RC24274 (W0706-009) June 1, 2007
Computer Science

IBM Research Report

Extending the Enterprise Service Bus for Digital Media

Paolo Dettori, Julio Nogima, Frank Schaffa
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Extending the Enterprise Service Bus for Digital Media

Paolo Dettori, Julio Nogima and Frank Schaffa
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
{dettori,jnogima,schaffa}@us.ibm.com

ABSTRACT

Offline digital media processes such as media production and distribution are evolving rapidly to enhance process automation and management and to cope with the proliferation of new formats and distribution channels. The adoption of web services as building blocks for developing increasingly complex media processes is an important step toward reuse of media processing tools. At the same time, the adoption of the Enterprise Service Bus (ESB) pattern provides a single point of integration for multiple services and provides common features such as service publishing and discovery, reliable message delivery and transformation. In this paper, we present an approach to extend the concept of ESB to support offline digital media processes. A key aspect of this technique is the combination of high-level abstract service composition and dynamic composition for adapting the abstract processes at run-time based on the specific multimedia objects exchanged during the execution. Since the abstract process defined on the extended ESB is sensitive to media formats and transport protocols, this approach reduces the complexity of creating flexible media processes.

Categories and Subject Descriptors

H.5.1 [**Information Systems**]: Multimedia Information Systems – *Audio input/output, Video.*

General Terms

Design.

Keywords

Digital Media, Enterprise Service Bus, ESB, Architecture, Web Services, MPEG-21, Service Oriented Computing, SOA, Service Composition, Abstract Process.

1. INTRODUCTION

Digital Media (DM) production and distribution is a complex process that requires capturing, storing, searching, annotating, editing, repurposing, quality-controlling and distributing multimedia content. The inclusion of metadata which can be exchanged across the tools participating in the process greatly increases the value of the content and provides useful information to the overall process [1]. Adopting well defined interfaces and standardized mechanism for the exchange of content and metadata is the first step toward more efficient and automated production processes [2]. The next step, service oriented computing, an architectural approach that utilizes services as atomic components for building new applications, provides further advantages by encapsulating the application logic of each tool with a well defined interface which promotes the reuse of the same service in different applications [3]. A further step is the adoption of the Semantic Web to provide richer semantic specifications to Web Services; in particular to specify in a standardized way what “a service does”, “how it works” and “how it should be invoked”. The Ontology Web Language for Web Services (OWL-S) is an OWL-based Web service ontology, which supplies a core set of constructs for describing the properties and capabilities of Web services and which has been adopted by the W3C [4]. The Semantic Web provides the capability to discover services matching certain criteria in automated or semi-automated way.

The approaches described so far provide considerable improvements over the current practice of building monolithic applications for processing DM production and distribution, since services can be used to encapsulate tools that can then be discovered and re-used in different applications. However, in today’s complex production environments, where tens or hundreds of different tools need to be connected, and where the number of production processes for different business requirements and media formats is increasing considerably, these approaches alone may prove inadequate to manage the complexity of building new production processes or adapting existing ones. An example of the growing complexity of point-to-point applications for DM production is illustrated in Figure 1. The diagram illustrates the integration points between DM applications and services. The key issue with the point-to-point integration is that each service needs to be connected to other services which are required to perform a certain task, and the result is a complex mesh of services where each service may depend on a number of other services. The consequence is a number of applications in the organization which serve specific departments needs, but rarely are re-usable. This type of architecture is rigid and slow to adapt to evolving production models.

We have developed an approach based on an Enterprise Service Bus (ESB) to provide a common platform to the services involved in the DM processes. The ESB provides the connectivity layer between services delivering messages from service requestors to service providers. A key advantage of an ESB-based approach is that a service needs to be integrated only with the bus, and becomes available to all the other participants connected to the bus through a common messaging infrastructure. The notion of ESB is well-established in the industry and the technical community [5, 6]. An ESB provides support for exchange of messages across services but does not address the exchange of DM. We have extended the notion of ESB to provide support for DM, extending core ESB concepts such as message transformation, persistence and routing decisions to DM objects.

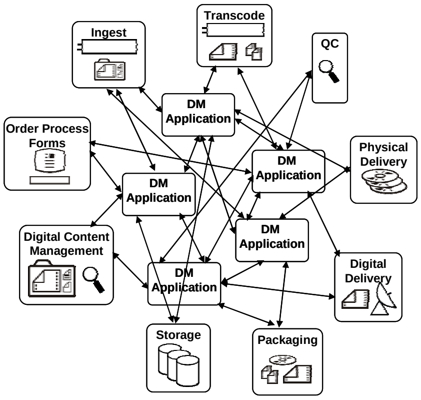

Figure 1. Example of Integration for DM Production

The rest of the paper is organized as follows: first in section 2.1 we introduce the ESB concept, and then we propose our approach for extending the ESB to support DM processes in section 2.2. Section 3 describes our implementation and validation of the approach and finally section 4 concludes the paper and presents future work.

2. ENABLING ESB FOR DIGITAL MEDIA

In this section we discuss the basis of our architecture.

2.1 Enterprise Service Bus (ESB)

An Enterprise Service Bus provides a common infrastructure for connecting service requestors and providers. Furthermore, the ESB enables publishing and discovering information on the available services and applying mediations on the messages that flow between requestors and providers. An ESB provides a service registry where information on the available services is maintained. Service requestors can use the registry to find services matching certain service characteristics. Message mediation supports adapting message formats between different services, logging messages, and changing dynamically the target of a specific message according to payload or control information in the message (dynamic routing). The combination of dynamic routing, message transformation and registry lookup enables more sophisticated features such as service virtualization (the capability of selecting the actual service provider at runtime based on user-defined criteria). Finally, the ESB provides reliable delivery of messages by persisting messages in a data store until the message is delivered to the destination.

2.2 Extended ESB

An extended ESB provides a common infrastructure to connect all the multimedia services participating in the media workflow. Figure 2 illustrates an example of ESB with a set of connected media services. In order to extend the concept of ESB to DM, the following aspects needs to be considered: a) common representation of the digital media assets across all services connected to the bus; b) an unified and reliable mechanism to move media content between services; c) a mechanism to provide media transformation for resolving media format mismatches; d) semantic representation of services based on media processing capabilities; e) abstraction of process orchestration for supporting multiple media formats and protocols with adaptive, dynamic media flows. We discuss our approach for each aspect in the following subsections.

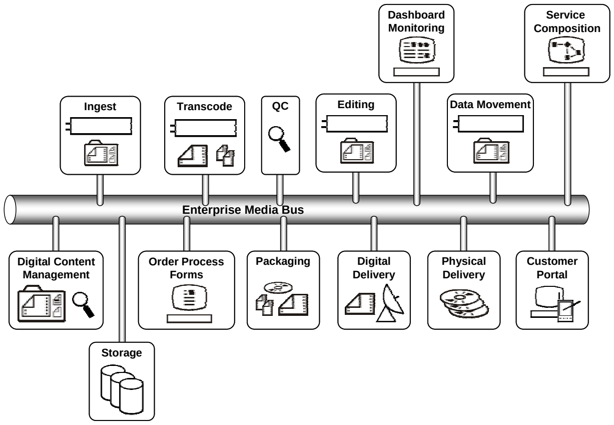

Figure 2. Enterprise Media Bus

2.2.1 Common Representation of Content

Services exchange messages on the ESB using the standardized Simple Object Access Protocol (SOAP) [7]. We have extended the concept of message exchanged across services adding a new data type which represents a media asset. The new data type is based on the MPEG-21 standard, and in particular on the Digital Item Declaration (DID) [8, 9]. The MPEG-21 DID provides an abstract model for declaring digital items, and an XML schema to represent the model in XML. In comparison to other industry standard such as the Material eXchange Format (MXF) [14], the DID offers the advantage of placing metadata separately from the content essence. This allows for a meaningful representation of the content to be stored and interrogated by the ESB in the service selection process without the overhead predicated by the content itself.

A set of entities provide the basic building blocks for defining the digital item: a *resource*, which represents any multimedia component such as a video or audio file or image; a *component*, which contain one or more resources; an *item*, which can contain other items or components; a *container*, which can contain one or more items and/or containers; a *descriptor*, which can be used to provide metadata about all the other components. In our approach, we represent media files with resources, providing a URI which identifies the asset and provides network access to each resource. We then group one or more resources in a component and describe each resource format using a set of tags providing common media characteristics. For example for a video asset these characteristics may include the codec used for encoding, the video bit rate, and the frame rate and frame size. The MPEG-21 DID provides also a standardized container to transport and exchange metadata information generated during the production process. Metadata can be represented using standardized formats such as MPEG-7, SCORM, Dublin Core or any proprietary schema. While it is expected that most of the services will be able to consume the basic media format information provided in the DID, only a few of the services might need to consume and process metadata information, and the metadata information required would differ from service to service. The capability of services to support a specific format and metadata model is then modeled in the semantic representation of the service, as described in section 2.2.4.

2.2.2 Unified Content Transport

An ESB is designed as a messaging infrastructure and is specifically not designed to transport large payloads such as large multimedia files. An additional content transport infrastructure is therefore required for moving content across services. The MPEG-21 DID exchanged between services does not contain the media resources but references to the media which can and often must be local to the services participating in the process. As a result, the Extended ESB supports the exchange of messages containing control information for the transport of content and metadata information describing the content. A specific category of service, the *Data Mover* has been defined for providing reliable transport of media across services. Data movement characteristics of each service are modeled in the semantic representation of the service to facilitate dynamic service selection. Data Mover agents are also installed in the services connected to the ESB for supporting movement of content so that it is accessible to the service once invoked by the ESB. Conceptually, a Data Mover can be represented as a virtual content bus controlled by the service selection and service invocation process of the ESB. The extended ESB will have then a message exchange layer and a content exchange layer. Implementing this content bus as a service gives the advantage of a pluggable solution for the content transport. Data mover implementations range from simple implementations based on standard network protocols such as FTP to commercial systems. We have tested both simple implementations and commercially available software.

2.2.3 Content Transformations

A standard ESB provides message transformation through mediation. A message transformation can be used to adapt a message generated from a service requestor to the format required by a service provider, but it is generally not intended to operate on media content provided as payload or as a reference in the message. The extended ESB provides transcoding services to transform the multimedia content to the format required by any of the service providers. Transcoding services are connected to the extended ESB and provide transcoding of multimedia content resources referenced in the MPEG-21 DID. The content transformation function is provided by a set of pluggable services, each service having differing capabilities as defined by their semantic representations. This approach allows selecting the most appropriate service for each particular content format to be transformed.

A particular category of content transformation is the transformation of the MPEG-21 DID metadata representing the content essence. Some services need not only to be able to process the content essence but also the content metadata, and in some cases only the content metadata. Different services will also require different representations and subsets of the common metadata. For this reason a number of metadata transformation services can be connected to the ESB and provide services such as transformation between MPEG-7, SCORM, MXF, and proprietary metadata representations.

Content transformation and transport provide the key services required to share content transparently across all the services participating in an ESB based workflow.

2.2.4 Semantic Representation of Services

One key aspect of automating the construction of media processes is the ability to discover, automatically, services with certain capabilities. For example, a particular process may require a task for converting a video file from a DV25 to a MPEG-4 format. To provide this capability, we have adopted the OWL ontology to represent taxonomies of basic multimedia service and to represent capabilities in terms of support of specific media format and media processing capabilities. A service registry provides support for publishing and discovery of services, and acting as a central repository for service metadata. Furthermore, the service registry enables service virtualization, supporting selection of services at runtime according to static service characteristics (e.g. QoS, cost) and real-time performance data (e.g. CPU utilization, disk space, queue length). Service requestors submit semantic queries to the service registry to locate services with the required processing capabilities (e.g. video transcoders) and media format support (e.g. the transcoder should support DV25 in input and MPEG-4 on the output).

Semantic service provider representation enables the ESB to map service requestor input, as specified in the MPEG-21 DID, with service provider capabilities to find a compatible service (or set of services) to perform the required media processing. This level of automation abstracts much of the detail required in media application integration and greatly simplifies the process of constructing media based workflows.

2.2.5 Abstraction of Process Orchestration

The proliferation of content formats and distribution channels (mobile devices, IPTV, DVDs, Blue-Ray and HD DVDs) is fueling an extraordinary growth in the number and variations of media processes for media production and distribution. Media content can be acquired from different contributors in multiple formats, and may need to be processed and distributed in a variety of formats. It is intuitive that the *abstract* definition of the media production process is relatively high-level and it mainly depends on business and production considerations (e.g. where acquiring content from and how content should be edited and processed for distribution). On the other hand, a number of variations need to be implemented in the process when new formats become available and new multimedia tools are introduced. Current approaches for service composition can be classified as static and dynamic [10]. These strategies are equivalent to design-time and run-time composition. Static composition has gained widespread acceptance in the industry, while dynamic composition strategies, mainly driven by the semantic web, are still evolving. To exploit the best of both strategies, we have designed a two-level approach for defining processes for services on the extended ESB. At the abstract level, a process can be defined and implemented using standard static service composition techniques such as WS-BPEL [10]. At the concrete level, composite services for adapting media formats or moving content are dynamically defined on the extended ESB. The problem of matching services for semantic service composition has been addressed by several research papers. Paolucci et Al. [11] describes a solution-based on DAML-S for semantic matching between service advertisements and required capabilities. The matching algorithm proposed in [11] is limited to comparing inputs and outputs of the service provider with inputs and outputs of the service requestor. To exploit the rich semantics of the metadata describing the multimedia formats, we have extended the semantic matching algorithms to model the multimedia capabilities of services and the media format descriptions packaged with the resource references in the MPEG-21 DID as part of the capability matching process. The matching algorithm is then used in a non-back trace Backward Chaining Algorithm [12] to create a composition plan. The execution plan is then translated into a state machine described using State Chart XML (SCXML) [13] and then dynamically deployed in a SCXML engine on the extended ESB. The dynamically defined composite service is then available for invocation by the abstract process engine. With this approach, a small set of abstract processes are dynamically extended at run time to cope with the increasing variety of media formats.

3. SYSTEM VALIDATION

To validate our approach with a real-world offline production process, we have built an extended ESB based on a commercial ESB product (IBM WebSphere Enterprise Service Bus). The ESB runs on top of a standard J2EE application server and provides mediation, dynamic routing and message adaptation as discussed in 2.1. We have designed and implemented a set of custom mediators for performing the functions described in 2.2, using the Java programming language and the ESB mediation API. The *Data Mover* unified content transport and transformation have been provided as external web services based on commercial products. The engine running the SCXML generated from the dynamic service composition algorithm has been based on the open-source Apache Commons SCXML engine. The service registry with OWL support has also been based on a commercial product (IBM WebSphere Service Registry and Repository). We have then defined an abstract Long Form video production and distribution process, which involves the following tasks: 1) submission of production requests; 2) resources planning; 3) content acquisition; 4) content editing; 5) content distribution to four different distribution channels (Web Commerce, Mobile Phones, VOD and a video screen in an open space on Second Life). The objective of this work has been the evaluation of: a) the feasibility of our approach; b) the applicability to real-world scenarios; c) time required to build new workflows; d) time required to add new services to the ESB and e) testing the advantages of working with abstract process definitions and having the extended ESB providing the dynamic adaptation of media formats and content transport. A demonstration has been presented at the National Association of Broadcasters (NAB) in Las Vegas, April 2007, and it has showcased in a single platform multimedia software tools from 11 software vendors. The demonstration has been a valuable ground for discussions with industrial parties involved in media production and distribution, and an opportunity for assessing the applicability of our approach to real-world scenarios.

4. SUMMARY AND CONCLUSIONS

In this paper, we have presented a novel ESB-based approach for supporting service-oriented production and distribution processes. Our discussion has been initially focused on showing how a bus strategy simplifies the creation of media processes when compared to traditional application-to-application integration. We have then introduced the concept of ESB and presented our approach for extending the ESB pattern to multimedia-intensive application domains. We believe that our work is novel in the following aspects: 1) the extension of an ESB for multimedia-centric processes; 2) a two-level approach combining static abstract service composition with dynamic composition adapting the abstract processes at run-time to the specific multimedia objects exchanged by the service providers and requestors. As part of future work, we are further testing the scalability of our method for the dynamic service composition.

5. REFERENCES

[1] Nack, F. Capture and Transfer of Metadata during Video Production. In *Proceedings of the ACM Workshop on Multimedia for Human Communication - From Capture to Convey (MHC 05),* November 2005.

[2] Hardman, L. Canonical Processes of Media Production. In *Proceedings of the ACM Workshop on Multimedia for Human Communication - From Capture to Convey (MHC 05),* November 2005.

[3] Papazoglou, M. P. and Georgakopoulos, D. Service-Oriented Computing. In *Communications of the ACM,* October 2003/Vol. 46, No. 10.

[4] Martin D., Paolucci M., McIlraith S., Burstein M., D., McDermott D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan, and K. Sycara. Bringing Semantics to Web Services: The OWL-S Approach. In *Workshop on Semantic Web Services and Web Process Composition (SWSWPC2004), Lecture Notes in Computer Science.* Springer, July 2004.

[5] Min Luo, Benjamin Goldshlager, and Liang-Jie Zhang. Designing and implementing Enterprise Service Bus (ESB) and SOA solutions", *Tutorial, IEEE International Conference on Services Computing, SCC 2005*, Orlando, FL, USA, July, 2005, p. xiv.

[6] Ueno, K., Michiaki Tatsubori. Early Capacity Testing of an Enterprise Service Bus. In *Web Services, 2006. ICWS '06. International Conference on,* Sept. 2006 Page(s):709 – 716

[7] Curbera F. , Duftler M. , Khalaf R. , Nagy W. , Mukhi N. , Weerawarana S., Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI, In *IEEE Internet Computing*, v.6 n.2, p.86-93, March 2002

[8] Bekaert J., De Kooning E., De Sompel H., Representing digital assets usingMPEG-21 Digital Item Declaration. In *International Journal on Digital Libraries*, v.6 n.2, p.159-173, April 2006.

[9] Burnett, I., Van de Walle, R., Hill, K., Bormans, J., and Pereira, F. MPEG-21: Goals and Achievements. In *IEEE Multimedia*, Oct-Dec 2003, pp. 60-70.

[10] Dustdar, S. and Schreiner, W. A survey on web services composition. In *Int. J. Web and Grid Services,* Vol. 1, No. 1, pp.1–30, 2005.

[11] Paolucci M, Kawamura T., Payne TR and Sycara K. Semantic matching of Web services capabilities. In

Proceedings of the 1st international Semantic Web conference, Sardinia, Italy, June 2002, pp 318–332.

[12] Liu J., Cui J. and Gu N. Composing Web services dynamically and semantically. In *Proceedings of the E-Commerce Technology for Dynamic E-Business*, 2004.

[13] Auburn R.J., Barnett J., Bodell M., and Raman T.V. State Chart XML (SCXML): State Machine Notation for Control Abstraction 1.0. *W3C Working Draft*, 2005.

[14] Devlin B., MXF – the Material exchange Format. *EBU Technical Review*, July 2002.