentially, this consists of 4 steps of (R)ecieve, (E)valuate, (P)repare and (S)end.

**Receive:** This step is a combination of receiving, storing and logging incoming messages from external clients. This is a straight forward step for which several technology solutions already exist. In summary, the system must have the capability to recognize an incoming message, react appropriately to erroneous (unrecognized, corrupted, unauthorized) messages, and provide an appropriate means of persistent storage (required due to the long duration of the processes to which these messages belong), as well as log the message arrival event.

**Evaluate:** This is the most interesting step, as it requires the system to react on the basis of one or more incoming message based on predefined conditions on the content, time, and existence/non-existence of messages. Designing a specification language to cater for these diverse conditions is indeed an interesting research question.

**Prepare:** This step relates to the composition of outgoing messages. Basically, it requires the ability to define functions on the data contained in one or more incoming messages, the ability to extract that data, and populate an outgoing message.

**Send:** The last step of sending the outgoing message, is simply the flip side of the first step of receiving the message. Again there are existing technologies that provide this functionality and thus does not form the core focus of this work.

### 2.3. TECHNOLOGY FRAMEWORK

In this section we will introduce the building blocks for the technology framework. Basically, a system that provides harmonized messaging must have at least the following components:

#### Collaboration Space

The most fundamental concept in HMT is that of a *Collaboration Space*. The concept of a collaboration space is similar to a database space, where users, privileges, data and methods may be defined. In HMT, the collaboration space will similarly contain information on three key aspects:

**Participants** specify the parties involved in a specific message interaction definition.

**Rules and constraints** model the harmonization requirements using the specification language.

**Message Types** define the structure to which Messages being exchanged must conform to.

At any given time there would be several messages of one or more defined message type being exchanged by multiple participants within a collaboration space. We call these *message objects*. In simple terms, message objects that arrive in the system may or may not trigger predefined rules. If and when a rule evaluates to true, subsequent (system generated) messages are composed and dispatched. The figure below illustrates the concept of message harmonization for *Composition* of messages as given in the previous section.

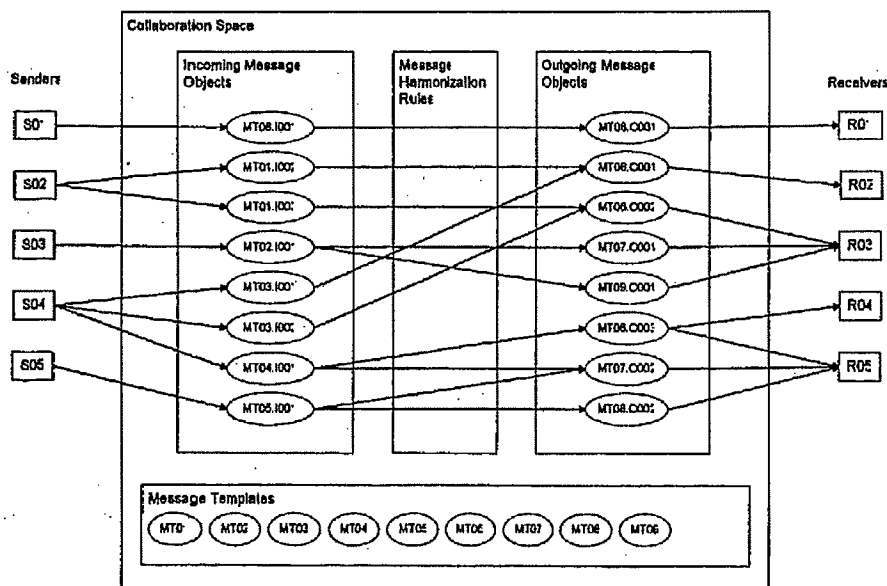


Figure 4: Message Harmonization in a Collaboration Space



Note that not all aspects of harmonized messaging can be graphically shown, as they may involve expressions with content and time. Thus the real power of the technology resides in the specification of complex rules that capture the behaviour of business interactions.

### **Interaction Modeller**

The interaction modeller provides a toolset to establish collaboration spaces, identify the associated participants and message types, and in particular to model rules and constraints needed by the collaboration space to support exchange of messages. For the latter to effectively take place, the interaction modeller must be equipped with a language to express the conditions that govern the harmonization requirements, which as mentioned above will range from basic messaging models such as queuing, publishing and subscribing, to typical coordination structures as found in workflow models, to complex expressions for correlation and batching, the ability to understand expressions with time and content, and the ability to transform and create new messages.

### **Harmonization Engine**

The Harmonization Engine is the core driving force for the system supporting essential functionality. This engine is primarily responsible for managing the collaboration spaces defined within the HMMS. Minimal features of the engine will include: Access to a *persistent storage* facility (HMT databases identified below); Management of *concurrent users* building message streams; *Exception handler* dealing with unexpected behaviours; and *Transactionability* in order to offer guarantee of completeness of execution.

### **HMT Databases**

The HMT framework will be supported by a number of underlying databases essential for the persistent storage and data management of various aspects. These consist of: Message Store - Message store allows reliable and persistent of messages. A relational DBMS can serve the purpose. The messages need to be stored in the message store while in transition from one participant to another and/or awaiting harmonization rules to be triggered.

### **Participants Repository**

To store information about participants, processes and other objects that need to exchange messages. It will also store participant profiles (including at least registration information, privileges, and requirements).

### **Rules and Constraints**

The harmonization engine needs rules and constraints within a collaboration space to effectively manage and route the messages. The rules and constraints repository maintains this information both for design time and run-time.

### **Message Catalogue**

Similar to a system catalogue in a DBMS, the message catalog stores type definitions for message objects.

### **System Log**

Will be required to record all system events to provide reliability and transactionability.

## **HMT Gateway**

The messaging gateway provides interfaces to participants to register and connect to the infrastructure and to send messages to and receive messages from the collaboration space. A dispatcher sub-component will evaluate identification and routing information to correctly route the messages within the collaboration spaces.

## **3. RELATED WORK**

Businesses are increasingly moving towards extensive automation of their private and public processes. This automation takes the form of complex interactions between heterogeneous and autonomous systems within the enterprise and often across multiple organizations. Controlling these complex interactions in order to effectively manage collaborative business processes is known to be a critical yet difficult problem using current technology solutions. Consequently, the areas of consideration are multi-faceted ranging from security, reliability and transactionability, quality of service guarantees, process validation, optimisation to semantic integrity of terminology used. The industry is currently flooded with initiatives and proposals towards e-business standards, which has made a systematic study of this area increasingly hard. These standards encompass trading partner agreements, business process specification, application integration, and network protocols.

In this section, we have attempted to identify some of the major related technologies and research directions. Where as a comprehensive survey of these technologies cannot be accommodated, we present below a short description of what we believe to be the technologies that will bear relevance to the concepts discussed in this paper.

Integration technologies such as brokers, application adapters, portals and messaging are fundamental elements of a collaborative business process environment. For this widespread enterprise application integration (EAI) and/or business to business (B2B) integration to become a reality, we need common architectures and open standards to support it. **B2B protocols** attempt to establish a common language between businesses, so that collaborations (which occur between two business partners) can take place without the need for pair-wise negotiation of integration. Such protocols are message centric by definition, describing the formal message exchange necessary for an interaction to take place between two business partners. B2B protocols have been an active area of research (Bussler, 2002) with two of the predominant solutions in this area being RosettaNet and ebXML.

An related area of active research for enabling automated inter-business interaction, and facilitating system integration is the utilization of **web service technology** ([www.webservices.org](http://www.webservices.org)). Web service technology's potential in the area of integration and interoperation has generated great interest, with initiatives from leading software vendors such as Hewlett-Packard, IBM, Microsoft, SAP, Oracle and Sun Microsystems. Web services are seen as a means of integrating applications, promoting interoperability and facilitating process management over decentralized environments. The loose coupling and dynamic binding characteristics of web services are the main justifications towards achieving the above.

The web service architecture is described by a Web Services Stack (Kreger, 2001), however the most appropriate stack structure remains a debated issue, with a number of alternative architectures offered by various consortiums and leading vendors. Despite this disagreement, moves have been made towards standardization, with a general consensus existing concerning the underlying protocols necessary in the architecture such as the Web Services Definition Language (WSDL) Universal Discovery Description (UDDI) and Simple Object Access Protocol (SOAP). WSDL, UDDI and SOAP however, are not alone

enough to facilitate complex and meaningful interactions with and between web services, which would allow private and public processes to harness the full potential of this technology. Currently, many organizations are attempting to address this problem, with proposals intended to extend the basic web service functionality primarily at the level which is often referred to as the orchestration or choreography layer of the web services stack (Uldell, 2002). These extensions are aimed at capturing more meaningful semantics than simply service invocations, enabling the modelling and implementation of business processes in the web service context. Prominent initiatives in this area include WSCI, and BPEL4WS.

An essential component of the next generation of distributed architectures is **message oriented middleware (MOM)**. MOM provides the basic means for target applications to communicate in a distributed environment. Messaging middleware however is not a new technology.

The past decade's move from client/server to Web applications has intensified the need to move information in real-time between disparate systems and in a more controllable manner. In response to these new developments, a set of Web-native asynchronous messaging technologies has emerged to take over where their legacy predecessors fall short. These include products based on standard implementations such as the Java Messaging Service (JMS) or those that span multiple standards and platforms.

In its new role, MOM has gained increasing deployment and has already delivered great benefits for communication between disparate systems, and as a grass roots component of the web services stack. In spite of the move from propriety networks to open standards, the fundamental functionality of MOM has not changed substantially. Looking at currently available solutions, we see that the focus of MOM has been primarily to deliver Security (authorization, digital signatures, non-repudiation); Reliability and Serializability (guaranteed delivery in the proper order); and Scalability (high volume and speed). The technology is driven by mainly two dispatch models.

One is point to point, where message exchange takes place between a sender and one recipient. This is often based on queuing methods, such as the IBM's WebSphere MQ series. A second dispatch model is publish-subscribe, which is used for content dissemination to multiple recipients or subscribers. Some essential enhancements to basic messaging technology have been proposed, for example in content-based routing which provides a dynamic model, using the contents of the message to filter messages to appropriate subscribers, see e.g. Elvin (Arnold & Segall, 1997), Gryphon (Strom et al, 1998), READY (Gruber et al, 1999).

The research on **workflow languages** provides both the foundation, and occasionally the bias towards meeting advanced process requirements. Several aspects of a workflow model including control flow, data flow, participant assignment, exception handling, temporal constraints, transactional, messaging have been studied (Jablonki and Bussler, 1996). Control flow is considered to be the foundation for capturing other aspects and as such, control flow modeling and verification has been an active area of research (Sadiq and Orłowska, 2000). Efforts to identify advanced workflow constructs to address specific process requirements have been undertaken (Aalst et al. 2003).

In the area of collaborative business processes, the work is generally presented in the context of extending web service functionality. These extensions are aimed at capturing more meaningful semantics than simply service invocations, enabling the modelling and implementation of business processes in the web service context. Prominent initiatives in this area include WSCI, and BPEL4WS.

## 4. SUMMARY AND OUTLOOK

HMT is intended to facilitate complex business process interactions, typically found in current e-business environments. Currently, there is substantial interest in the industry from vendors, standards bodies, as well as research communities in this immensely important area. The HMT differs from these approaches since it builds upon message interactions as its core building block. It offers to extend the simple messaging approaches by adding additional process-oriented extensions. The key contribution of this approach is to effectively merge message-oriented and process-oriented approaches for achieving both inter and intra enterprise application integration.

The focus of this paper was primarily to present the vision of a technology based on harmonized messaging, and consequently identify some open and interesting question. Below we list the main issues and challenges in developing the proposed technology. We believe that the specification, usability and simulation requirements present *research* challenges. In addition, this work may invoke interesting *technology* questions on the underlying issues of data management/persistence, scalability & performance, reliability and security.

### **Specification**

We see two main aspects of the specification framework: Informational: Specification of the minimal relevant data that will constitute a message exchange; and Behavioural: Specification of the relationships between external (received) and internal (sent) messages.

### **Usability**

High level language (preferably visual) that masks the complexity of the specification language and provides a verifiable one-to-one mapping.

### **Persistence**

This entails effective management of large volume message stores, archival strategies, and effective indexing.

### **Scalability and Performance**

Handling scalability and performance under high volume network traffic is an essential aspect of the system.

### **Transactionability**

Recovery from crash and system failure.

### **Security**

Providing security and desirable levels of trust. This aspect has heightened importance in the context of cross organizational communication.

### **Simulation**

An essential function is to provide validation of the specification against redundancy, contradictions, cycles etc. at pre-deployment; and monitoring and reporting for the active system.

## REFERENCES

- [1] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns, Distributed and Parallel Databases, 14, 5-51, 2003, Kluwer.
- [2] Arnold D. and Segall B. Elvin has left the building: A publish/subscribe notification service with quenching. AUUG97 Technical Conference. 1997. Brisbane, Australia.
- [3] BPEL4WS – Business Process Execution Language for Web Services (2002) [www-106.ibm.com/developerworks/webservices/library/ws-bpelcoll/](http://www-106.ibm.com/developerworks/webservices/library/ws-bpelcoll/)
- [4] Christoph Bussler. (2002) The Role of B2B Protocols in Inter-Enterprise Process Execution. 3rd VLDB Workshop on Technologies for E-Services TES'02 <http://gkpc14.rbg.informatik.tu-darmstadt.de/tes02/>
- [5] Robert .M Colomb & Maria .E Orlowska. (1994) Interoperability in Information Systems. Information Systems, 5(1), pp.37-50, 1994.
- [6] ebXML [www.ebxml.org](http://www.ebxml.org)
- [7] R. E. Gruber, B. Krishnamurthy, and E. Panagos. (1999) The architecture of the READY event notification service.. IEEE International Conference on Distributed Computing Systems Middleware Workshop. 1999.
- [8] IBM. WebSphere MQ series [www-3.ibm.com/software/ts/mqseries/](http://www-3.ibm.com/software/ts/mqseries/)
- [9] Stefan Jablonski, and Christoph Bussler (1996): Workflow Management: Modeling Concepts, Architectures and Implementation. London; Sydney, International Thompson Computer Press.
- [10] Heather Kreger (2001) Web Services Conceptual Architecture (WSCA) 1.0. IBM Software Group May 2001.
- [11] ODBase (2002 – 2004) International Conference on Ontologies, Databases and Applications of Semantics. ([www.cs.rmit.edu.au/fedconf/odbase/2004/](http://www.cs.rmit.edu.au/fedconf/odbase/2004/))
- [12] RosettaNet [www.rosettanet.org](http://www.rosettanet.org)
- [13] Wasim Sadiq and Maria E. Orlowska. (2000): Analyzing Process Models using Graph Reduction Techniques. Information Systems 25(2):117-134.
- [14] SOAP – Simple Object Access Protocol Specifications Version 1.2 (2003) [www.w3.org/TR/soap/](http://www.w3.org/TR/soap/)
- [15] Robert Strom, Guruduth Banavar, Tushar Chandra, Marc Kaplan, Kevan Miller, Bodhi Mukherjee, and and Michael Ward Daniel Sturman. (1998) Gryphon: An Information Flow Based Approach to Message Brokering. International Symposium on Software Reliability Engineering '98, 1998.
- [16] UDDI [www.uddi.org/](http://www.uddi.org/)
- [17] Jon Uldell. (2002) Orchestrate Services. Info World. July 2002. [http://www.infoworld.com/article/02/07/05/020708plweborch\\_1.html](http://www.infoworld.com/article/02/07/05/020708plweborch_1.html)
- [18] WSCI – Web Services Choregraphy Interface 1.0 (2002) [www.w3.org/TR/wsci/](http://www.w3.org/TR/wsci/)
- [19] WSDL – Web Services Description Language 1.1 (2001) [www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl)

**Received:** 20 May 2004

**Accepted:** 08 December 2004