A Media Delivery Framework for On Demand Learning in Manufacturing Processes

Martin Zimmermann
Production Systems Engineering
University of Applied Sciences Offenburg
77654 Offenburg, Germany
m.zimmermann@fh-offenburg.de

Abstract — **This paper presents a streaming-based E-Learning environment where closer integration between learning and work is achieved by integrating multimedia services into manufacturing processes. It contains a comprehensive and detailed explanation of the proposed E-Learning streaming framework, especially the adaption of streaming services to mobile environments. We first analyze several scenarios where E-Learning streaming services can be integrated into manufacturing processes. To allow systematic and tailor-made integration, we develop a model and a specification language for E-Learning streaming services and apply the model using practical scenarios from real manufacturing processes. Adaption of multimedia streaming services to mobile devices is discussed based on Synchronized Multimedia Integration Language (SMIL). Last, we comment on the benefits of using E-Learning streaming services as part of manufacturing processes and analyze the acceptance of the developed system. The key components of our E-Learning environment are 1) an xml based streaming service specification language, 2) adaption of multimedia E-Learning services to mobile environments, and 3) Web Services for searching, registration, and creation of E-Learning streaming services.**

Keywords - Just-in-time Learning, Media Streaming Services, Manufacturing Processes, Mobile Devices, Push Service, Pull Service, SMIL, Web Services.

I. INTRODUCTION

This article is an extended and revised version of the conference paper entitled “On Demand Learning in Manufacturing Processes” [1]. It contains a more comprehensive and detailed explanation of the proposed E-Learning framework, especially the adaption of multimedia streaming services to mobile environments.

Most E-Learning projects tend to separate learning activities from everyday work. This paper presents an approach where closer integration between learning and work is achieved by integrating multimedia services into manufacturing processes. A major challenge to which companies must respond is the integration of advanced E-Learning technologies. What is actually needed is a learning “on demand”, embedded into work processes, responding to both requirements from the work situation and from employee interests, a form of learning crossing boundaries of e-learning and performance support.

The goal of E-Learning services integration in manufacturing processes is, through the development of new IT solutions, to accelerate and enhance the ability of manufacturing industry to capitalize on the emergence of a powerful global information infrastructure.

Manufacturing processes involve the control and management of manufacturing systems ranging from basic assembly processes to the high-tech manufacture of pharmaceutical, telecommunications and electronic equipment. Categories for such manufacturing processes are assembly line / flow shop, and cellular manufacturing / group technology. As an example, in case of assembly line based processes a line of dissimilar machines are grouped in the line (sometimes more than one to balance flow). Innovation, productivity, flexibility, and continuous improvement are key ingredients to success in the constantly evolving world of manufacturing.

Multimedia networking services support monitoring, controlling and supervising production processes in order to achieve high levels of efficiency and environmentally friendly production. The new flexibility of workers and work environments makes traditional conceptions of training in advance, in rather large units and separate from work activities, more and more obsolete. Manufacturing scenarios where E-Learning services can be integrated are shown in Fig. 1.

Computer networks provide a rich environment for constructing such interactive, intelligent, active, and collaborative learning environments. They offer certain distinguished advantages over traditional learning environments. In a traditional learning environment, employees are required to gather at a certain time and place to attend a lesson. In contrast, on demand-based learning allows employees from geographically separated locations to join lessons.

The proposed method adopts the client-server architecture to support multimedia content transmission. A media server may be a single machine or a group of machines to manage contents of the Internet-based learning system. It handles requests from users for push and pull services. In particular, it determines and schedules the required e-learning contents with optimal details to transmit to the users.

The following E-Learning streaming services can be identified:

- Pull (on demand) service: Allows on demand access to remote E-Learning content, e.g., video content illustrating configuration of a Computerized Numerical Control (CNC) machine.
- Push service: As an example, a push channel from a remote expert to the manufacturing personal (“how to configure / operate a machine”). Live audio and video (expert) as well as media files can be delivered (see Fig. 1).
- Push service: For recording a failure scenario during a manufacturing process (by recording the voice comments of a worker and by recording the machine status by video). This multimedia documentation can be used later to analyse details.

A further service type are peer to peer and multiparty conferences, e.g., for maintenance and remote diagnosis: teleoperation and remote diagnosis and maintenance of distant plant and production equipment can be achieved by using peer to peer E-Learning streaming services.

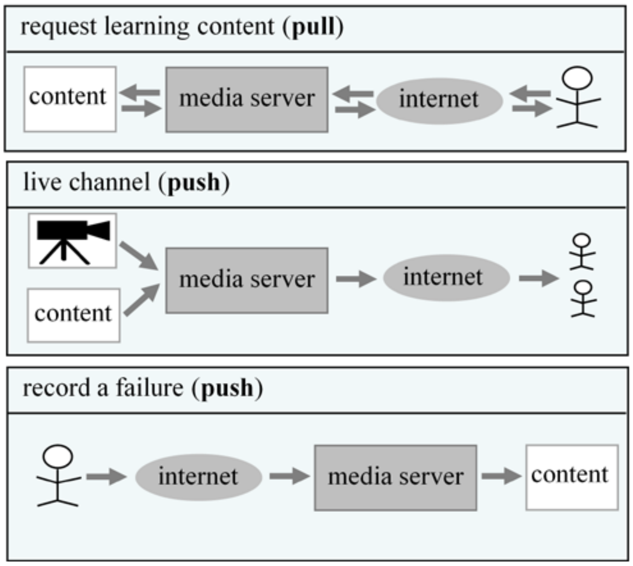

Figure 1. E-Learning services in manufacturing processes

The remainder of this document is as follows. Section II makes a comparative digest into related work in this field of interest. In Section III, we discuss a XML-based language, which allows specification of E-Learning streaming services while Section IV presents a generic SMIL-based adaptation methodology suitable to adapt E-Learning services to different access mechanisms, device capabilities and user preferences. Section V outlines the architecture and describes the major components and technologies to solve the problems stated in Section I. Finally, Section VI gives a brief summary and concludes with a note on future work.

II. RELATED WORK

Khasawneh et al. present a framework for an on-demand E-learning management system that make use of broadband network for the delivery of distributed "educational activities" such as distributed courses, tutoring sessions, lectures, workshops, etc. [2]. The developed scheme is tailored towards personalized learning using distributed information in a dynamic and heterogeneous learning setting, i.e., a connected network of learning management entities and educational systems where learners are individually supported in accessing distributed resources. However it is restricted to on demand scenarios and multimedia files.

Borissova et al. introduce a framework for design and development of an interactive multimedia E-learning system for engineering training [3]. The main goal of the project is to encourage low cost developing of effective and customized e-learning systems for engineering training by using popular and inexpensive software tools especially for virtual simulation of engineering system operations.

The Venice [4] project proposes a Web Services-based framework for VoIP applications. By using a service oriented architecture, the authors aim at easing the integration of supplementary services, the compatibility between different signaling layer protocols for call control and the installation of software updates on client devices. However, the Venice project is specifically tailored to VoIP scenarios, i.e., the authors do not address reusability between different multimedia applications nor provide a generic platform for the development of these.

Based on a distributed messaging middleware, the Global Multimedia Collaboration System (Global MMCS) [5][6] provides a framework for an audio/video collaboration system, which bridges the gaps between nowadays multimedia applications by providing a common signaling protocol with gateways to existing protocols like SIP or H.323 [10]. However, the authors do not address other multimedia applications and the reuse of components between these.

In [7], an approach is suggested to combine the areas of e-learning and Web Services, by providing electronic learning offerings as (individual or collections of) Web Services as well. It elaborates on this by showing how content providers and content consumers (i.e., learners) can communicate appropriately through a Web Service platform with its common description, publication, and retrieval functionalities. However, it does not support live channels.

The JSR 309 [16] is designed to provide server-based Java applications with multimedia capabilities. It targets a large range of applications from simple ring-back tone applications to complex conferencing applications, by providing: media network connectivity to establish media streams, IVR functions to play/record/control multimedia contents from file or streaming server, ways to join/mix IVR function to network connection to create conferences and call bridges. However the proposed API approach is restricted to Java based applications and it does not support enhanced VoIP functions, e.g., ring groups or call queues. It mainly provides low level objects like players, recorders, mixers and connections that developers can manipulate or combine together to obtain all the multimedia capabilities.

Scholz et al. present a generic framework for multimedia applications consisting of a set of reusable Web Service components, a modeling language based on finite state automata and a compiler [17]. The authors concentrate on the signaling plane protocols, especially their similar structure and purpose, i.e., the definition of possible states for both, client and server, and the transitions between these states via the exchange of messages.

Chou et al. describe a service-oriented communication (SOC) paradigm based on Web Services for real-time communication and converged communication services over IP [11]. This approach extends Web Services from a methodology for service integration to a framework for SOC. In particular, it introduces the generic Web Services-based application session management (WS-session), the two-way full duplex Web Services interaction for communication, and the development of Web Services Initiation Protocol.

In [12], a mobile streaming media CDN (Content Delivery Network) architecture is presented in which content segmentation, request routing, prefetch scheduling, and session handoff are controlled by SMIL [13] (Synchronized Multimedia Integrated Language) modification. The approach concentrates on the segmentation aspect, which is important for mobile users.

Oliveira et al. present a proposal to solve the problem of the adaptation of multimedia services in mobile contexts [19]. The approach combines context-awareness techniques with user interface modeling and description to dynamically adapt telecommunications services to user resources, in terms of terminal and network conditions. The solution is mainly characterized by the approach used for resolving the existing dependencies among user interface variables, which is based on the constraints theory, and by the mechanism for acquiring the user context information, which uses the Parlay/OSA interfaces.

III. SPECIFICATION OF E-LEARNING SERVICES

To overcome the restrictions of traditional learning environments discussed in Section I, we present a novel concept for a model-driven development and deployment of E-Learning streaming services. Its central idea is the specification of E-Learning services in terms of manufacturing resources, media objects, and delivery policies. By using Web Services for composition of E-Learning streaming services, we facilitate reuse, customizability and technology-independence.

A. *Media Streaming Model*

The information model in Fig. 2 represents a model of our E-Learning streaming services (UML class diagram). Different fundamental components (represented by rectangles) of the system are shown in the form of classes and relationships.

Manufacturing processes are composed of manufacturing resources, e.g., manufacturing automation devices, equipment & machinery, material & manufactured parts, and manufacturing personnel. E-Learning streaming services are related to such manufacturing processes and / or single manufacturing resources. As an example, for a single machine a video on demand can be requested by the manufacturing personal to study the detailed configuration steps for such a machine.

According to the model in Fig. 2 an E-Learning service can be described by its media objects (e.g., audio and / or video objects), related manufacturing resource, distribution and replication policy, and the quality of service used for content delivery. Media streams can also be categorized according to how the media objects are delivered.

Fig. 2 shows how pull services, push services and conference services inherit from the E-Learning base class. This generalization relationship indicates that these sub classes are considered to be a specialized form of the super class (E-learning). In pull services data delivery is initiated and controlled from a client whereas in push services a server initiates data transfer and controls the flow. Content Caching means to replicate media objects across different E-Learning servers. Such content caches improve application performance by storing frequently requested content closer to end users of the content and by offloading servers from repetitive requests.

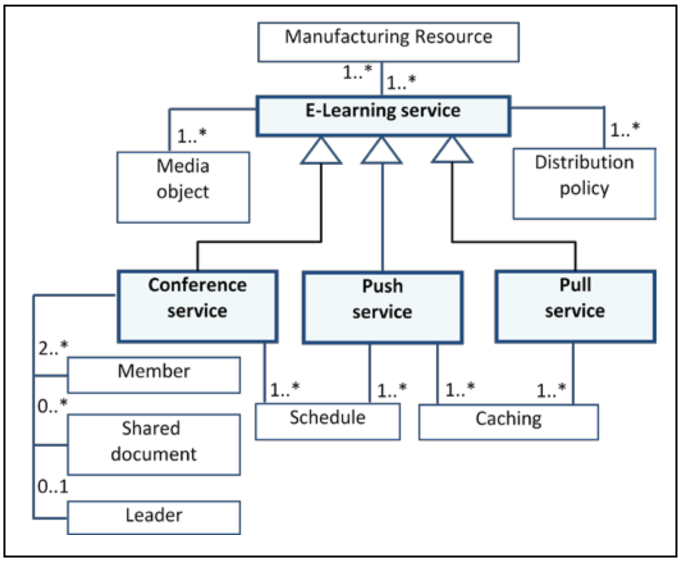

Figure 2. E-Learning service model

Conference services support scheduled and ad-hoc conferences. Such conferences contain the following attributes: date and time when the conference will take place, the invited participants and as an optional part the leader, the agenda as well as conference documents.

B. *Use Case: Live training*

The following scenario (Fig. 3) illustrates one of our implemented key use cases, a technical training (how to configure and operate a machine) for a computerized numerical control machine (CNC) as the related manufacturing resource.

The operator shows the handling of a machine to the distributed audience (video). A background speaker (at a separate location, e.g., office) explains the configuration steps (video / audio). To enable the interaction between the speaker and the learning community, each participant is offered an audio back channel. In the scenario there are different QoS requirements, e.g., for the video transmission showing the manufacturing machine a high resolution and frame rate is required.

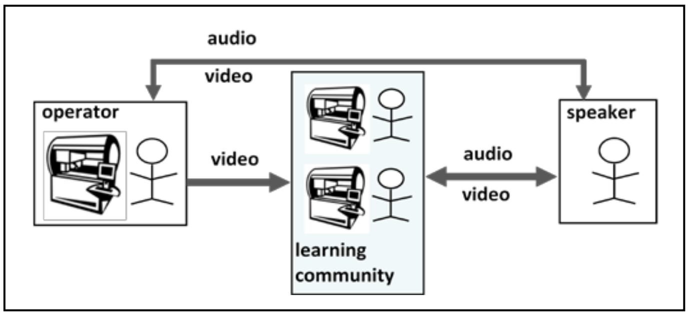

Figure 3. Example E-Learning scenario

C. XML Specification

For the specification of the properties of an E-Learning service we use XML. A specification of an E-Learning service is a structured composition of autonomous objects (see Fig. 4):

- Resources or manufacturing processes (category, type, keywords).
- A collection of media objects, e.g., audio and video objects (live or files) together with temporal relationships.
- Distribution policy, e.g., push or pull
- Roles and related media objects, e.g., speaker role.

This information can be used later, e.g., by the manufacturing personal, for searching and accessing E-Learning services.

Fig. 4 illustrates the specification of the live training scenario introduced in subsection A. The first part contains the description of manufacturing resources related to the E-Learning service. The list of media objects, which are used in the service is part of the section `<MediaObjects>` together with the type, bitrates and source.

For specification of temporal relationships we introduce two timing elements: `<seq>` and `<par>` similar to SMIL [13]. Both elements represent timing containers: they influence how the timing of their children is defined. A `<seq>` (sequential) container specifies that its children get processed sequentially: each child, whether it contains media objects or hierarchical structure, is processed when its predecessor ends. Fig. 4 shows the top-level sequential behavior in our E-Learning service: A `<par>` element specifies that its children play in parallel. Two videos and one audio fragment are rendered starting at the default time of the beginning of the `<par>` container.

The section `<DistributionPolicy>` defines the details of media delivery policy. The media streams in Fig. 4 are delivered based on a push policy. This includes also the definition when the media objects must be delivered (date and time), together with the available bit rates. There is also a recording option that allows creation of an archive recording while the live stream is going on.

Finally the roles, e.g., the speaker role are introduced together with a list of media streams, which are consumed and supplied by each role. As an example, the speaker role is a consumer of a video and an audio stream delivered by the role operator.

```
<StreamingService>

<ManufacturingResources>
   <resource  category = ”num control machine” />
   <keywords>live training </keywords>
</ManufacturingResources>

<MediaObjects>
<par>
  <video id = ”videoSpeaker” type = ”live”
                bitrate= … src =  … />
  <audio id = ”audioSpeaker” type = live”
                bitrate= … src = …/>
  <video id = ”videoOperator” type = ”live”
                bitrate= … src =  … />
   . . .
<par>
</MediaObjects>

<DistributionPolicy>
  <push codec=”MPEG4” date= …  time=…
        recording=”yes”  file = …
  </push>
</DistributionPolicy>

<Roles>
   <role name=”Speaker”/>
     <consumerOf>
        <video id=”videoOperator” />
        <audio id = ”audioLearningCommunity”… />
     </consumerOf>
     <supplierOf>
        . . .
     </supplierOf>
   </role>
   . . .
</Roles>
. . .
</StreamingService>
```

Figure 4. E-Learning service specification

IV. SMIL-BASED MULTIMEDIA INTERFACES

While the modeling and specification approach presented in the previous section specifically focuses on design time of E-Learning streaming services, this section presents a SMIL-based approach for adaption of multimedia streaming services to mobile devices. After a brief discussion of key differences between traditional devices and mobile devices, we present our concept in detail.

A. Multimedia Devices

The acceptance of new E-Learning services will only be effective if the users have the possibility to access them anywhere, and in any technological circumstances. Ubiquitous access to E-Learning services is becoming increasingly important with the proliferation of wireless devices and technologies. This requirement means that there is a significant challenge of being able to transform E-Learning services in order to adapt them to a great variety of delivery contexts, i.e., various devices.

Examples in our manufacturing scenarios include:

- Mobile devices that are used by a problem solution team to display just in time information that is relevant to manufacturing problem, e.g., a previously recorded video showing a failure scenario.
- Ubiquitous collaborative learning via cell phones or PC tablets that allow the learner to play an active role in both the knowledge building process and decision making anywhere, anytime.
- Mobile technologies can provide situated learning by extending the learning/training environment beyond the production areas into authentic and appropriate contexts of use.

However, there are a number of key differences between a traditional device such as a PC and a mobile device, such as a PDA. These include especially screen size and resolution, data input capabilities, as well as browser features.

The screen size and resolution capabilities of a traditional PC are quite different from that on a mobile device. PCs have large screens of 15 inches or more with typical display resolutions of 1280 x 1024 or higher. Mobile display resolutions start at around 128 x 128, barely 1/100th of the available resolution on a PC. Although mobile device display resolutions on smart-phones are typically 240 x 320 and can reach 640 x 480 or greater, the physical size of the display is limited by the requirement to have a small pocket-sized device. Increasing the resolution provides more dots-per-inch, but delivers limited benefit on such a small screen. The available screen estate is important when designing E-Learning services for mobile devices, and simply reformatting existing content is not appropriate.

The data input capabilities on mobile devices are restricted compared to PCs. Most mobile devices have only a numeric keyboard and entering text requires multi-tap key entry. The data input constraints should be considered when designing the mobile application, and data entry should be as limited as possible.

B. Adaption of Multimedia Streaming Services

Among the alternatives that may be considered as standardized multimedia formats, i.e., HTML5, MHEG, MPEG-4 and SMIL, we decided to choose SMIL. The main reason behind this choice is the SMIL-based E-Learning streaming specification language (Section III), which is also based on SMIL. The well-designed and extensive SMIL language elements for adaption of multimedia streaming services to mobile devices is another reason.

SMIL (Synchronized Multimedia Integration Language) is a W3C recommended XML-based markup language, which facilitates the construction of accessible multimedia applications for the internet and mobile devices. Collections of XML elements and attributes can be used to describe the temporal and spatial coordination of one or more media objects. SMIL defines markup for timing, layout, animations, visual transitions, and media embedding, among other things. SMIL allows the presentation of media items such as text, images, video, and audio, as well as links to other SMIL presentations, and files from multiple web servers.

A SMIL application references to media objects, not the media data itself, and instructions on how those media objects should be combined spatially and temporally. Relations between media objects (and substructures) are described, and the computation of the timeline follows from this. The main temporal composition operators available are parallel composition, sequential composition and selection of optional content. Composition is hierarchical: nodes cannot become active unless all of their ancestors are active. The declarative containment model has one large advantage: SMIL presentations can adapt automatically to varying bandwidth conditions and alternate content with different durations. The hierarchical temporal composition model represents also a container for timed metadata, and allows structure-based deep linking into the content.

There are several alternatives available for making an E-Learning service compatible with different (mobile) devices:

- Resize the base layout: requires analyzing the used mobile devices and creating a layout that is not bigger than the most restrictive presentation environment. This is very safe strategy, but it can lead to frustration among E-Learning participants who have more sophisticated devices.
- Use device default layout: if no layout section is defined in a presentation, the SMIL player can place objects where it thinks they fit best. While this sounds attractive, it usually results in all content being stacked on top of all other content. The result is rarely pleasing or useful.
- Explicitly allow media objects to be scaled: SMIL provides values for media object placement that allow media objects to be scaled. While this is potentially useful, small devices will often have trouble when scaling media objects and especially video.
- Define multiple layouts using SMIL content control: this is the most useful way of handling multiple devices. SMIL provides a wealth of facilities for providing alternative layouts within one application.

In the following, we illustrate how to apply multiple layouts based on SMIL. One valuable SMIL feature in this context, especially for mobile devices, is the content control. It can be used to select over a number of layouts or media objects based on system properties. It is based on the switch element, which allows only one of its child elements to be chosen, the first one which is acceptable. It can be used anywhere in a SMIL document.

The general layout and some positioning models used in our E-Learning example are shown in Fig. 5. SMIL supports the notion of a top-level container window (called either the root-layout or top-Layout, which contains one or more regions. Each region defines a rectangular collection of pixels and a region stacking order (called the z-index).

The layout contains the root-layout container window (not shown) and several media rendering regions. Some of the regions are assigned an explicit stacking order, others take the default of their parent (in this case, the default for the root is '0'). Each region contains positioning information when screen space is used.

Fig. 5 illustrates three layout strategies for different devices. Depending on the available display area, the video objects part of the E-Learning service are handled in a different way. A standard layout containing three regions (Fig. 5 a): Two video regions, one for the speaker and one for the operator; an additional region for a text object, illustrating additional information, e.g., specification of configuration steps in a manufacturing scenario.

Fig. 5 illustrates also two different layout strategies for mobile devices:

- Fig. 5 c shows a simple positioning model on a small device where only the video of the operator is visible (accompanied by the speaker audio);
- The layout in Fig 5 b shows an overlapping layout, i.e., the small video (speaker) is part of the main video region. SMIL handles such overlaps by giving all regions z-index stacking levels. When two regions overlap, the region with the higher z-index "covers" the other with its overlap. In environments where bandwidth is restricted a image of the speaker could be played instead of the speaker's video.

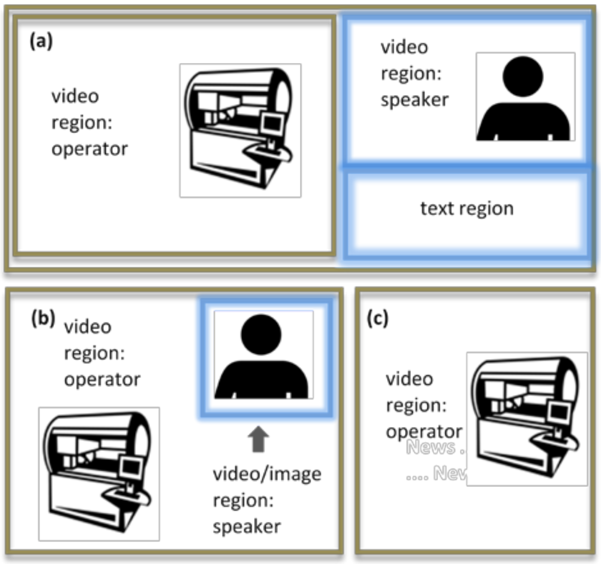

Figure 5. Layout strategies for different devices

The code fragment in Fig. 6 provides an overview of how SMIL handles layout such diversity of devices. One consideration for mobile devices is to add additional content control to address the needs of small devices.

Fig. 6 shows several layout sections within a common switch statement. Every visual media object must be assigned to a region, which is a rectangular area that is places within the root-layout area. The first section contains the standard layout, which has been designed for a display area of 1024 pixels high and 1280 pixels wide (or greater). If this room is available, then the base layout will always be used, i.e., two video regions and an additional text region (not shown in Fig. 6). If a smaller display is available, a second layout is used. This layout could use relative values instead of absolute positioning. It also could allow certain media to be scaled. Rather than using the id attribute of the region element to define region names, the `regionName` attribute is used. This allows multiple layouts to be defined that create regions with the same name. In the body section, a reference to a region is made. The player will select the appropriate layout for the device used during playback.

```
<switch>
<layout systemScreenSize="1024X1280">
... define a standard layout ...
... 3 regions: 2 video regions and   ...
... 1 text region  ...
  <region regionName="videoOperator"
       left="0" width="784 . . . />
  <region regionName="videoSpeaker" . . .
       left="784" . . ./>
</layout>

<layout systemScreenSize="480X640">
... define a smaller layout ...
... 2 overlapping video regions ...
    <region regionName="videoOperator"
       left="0" . . . z-index="1" />
    <region regionName="videoSpeaker" . . .
       left="0" z-index="2"/>
</layout>

<layout>
... define a small and simple layout ...
... 1 video region ...

</layout>
<switch>
```

Figure 6. Layout strategies in SMIL

There are several SMIL profiles developed for use on mobile telephones. While these provide general guidance to manufacturer and industry standardization groups, most handset vendors do not directly implement SMIL. Instead, the embed SMIL functionality in layered standards that include signally, packaging and transport protocols that are optimized for mobile handsets. One such layer set of protocols is developed by 3GPP: the third-generation partnership platform. One common packaging of SMIL in this environment is via MMS, the multimedia messaging system.

SMIL 3.0 contains elements and attributes which provide for runtime content choices and optimized content delivery:

- `BasicContentControl`: contains content selection elements and predefined system test attributes.
- `CustomTestAttributes`: contains author-defined custom test elements and attributes.
- `PrefetchControl`: containing presentation optimization elements and attributes.

- `SkipContentControl`: specifies attributes that support selective attribute evaluation.
- `RequiredContentControl`: defines the `systemRequired` attribute to specify the namespace prefixes of modules required to process a particular SMIL file.

For example, the `BasicContentControl` attributes define a list of test attributes that can be added to language elements, as allowed by the language designer. Conceptually, these attributes represent Boolean tests. When any of the test attributes specified for an element evaluates to false, the element carrying this attribute is ignored.

The example in Fig. 7 is based on test attributes part of `BasicContentControl`. It uses the bandwidth related `<switch>` statement with one that considers display size as well. The second video (speaker video) in Fig. 7 will only be activated if the bitrate is 34400 (or greater) and the screen size is at least 480x640.

```
...
<par>
   <video src="videoOperator.mpg" ... />
   <switch>
      <video region="videoSpeaker"
             src="videoSpeaker.mpg"
             systemBitrate="34400"
             systemScreenSize="480X640"/>
    </switch>
</par>
...
```

Figure 7. Bandwidth related switch statement

We tested the E-Learning scenarios in two devices, a PDA and a laptop, both with the Microsoft Internet Explorer browser as the SMIL player. Microsoft Internet Explorer implements the SMIL profile designated by XHTML+SMIL. This choice had the advantages of being supported by a disseminated browser (even in PDAs and other mobile devices) and of supporting the Web forms mechanism, which was an essential condition for implementing user input capabilities. The main disadvantage was related with the different approach taken by the XHTML+SMIL profile for managing the presentation spatial layout. This profile bases its layout functionality on the XHTML and Cascading Style Sheets layout model and does not use the construction defined by the layout language elements of the SMIL szandard. However, the presentations generated by both approaches are equivalent, in what concerns spatial layout.

V. Service Oriented Architecture

This section presents an open and flexible service-oriented architecture for the dynamic composition and execution of E-Learning streaming services. First, we introduce a set of Web Services and explain how a streaming service is initialized including the dynamic creation and composition of remote streaming components. Then, media streaming to Apple iPhone is illustrated as an example. Finally, we highlight details on the experiences from the E-learning designer's point of view as well as from the end user perspective.

A service-oriented architecture (SOA) is basically a set of services interacting with each other and coordinating some activity. Service providers and service consumers are the two main entities acting on behalf of a user. The Web Service technology additionally addresses a standardized description of a service's functionality using an XML dialect [14]. Using the Web Service Description Language (WSDL) [8], a service provider describes the functionality (interface) of a service in a platform, language, and operation system neutral way while a service requestor talks to these services using SOAP over HTTP (or other transport protocols).

A. *E-Learning Streaming Services*

Streaming services are managed by a set of Web Services (see Fig. 8). Such a Web Service is a URL-addressable software resource that performs operations and provides answers. In our case, the operations offered by the service interface are (see Fig. 8):

- `searchStreamingService`: allows searching a streaming service (based on the elements and attributes which form the xml service specification (e.g., category of a manufacturing resource, type of media object, distribution policy, etc.).
- `registerStreamingService`: enables validation and registration of new streaming services
- `createStreamingService`: instantiates streaming software components and establishes interconnections between components.
- `startStreamingService`: the operation is typically invoked by a scheduler (in case of push based services, driven by date and time values).
- `subscribeToStreamingService`: allows users to subscribe to (future) streaming services (e.g., a remote video based training related to a certain CNC machine). When a new media streaming service is registered or started, a user will be notified automatically by the service manager (as soon as such a service has been registered to the xml database).

OPERATION	PARAMETERS	RESULT
registerStreamingService	service specification (xml)	result code
createStreamingService	service id	result code
searchStreamingService	keywords, category time, date delivery policy (e.g. pull)	list of services
requestStreamingService	service id	result code [rtsp channel]
subscribeToStreamingService	keywords. category time, date delivery policy (e.g pull)	result code [future notifications]
startStreamingService	service id	result code

Figure 8. Operations part of the Web Service

After searching a streaming service, the client (e.g., RTSP media player) requests depending on its role the related SMIL specification which defines the layout and temporal relationships between media objects. In the next step (step 7 in

Fig. 9) it will set up three network channels with all involved RTSP servers. Media data is delivered using the RTP over UDP, as shown in Fig. 9, step 8.

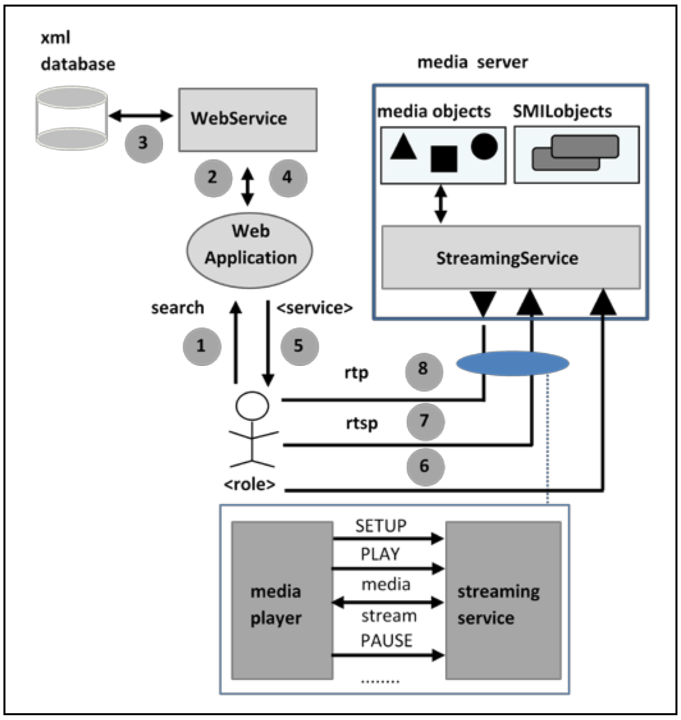

Figure 9. E-Learning Web Services

A full-duplex TCP connection is used for control and negotiation. A simplex UDP channel is used for media data delivery using the RTP packet format. A full-duplex UDP channel called RTCP is used to provide synchronization information to the client and packet loss information to the server. RTSP initiates and controls delivery. The XML service description includes references to media streams, using the URL method rtsp.

According to our service specification language, the following properties of E-Learning services can be used as part of a query:

- Manufacturing resources, e.g., category or keywords
- Media objects properties, e.g., recorded live training
- Distribution policy, e.g., on demand service
- QoS properties, e.g., bitrate, or used codec
- Date / time duration attributes

B. *Software Components*

Our approach is based on a clear separation of a streaming service specification and its implementation by a distributed application and can be used for different streaming paradigms, e.g., push and pull services.

The following figure (Fig. 10) illustrates the user interface and the management interface. The services are managed by a service manager, which provides in the current implementation a simple interface to search, and start (request) services. Moreover, a management interface enables creation of new services, and deletion of existing services. Service specifications are stored as XML documents in a XML database.

A new e-Learning streaming service specification is first analyzed by a Web Service (operation create). Driven by a component library, which contains existing streaming components, such as encoders, media servers, etc., and a set of configuration rules, a Web Service creates a distributed streaming application configuration. Based on such a service specification, the service manager also supports retrieval of existing services.

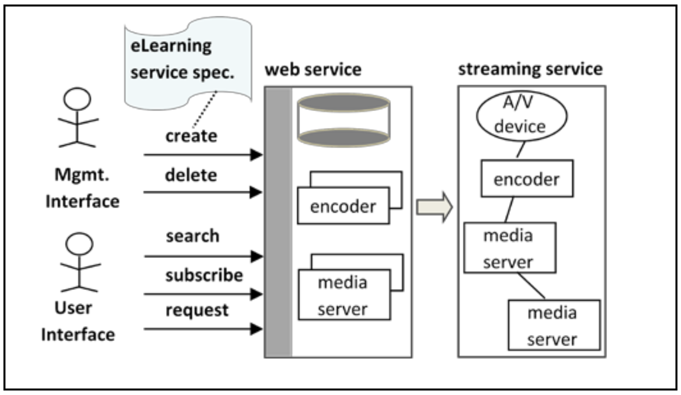

Figure 10. User interface and management interface

Push services are started automatically according to the specified date and time values. Starting a service means to create the required streaming components, e.g., an encoder and a media server component as well as to establish the communication relationships. Before a new service is created, a consistency checker is responsible to test the availability of nodes, and media objects according to the definition as part of a service specification. A service specification is then analyzed by a Web Service, which will select appropriate streaming software components.

Multimedia services are stored as XML documents in a XML database. We use Apache's Xindice, which is a database designed from the ground up to store XML data or what is more commonly referred to as a native XML database. Advantages are: faster for XML than other databases, no mapping to relational required, quick fragment retrieval provided, and optimized XML querying supported.

The prototype has been implemented using the PHP programming language and the PHP Web Services Development Pack nusoap [9] for the creation of and access to Web Services. We use the existing media server components from RealNetworks, and the OSS software Asterisk [15] as a VoIP server. The first (subjective) test results based on implementation of push and pull services as well as conference management are very promising. However, the used OSS VoIP system Asterisk does not offer scheduled conferences, i.e., a direct mapping to the Asterisk conference management functions is not possible. Information related to scheduled conferences is stored in a xml database. As the prototype is still under development, an objective measurement of processing time and delay has not yet been made.

C. Streaming to Mobile Devices

We use a commercial media delivery platform from RealNetworks (Helix Mobile Server) [20] which supports live and on-demand streaming of 3GPP content to any standards-compliant media player. The server supports 3GPP Release 4 Specifications including the .3gp file format and the payload formats associated with MPEG-4, AMR, AAC and H.263. It also implements 3GPP Release 5 Specifications including AMRwideband and enhancements for RTP and RTSP. The Helix Media Delivery Platform supports the various methods used by the Apple devices to access media complete with support for Android, Symbian and Windows 7 Mobile OS clients.

Another reason why we decided to use Helix Mobile Server is the Custom Statistics Reporting facility. When deploying media streaming services to mobile devices, it is important to have accurate statistical information on client usage. The Helix Media Delivery platform, i.e., the Helix Mobile Servers as well as Helix Clients have a statistics-reporting mechanism through which media players can relay back playback statistics to the media server, either at end of a session or on an interval basis. This feature is used to gather statistics about usage of E-Learning streaming services.

We do not use Java Media Framework, because it is not stable enough, especially for long running streaming applications.

In the following, media streaming to Apple iPhone is illustrated as an example. From the user's point of view, the starting point of such a scenario for mobile environments is the request of a SMIL file, which contains the layout specification and the required audio and video files. In the case of Apple mobile devices, the methods used for accessing media objects is very different than delivering content through RTSP (as other mobile clients use) or RTMP (as Adobe flash media players use). Apple has implemented a client-driven system for downloading full or segmented audio and video files through the HTTP or HTTPS protocol. With on-demand media files, these are delivered to the Apple device by converting the original source media file into a series of 10 second segmented media files encapsulated in a MPEG-2 media stream (.ts container) and making them available to be delivered to the client through http or https.

After creating the MPEG-2 TS segments, an index file (.m3u8) is created (or updated if the content is live) that lists the order of and URLs to each segmented media file (Fig. 11). This index file is also known as a playlist and will also include some additional information about the content itself (segment length, if content is encrypted where the decryption key can be found, multirate bandwidth information etc). After receiving the index file, the iPhone OS media client downloads each segment files listed in the playlist using HTTP(S) and plays back the media in order listed. Once the device reads the playlist and works out the bandwidth available (if required for a multi-rate clip) the client then requests the relevant .TS media segment for playback to start (step 5 in Fig. 11 below).

Helix Media Server has the ability to on-the-fly segment (and optionally encrypt) a correctly formatted on-demand media file into the structure required by the Apple device (.ts media segments and m3u8 playlist as seen in step 4 in Fig. 11 above). This process happens instantaneously on the first request by an Apple device (step 3 in Fig. 11) and the playlist is then sent back to the device (step 5 in Fig. 11).

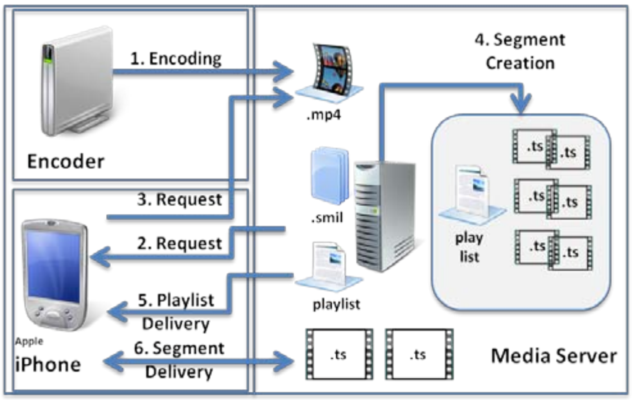

Figure 11. Streaming to Apple mobile devices

Once the device received the playlist and calculated the available bandwidth (if required for a multi-rate E-Learning service) the client then requests the relevant .ts media segment for playback to start (step 6 in Fig. 11).

D. User Experience

The goal of the project was also to evaluate the user interfaces and the usage of the E-Learning streaming services. Two actors have to be regarded to analyze the acceptance of the system: the designer of an E-Learning streaming service (i.e., person responsible for specification of E-Learning streaming services in terms of manufacturing resources, media objects, delivery options, and device adaption policies) as well as the end user accessing on-demand or push streaming services. As in any E-Learning environment streaming services also need to be user centered taking into account the user's characteristics and abilities interacting with the environment and learning materials, and comply with usability.

Designers of E-Learning streaming services have a strong tendency towards reusing designs that worked well for them in the past. However, currently our system does not provide templates for typical E-Learning scenarios. Additionally, the E-Learning specification based on XML files, especially the adaption part which deals with mobile devices was not well accepted. Based on the XML schema for the specification language a graphical user interface will be implemented better suited and more accepted by developers of streaming services.

In a production environment robust and easy to use search functions are one of the key requirements. Our original search functions had to be improved. Context-driven search results, driven by the concrete manufacturing resources, which are part of a single work place are regarded to be important for the machine operators. As an example, depending on the available machine(s), which are part of a single work place, the search functions have to offer the “right” E-Learning services related to the given local manufacturing resources.

The most accepted E-Learning streaming services are on demand video services. From the production management point of view, push services are essential for quality improvements. Such services allow a “just in time” documentation of occurred failures, during runtime of a machine by ad hoc video and audio recordings. These recordings can be analyzed later and appropriate solutions can be developed to improve manufacturing processes.

The challenge of mobile devices is to understand and explore how best they can be used to support learning, training, and performance support. The use of mobile technologies should not be viewed as isolated activities; instead, they should be viewed as richly ubiquitous, and effective collaborative devices having during maintenance, or helping to diagnose during a manufacturing process problem solving effort.

As well as the limited screen size and data input capabilities, the use cases for mobile devices should also be considered when designing E-Learning services and content for mobile devices. Services which are used on-the-go should be made available on the mobile. Just taking an existing PC-based E-Learning service and reformatting it for the mobile phone is rarely the right solution.

Due to the variation in screen resolution and browser markup support, it is necessary to optimize E-Learning content for the specific type of mobile device. This involves designing an appropriate layout for mobile devices, i.e., applying the appropriate mobile page flow, etc. It is also necessary to resize videos and images based on the device resolution.

In order to optimize the content based on the mobile device type, it is necessary to identify the type of device or browser. This can be done automatically using the SMIL attributes for content control. When the device is identified, the characteristics of that device required for content optimization can be retrieved from the SMIL application. The necessary information includes screen resolution, the number and size of regions, etc.

The following research questions are essential for user acceptance of mobile devices:

- What types of training and / or performance support can be effectively delivered via mobile devices that are consistent with the E-Learning environment for manufacturing processes?
- How is the mobile technology training/performance support integrated within the identified E-Learning environment, taking into consideration the identified E-Learning environment’s system architecture?
- What are the existing mobile technologies “best practices” that can be applied to the identified E-Learning environment?

The most interesting metric for the media server performance is the number of concurrent users the server can support. In [22], several streaming use cases are described and tested to report the number of successful streams as well as streams that failed to complete correctly. Streams that died or could not even start were reported as errors. Performance results indicate that the network connection is the bottleneck for HTTP streaming while the CPU is the bottleneck for RTSP based streaming. However, in our case we have a maximum number of 10 parallel streams, i.e., we could not observe any server bottlenecks.

VI. Conclusion and Future Work

In this paper, we have presented a service-oriented architecture for E-Learning streaming services – realized using Web Services technology. The major contributions and extensions of our approach aim to provide a high level specification of E-Learning services in terms of user roles, media objects, distribution policies, etc. The introduced Web Service supports creation of tailor-made media streaming applications, using existing software components, e.g., media server or VoIP software components.

The work presented in this paper is also proposal to solve the problem of the dynamic adaptation of multimedia services in heterogeneous device contexts, especially in case of mobile devices. We presented a generic SMIL-based adaptation methodology suitable for the adaptation of multimedia services according to different conditions of access devices and networks.

We report the current status of our prototype. The prototype has shown that our XML based language is well suited for automatic generation of implementations. At implementation level, the different aspects are integrated in a general object-oriented architecture supporting modularity and reuse of software. The deployment of new service operations is very easy to accomplish due to the modular structure of the service oriented architecture and the well designed and easy to use PHP development pack nusoap.

Currently, the evolution of E-Learning is mainly driven by then convergence of two traditionally separate worlds: just in time learning and Web 2.0 tools, such as social networks. Learning takes place through conversations about content and grounded interaction about problems and actions, i.e., employees learn from each other through communities of practice, blogs, wikis, and other forms of self-published content. Another trend that has emerged in E-learning over the past years is that there has been a steady shift away from traditional PCs to mobile devices. The type of mobile learning we expect in the future will be much more akin to performance support - checklists, quick guides and short ‘how to’ videos, e.g., a video illustrating configuration or handling of a complex machine.

Future work will concentrate on the extension of the service model to support additional functionality, e.g., application sharing as well as the integration of authentication services (authentication of users). Another objective is to improve the search functions by a recommendation system to identify interesting E-Learning material based on the current context (e.g., location) and the user's personal ontology, similar to the approach introduced by Woerndl et al. [21].

ACKNOWLEDGEMENTS

The proposed framework is part of a multimedia project, financed by the state government of Baden-Württemberg in Germany.

REFERENCES

[1] M. Zimmermann, "On Demand Learning in Manufacturing Processes - Implementation by Integrated Multimedia Streaming Services", AICT 2010, pp. 106-111, 2010 Sixth Advanced International Conference on Telecommunications, 2010.

[2] B. Khasawneh and S, A. El-Seoud, "Framework for on-demand e-learning resources allocation and distribution", OERAD International Journal of Computing & Information Sciences, Vol. 4, No. 2, 2006.

[3] D. Borissova and I. Mustakerov, "A Framework of Multimedia E-Learning Design for Engineering Training", ICWL 2009, in LNCS 5686, pp. 88–97, 2009, Springer, Berlin Heidelberg, 2009.

[4] M. Hillenbrand, J. Götze, and P. Müller, Voice over IP - Considerations for a Next Generation Architecture. In EUROMICRO-SEAA, pp. 386–395, IEEE Computer Society, 2005.

[5] A. Uyar, W. Wu, H. Bulut, and G. Fox, "Service-Oriented Architecture for a Scalable Videoconferencing System", In Proc. of IEEE Int. Conf. on Pervasive Services 2005, pp. 445–448, IEEE Computer Society, 2005.

[6] W. Wu, G. Fox, H. Bulut, A. Uyar, and T. Huang, "Service Oriented Architecture for VoIP conferencing", In Int. Journal of Communication Systems, Special Issue on Voice over IP - Theory and Practice, volume 19, pp. 445–462, John Wiley & Sons, 2006.

[7] G. Vossen, P. Westerkamp, "E-Learning as a Web Service," ideas, Seventh International Database Engineering and Applications Symposium (IDEAS'03), 2003.

[8] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1", http://www.w3.org/TR/wsdl, 2001

[9] http://sourceforge.net/projects/nusoap/, Retrieved December 29, 2010, from http://sourceforge.net/projects/nusoap

[10] ITU. Recomm. H.323 (1999), "Packet-based multimedia communications systems".

[11] L-L. Wu Chou and L. Feng, "Web Services for communication over IP", In: Communications Magazine, IEEE, Volume: 46, Issue: 3, 136-143, 2008.

[12] T. Yoshimura, "Mobile Streaming Media CDN Enabled by Dynamic SMIL", WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA, ACM 1-58113-449-5/02/0005, 2002.

[13] SMIL: http://www.w3.org/TR/smil20, Retrieved December 29, 2010, from http://www.w3.org/TR/smil20.

[14] D. Fallside, "XML Schema Part 0: Primer", http://www.w3c.org/TR/xmlschema-0, 2001.

[15] Asterisk: The Open Source Telephony Project, http://www.asterik.org

[16] T. Ericson and M. Brandt, "JSR 309 - Overview of Media Server Control API", Public Final Draft, Media Server Control API v1.0, 2009.

[17] A. Scholz, "WS-AMUSE - Web Service architecture for multimedia services", International Conference on Software Engineering, In Proceedings of the 30th international conference on Software engineering, Leipzig, Germany, 2008.

[18] http://java.sun.com/javase/technologies/desktop/media/jmf/index.jsp, Retrieved December 29, 2010, http://java.sun.com

[19] J. M. Oliveira and E. Carrapatoso, "Dynamic Generation of SMIL-Based Multimedia Interfaces", JOURNAL OF MULTIMEDIA, Vol. 3, No. 4, 2008.

[20] RealNetworks, Inc., "Helix Media Delivery Platform for Mobile Networks", RealNetworks White Paper, Series Version 2.1, 2010.

[21] W. Woerndl, G. Groh, and A. Hristov, "Individual and Social Recommendations for Mobile Semantic Personal Information Management", International Journal on Advances in Internet Technology, vol 2 no 2&3, ISSN: 1942-2652, 2009.

[22] Y.-J. Lee., O.-G. Min, and H.-Y Kim, "Performance evaluation technique of the RTSP based streaming server", Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS), 2005.