Integrating the Virtual World: Teaching EAI and e-business Integration at Universities of Applied Sciences

Karl Flieder

Integrating the Virtual World: Teaching EAI and e-Business Integration at Universities of Applied Sciences

Karl Flieder

FH Campus 02 – Visiting Lecturer at FH Joanneum, Universities of Applied Sciences, Graz

Key words: *Web services, e-Business, EAI, technology-enhanced learning, active learning, collaborative learning, cost-effectiveness*

Abstract:

In this paper we report on a technology-enhanced learning approach for teaching integration issues. Virtually available Web services have evolved considerably and teaching enterprise application as well as e-Business integration at universities has become commonplace. However, this evolution is limited by the time that is needed for developing learner-centered applications and, most of all, by the resulting costs. By applying a collaborative learning model and by incorporating shared code, available from the virtual world, we were able to increase the motivational background as well as the learning success of the students. By demonstrating applications with a historical background, we strive for a convergence of natural sciences and humanities.

1 Introduction

Over the past several years, teaching enterprise application and e-Business issues has become a challenge due to the rapidly emerging technologies and due to less financial resources at Austrian universities. A current trend in practical Internet computing as well as in research and development is the strategic focus on *how* to best use the World Wide Web (Web). As a matter of fact, applying comforts provided by the Web such as open source software and tools has become commonplace in the classroom. At the FH Joanneum, the subject "Enterprise Application Integration" (EAI)[1] is embedded in e-Business[2] integration concepts and techniques. It was set up as an integrated course that combines theory and practice (Figure 1). Emerging technologies such as Web service interfaces[3] allow the use of Web 2.0 applications for practical e-Business demonstrations in the classroom. Generally, Web services are an inherent part of a Service-Oriented Architecture (SOA), the actual paradigm for architecting business integration needs. Web services are taken advantage of the public resource Internet

---

[1] *"EAI seeks to eliminate islands of data and automation caused by disparate software development activities and to integrate custom and package applications (including legacy) to drive operational efficiency that enable applications to interoperate effectively."* ([24], p. 502):

[2] According to DAVYDOV [4], *"e-Business encompasses sophisticated business-to-business interactions and collaboration activities at a level of enterprise applications and business processes, enabling business partners to share in-depth business intelligence, which leads, in turn, to the management and optimization of inter-enterprise processes such as supply chain management."*

[3] PAPAZOGLOU and RIBBERS ([24], p. 585) define Web services as follows: *"Web services form the building blocks for creating distributed applications that can be published to and accessed over the Internet and corporate intranets. They consist of a set of open Internet standards that allow developers to implement distributed applications."*
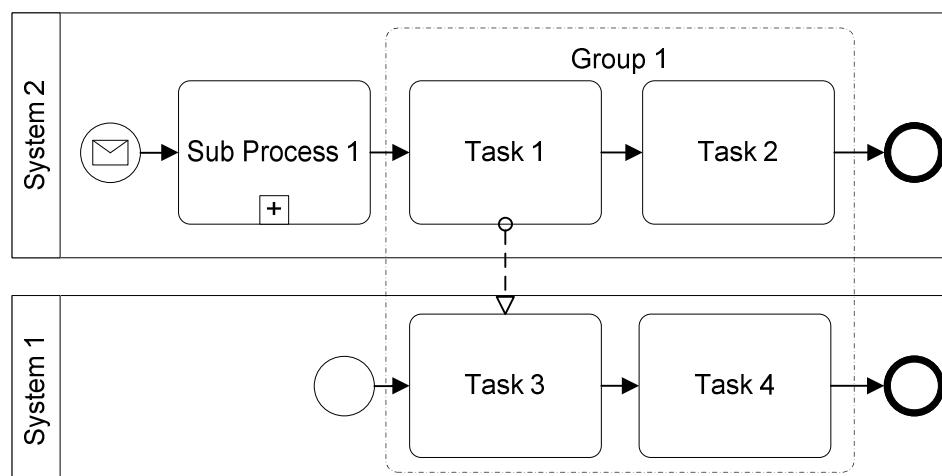
by transmitting SOAP-messages via Hyper Text Transfer Protocol (HTTP). In this way, a seamless integration of intra- as well as cross-enterprise applications can be achieved [23]. Despite some similarities as regards their technological foundations, cross-enterprise e-Business integration and intra-enterprise application integration address different business needs. Fundamentally, e-Business is about creating new services and streamlining business processes by leveraging the company's computing infrastructure. It supports e-Business standards including XML and EDI (Electronic Data Interchange) as well as security issues. Emerging security mechanisms for cross-enterprise communication such as XML Security and WS Security increasingly enable companies to integrate external Web services provided by private or public sources in a secure manner [7]. EAI basically deals with the integration of multiple data sources, heterogeneous systems and applications within the enterprise. Very important from a technical point of view is that public (Web) services may be integrated, too. For the realization of our exercises and of a mandatory and collaborative semester project at our university we make use of tools and Web services, which are publicly available from communities and leading Web 2.0 companies.

Figure 1: Thematic web of the integrated course Enterprise Application Integration (EAI).

A typical example for using EAI and e-Business integration is a workflow that performs event-based, automated as well as human tasks by means of sequences, messages, and data flows between heterogeneous systems. Besides an asynchronous message queuing infrastructure, which is used to ensure a loose coupling between the heterogeneous applications, a professional integration server is needed for this approach in order to realize inter-connections between different data sources, applications and public Web services. In this way, business processes can easily be automated and the *multiple use* [13] of components and compounded applications can be accomplished. In addition to the availability of commercial equipment, we benefit from well-balanced Web service interfaces, provided by the Web 2.0 [17] and sharing one common characteristic: virtual accessibility. Figure 2 shows an abstract workflow scenario, modeled in Business Process Modeling Notation (BPMN) including a sub process, which logically groups activities and different tasks.

Figure 2: A typical workflow scenario including two systems and performing different tasks by means of message and data flows including start- and end-points.

2 Collaborative Learning Model

According to MÖDRITSCHER ET AL. (2004), in a constructivist learning environment, *"the learner plays an active role in the learning process constructing his own knowledge through experiences in a context into which the target domain is integrated."* [14]. From a teacher's perspective it must be said that a 30-hour course (3 ECTS) can only provide a glimpse of what the next generation of intra- and cross-enterprise integration technology for everyday use may look like. Therefore, we aim at inspiring solutions for the students' real-world needs in the (near) future. In order to support critical thinking, analytical strategies, and working with classmates our teaching strategy lays great emphasis on the practical work in groups, based on four activities related to *Active Learning* [19]: (a) collect, (b) relate, (c) create, and (d) donate.

- **Collect:** By learning and working with an enhanced script and by doing selective homework, the students gather the basic information related to the subject EAI in the context of e-Business. For teaching the appropriate theory we also explore enterprise integration patterns [12].

- **Relate:** Learning often improves through the formation of groups that facilitate motivation, communication, knowledge sharing, social skills as well as the development of human and social intelligence. During our exercises, teams of typically three people work and learn together. As a result the students benefit from the collaboration with their like-minded colleagues, who jointly contribute their know-how.

- **Create:** To bring a fusion between the dry theory and its practical application we also work on exercises based on virtually available Web service interfaces. Companies such as Google, Amazon, EBay, or WIPO (World Intellectual Property Organization) provide very useful examples. Software tools, for example a professional integration server as well as open source tools SoapUI [20] and Eclipse [5] allow us to achieve a high level in acquiring abilities for integrating heterogeneous systems and applications.

- **Donate:** The ability to solve complex real world problems is obviously superior to simple experiments. Therefore, the students have to work collaboratively on a so-called semester project. Their main goal is to apply the knowledge gathered and the abilities acquired during two semesters to a real world project. Mastering the scalability and complexity of this project is an additional factor for success.

3   Virtual Toolbox for Reality-based Interactions

In this section we present an open source tool, currently credited as a full-featured product for working with Web service interfaces that extends our commercial vendor toolset. Furthermore, we present examples of well-balanced Web Service Application Programming Interfaces (WS APIs), shared code provided by globally acting technology leaders of the Web 2.0. In addition to our professional integration server, which we use for implementing workflows including BPEL controls [9], we make use of a "virtual toolbox" for teaching basics of asynchronous messaging [10], Web service interaction schemes and basics of Service-Oriented Architectures (SOA).

3.1   SoapUI

The tool SoapUI is an open source desktop application for inspecting, developing, mocking, testing, simulating, and invoking Web services over HTTP. This tool is recommendable for any case of functional testing and for developing Web services. Many of the problems and errors that may occur while working with a WSDL-file (Web Service Description Language), for example problems with a XML schema, can easily be traced. Furthermore, it offers real-time analyses, test cases, load testing, behavioral diagrams, performance statistics, exporting logs, and other useful features. Its partial weakness in security matters serves as a practical example for teaching security in a collateral way. Nevertheless, this tool is a good starting point for reality-based interactions with the virtual world.

3.2   Amazon.com's Simple Queue Service

One of the latest achievements of technology leaders in the virtual world is the Amazon Web Services (AWS) collection. It also provides business integration technology based on Web service interfaces, performing Client-Server communication over the Internet, and allows a tremendous leap forward in teaching integration issues. From a technical point of view, the Amazon Simple Queue Service (Amazon SQS) offers a reliable and highly scalable virtually hosted message queuing system for the handling of messages as they occur in distributed computing. In this system Amazon is responsible for (a) virtualized configuration, (b) virtualized message store, (c) hosted message routing, and (d) virtualized software repository. By taking advantage of this messaging backbone, our students are able to build loosely coupled workflow applications, combining human and automated tasks without the need of a cost-intensive local messaging system. This innovation is a powerful alternative to proprietary and expensive messaging solutions currently used in many enterprises. The functionality provided and supported by this API is basically the same as professional messaging systems offer, for example [2]:

- Creation of an unlimited number of queues for storing messages
- Vesting someone with access privileges for accessing a particular queue
- Adding new messages to a queue at any time, up to 256 KB of text in any format
- Checking a queue at any time for messages waiting to be read
- Pushing messages to – or pulling messages from – a queue for authorized users, with the queue creator determining which users are allowed to perform operations
- Accessibility through messaging standards like REST or SOAP and query interfaces designed to work with any Internet-development toolkit
- Permission features (grants) and a locking mechanism that prevents other computers from trying to process messages while one is processing them
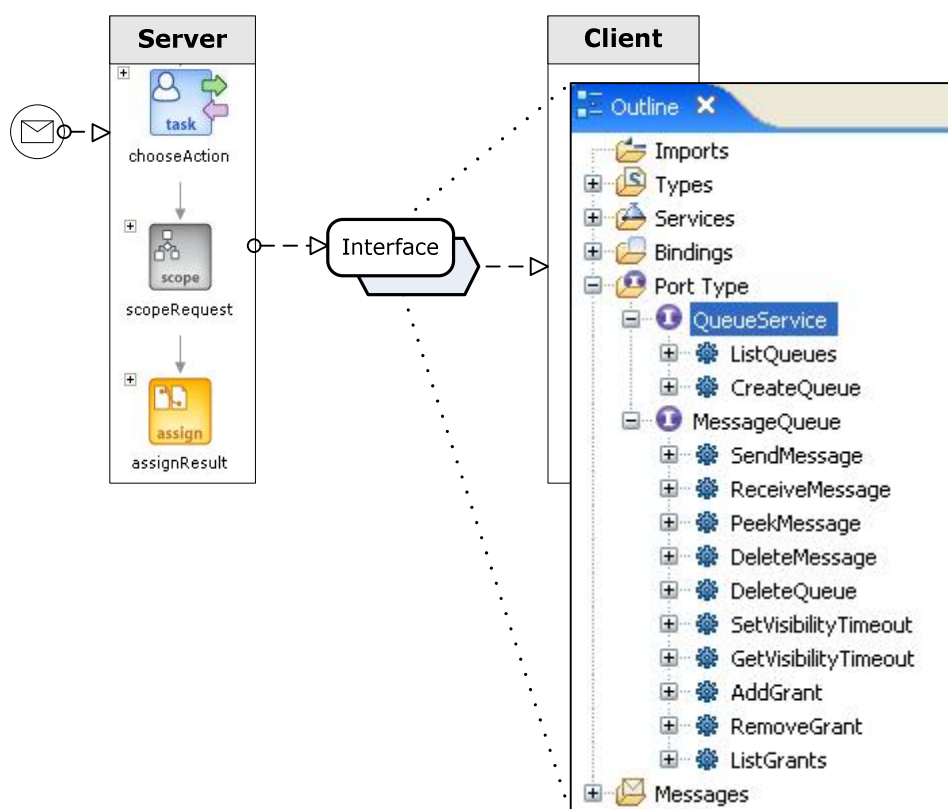
Figure 3: BPEL-based workflow interaction scheme of the Amazon SQS, partly visualizing the WSDL-file.

3.2.1 Basic Security Features

Implementing e-Business solutions that work over the public and insecure Internet requires security efforts on different levels. This is the reason why the Amazon SQS offers basic security features such as *Authentication*. To use this and other WS APIs securely, each request must carry information about the identity of the sender to ensure that requests for a certain resource can only have been produced by authorized senders. Authentication is accomplished by means of a secret access key (`AWSAccessKeyId`, see Figure 4). Since the access key is not protected during the transmission of a SOAP request, additional information such as a request signature – implemented as HMAC (Hash-based Message Authentication Code) – is needed. A more comfortable alternative, especially in an integration scenario with our professional integration server, is to use X.509 Certificates. Unfortunately, Amazon Web services do not implement a full Public Key Infrastructure (PKI). Instead, the X.509 Certificate is only used to authenticate requests. However, to achieve a higher level of security, for example integrity and non-repudiation, it is necessary to implement additional security features.

3.3 *Amazon.com's e-Commerce Service*

Amazon.com's e-Commerce Service (Amazon ECS) exposes Amazon's product data and e-Commerce functionality. It allows developers, Website owners and merchants to leverage the data and functionality that Amazon uses to power its own e-Commerce business. By working with this WS API, our students get the opportunity to rapidly establish e-Business solutions based on well-tested resources that facilitate user participation.
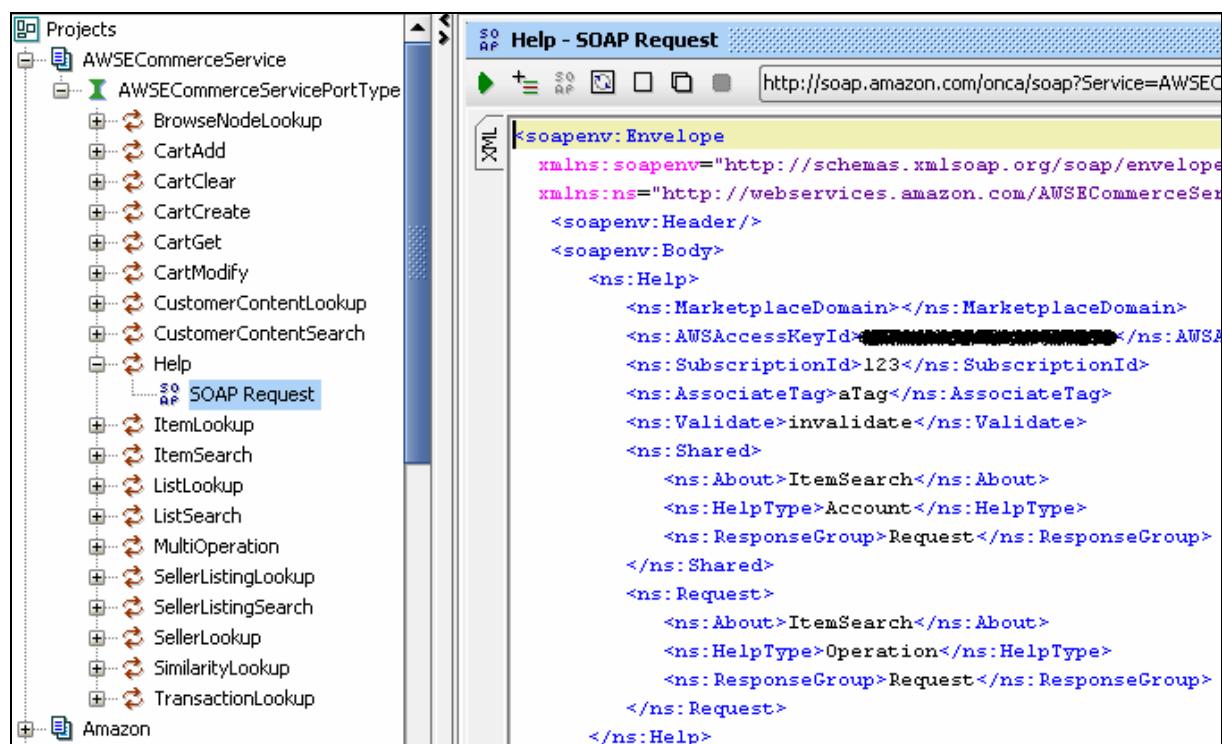
Figure 4: Interface description of the Amazon ECS including operations and a SOAP request visualized in SoapUI.

With the Amazon ECS, developers as well as our students can quickly add rich content and powerful capabilities to Web sites as well as off-line applications by using the following features [2]:

- Detailed product information on all Amazon.com products
- Access to Amazon.com product images
- All customer reviews associated with a product
- Extended search functionality
- Remote shopping cart
- Amazon wish list search
- Precise response groups
- Multi-operation and batch interfaces
- Detailed error messages
- Built-in help functionality

3.4   The Google API

One of the first examples of public Ws APIs, which attracted our students, was the Google search API [11]. It enables our students to automate search requests in Google's index of billions of Web pages (`doGoogleSearch`), to check the spelling of words (`doGoogleSpellingSuggestion`), and to access information in the Google cache (`doGetCachedPage`). At present, this API is only available for applications based on REST (Representational State Transfer), another emerging Web-2.0-based interface technology for integration issues. Depending on one's individual needs, this API might be a better choice instead of the SOAP-API. It tends to be better suited for search-based Web applications and supports additional features like video, news, maps, and blog search results.

For exercise purposes, our students configure workflow applications that connect remotely to the Google services. Internet communication is performed via SOAP messages, an XML-based technology performing Remote Procedure Calls (RPCs). A WSDL-file (Web Service Description Language) is responsible for the formal description of the Web service provided. The SOAP messages that finally invoke a Web service can be generated with tools such as the SoapUI. In Figure 5 – on the left side – three SOAP calls, each of which includes an individual operation and the correlating parameters, are interlinked in parallel. The Web service connector in the middle of the workflow (`ws_google_api`) performs a *request* and subsequently the *response* is mapped into the appropriate message style described by the WSDL-file. The response message received may be processed in any application connected through an additional sub-workflow.
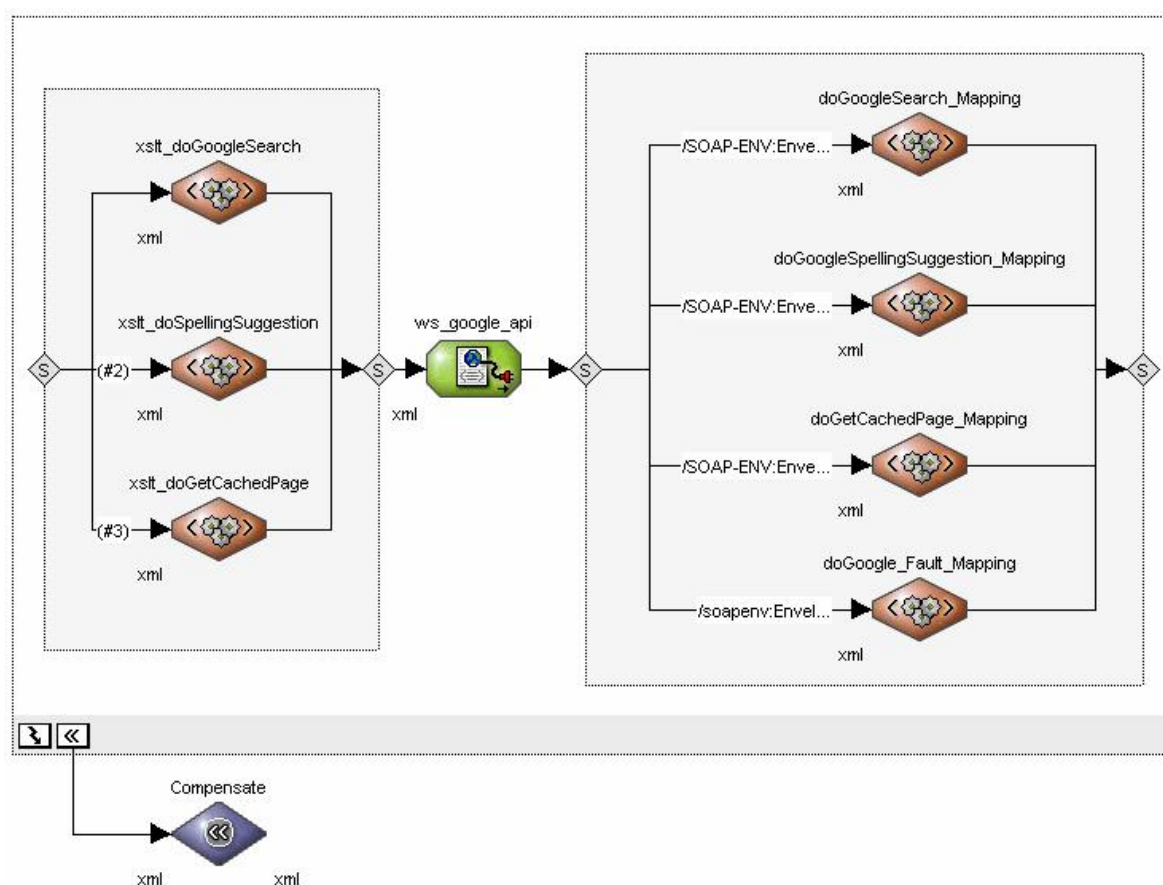

Figure 5: The Google search workflow including XSLT transformations, the Google WS API and BPEL controls such as `Switch` and `Compensate`.

3.4.1 Service Orchestration with BPEL

In addition to workflow controls we use BPEL (Business Process Execution Language) controls to explain the benefits of building automated business processes. A BPEL enhanced workflow consists of a collection of *basic activities* such as invoke, receive, reply, and wait, and of *structured activities* such as sequence, flow, and switch. Structured activities define the assembly of a workflow by combining basic activities, for example web service invocations, into structures that express workflow control patterns [1]. As Figure 5 shows, with BPEL-statements such as `switch` and `compensate` the orchestration of individual service invocations within a workflow scenario for transactional purposes can be accomplished. Since the specification of BPEL builds upon the extensibility support in WSDL, BPEL controls may contain specific attributes and activities defined by the tool provider.

4   Interdisciplinary Challenge

NAMBISAN (2007) states that *"future technologies need a broader understanding of complementary disciplines and fields, superior collaboration skills, awareness of the diverse global social and cultural contexts in which technologies are applied"* ([15], p. 30). It seems that the ability to intertwine different disciplines is a key characteristic of a successful education. We believe that the application of information technology in an interdisciplinary context will supervise a broader understanding of technology in general. By choosing the appropriate tools, interfaces and applications, we strive for this interdisciplinary challenge and try to interlink e-business products, services, applications, and activities with topics such as Artificial Intelligence (AI), the history of automation, or even knowledge management. Many of the modern amenities in the Web 2.0 have the ability to satisfy post-industrial human needs by building applications with them, for example [25]:

- Information empowerment
- Knowledge sharing
- Virtual social communities
- Service enrichment

To absorb the know-how of leading Web 2.0 companies and online communities successfully, and to integrate it systematically into the learning process, it is important to get a basic understanding of the motives behind. We believe that this can be achieved best by applying – or at least looking at – interesting applications in an interdisciplinary context. In this way, different domains such as AI, the history of automation or even knowledge management can be interlinked – technology is mapped according to human needs. The application "The Mechanical Turk", described in the next section, is a wonderful example. It is a Web application for collecting and managing "natural", i.e. human intelligence [21], [6].

4.1   The Mechanical Turk

In the past, there had already been attempts to provide simple messaging systems as well as applications related to Artificial Intelligence (AI) [16]. However, comfortable, usable infrastructures and interesting applications emerged during the advent of the Web 2.0. Amazon's messaging backbone SQS, introduced in section 3.2, is such an example. Meanwhile, the virtual community has realized applications with it. The Mechanical Turk implements an application related to AI. Embedded in an interdisciplinary context, this application works well as an example for our interdisciplinary approach. Usually, a human being requests a task while the computer completes it and provides the results. In contrast, by using Amazon's SQS messaging backbone, the Mechanical Turk reverses this process and asks a human being to perform a task. The man behind the technology returns the result and completes the task. Typically these tasks are extraordinarily difficult for computers, but simple for humans to answer.

4.1.1   Philosophical Implications

Following the philosophical roots of the Mechanical Turk logically, we reach a field generally known as natural language processing. Closely linked to it is a philosophical question, first coined by ALAN M. TURING (1950) [22]: *"Can a Machine Think?"* TURING assumed that the definition of computer intelligence would be answered by using the following conjecture: *"It is proposed that a machine may be deemed intelligent, if it can act in such a manner that a human cannot distinguish the machine from another human merely by asking questions via a mechanical link."* TURING'S suggestion was, that if the responses from the computer were

indistinguishable from those of a human, the computer could be said to be thinking. The Turing test, however, is controversial and SEARLE counter-argued with the Chinese Room experiment [18], [8].
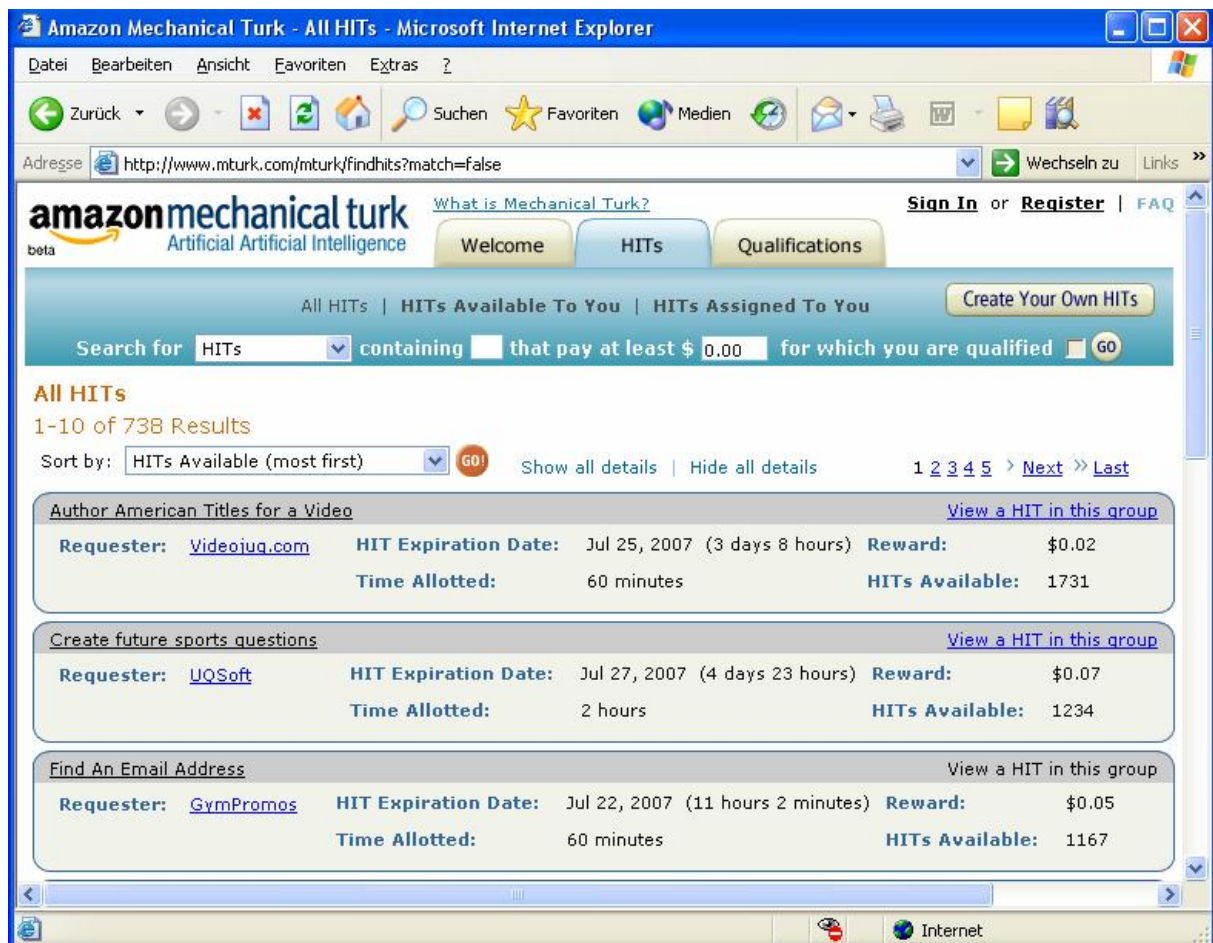

Figure 6: The Mechanical Turk application based on the Amazon SQS.

5   Values Perceived

By working with several WS APIs, with the tool SoapUI as well as with our professional integration and e-Commerce servers, many technical concepts related to e-Business integration can easily be demonstrated. Additionally, economic concepts such as Value Added Networks (VANs) can be explained my means of Amazon.com's e-Business and Simple Queue Service (Amazon SQS) infrastructure. Moreover, our students get prepared to master collaboratively their semester project, i.e. implementing a real world e-Business problem. The presented example of the Google search workflow in section 3.3 provides an overview of how the basic connectivity through Web service interfaces, the use of workflow elements for building automated tasks and the orchestration of different Web services by means of BPEL controls can be achieved.

From an economic point of view, the big advantage of virtual WS APIs is, that the principles of asynchronous messaging, a key to success in EAI and e-Business integration projects, might be told without the need for establishing a local and therefore expensive messaging infrastructure. Instead, basics of asynchronous messaging, necessary security mechanisms, Web service interaction schemes as well as functional testing can be demonstrated by means

of publicly available and cost-effective resources. This is the reason why some of the students get motivated to occupy themselves with this kind of technology even in their leisure time.

Once the exercises are implemented, it is particularly important to examine whether they actually achieved the desired results or not. We have found that utilizing our teaching model enables the students to create an enormous pool of knowledge and creativity. Different sets of motives drive the students to create their own exercises on an advanced level. The desire for understanding this kind of state-of-the-art technology and the opportunity to build unique solutions for their own are the students' primary motives.

An example of how technical knowledge, i.e. applying security in virtual environments, and philosophical implications can be intertwined is the following: While the Turing Test (TT) at present seems to be a more or less unsolvable problem, the Reverse Turing Test (RTT), or Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHA), has become commonplace at Web sites like Yahoo!, Hotmail, and PayPal for security reasons [3].

6   Conclusion and Outlook

In this paper we have introduced some of the latest achievements on the Web development frontier. We have presented an e-Business infrastructure based on Web services as well as an advisable open source tool. The WS APIs introduced are highly suitable for teaching issues in the fields of Enterprise Application Integration (EAI) and e-Business integration. This was the main reason why we adopted them for our collaborative learning model. We believe that this approach is an excellent way to prepare the next generation of engineers for the emerging global innovation environment, because they will most likely require a much broader perspective of how innovative technologies as well as solutions are developed. After finishing their studies, some of the forward-thinking students will integrate, automate – simply streamline core internal and external business processes, while others will rather engage in business development or even in research activities. For all of them e-Business and application integration will be a good preparation for their daily business in the future.

References:

[1] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P. (2003): Workflow Patterns. Distributed and Parallel Databases, 14(1), 5-51.

[2] Amazon.com Web services (2007): http://aws.amazon.com (2007-09-04).

[3] Chakrabarti, S., and Singbal, M. (2007): Password-based Authentication: Preventing Dictionary Attacks. Computer 40 (2007) 6, 68-74.

[4] Davydov, M. M. (2001): Corporate Portals and e-Business Integration. McGraw-Hill.

[5] Eclipse (2007): http://www.eclipse.org (2007-06-19).

[6] Flieder, K. (2007): Kempelens „Schach-Türke": Seine Renaissance im Web 2.0. Wissensmanagement – Das Magazin für Führungskräfte 6/2007, 10-12.

[7] Flieder, K. (2007): Sicherheit in der Prozessintegration mit Web Services und SOA. In Horster, P. (ed.): DACH Security 2007.

[8] Flieder, K. (2007): The Yijing – The Philosophical Foundation of the Binary System? In Böszörmenyi, L. (ed.): MEDICHI 2007 – Methodic and Didactic Challenges of the History of Informatics, OCG.

[9] Flieder, K. (2006): Geschäftsprozessmanagement – Geschäftsprozesse mit dem virtuellen Reißbrett steuern? WING-business 39 (2006) 3, 35-38.

[10] Flieder, K. (2005): Testing and Visualizing a Message Queuing Infrastructure. Proc. of WMSCI'05, Orlando, 51-57.

[11] Google Search API (2007): http://api.google.com/GoogleSearch.wsdl (2007-09-04).

[12] Hohpe, G. and Woolf, B. (2003): Enterprise Integration Patterns. Addison-Wesley.

[13] Makolm, J., and Orthofer, G. (2006): Multiple Use. OCG-Journal 31 (2006) 2, 28-29.

[14] Mödritscher, F., Garcia-Barrios, V. M., Gütl, C (2004): The Past, the Present and the Future of adaptive E-Learning: An Approach within the Scope of the Research Project AdeLE. ICL Villach.

[15] Nambisan, S. (2007): How to Prepare Tomorrow's Technologists for Global Networks of Innovation. Communications of the ACM, 48 (5), 29-31.

[16] O'Leary, C. (2003): Technology for automated assessment: The World-Wide-Mind. Proc. 4th Irish Educational Technology Conf. – EdTech-03, Waterford.

[17] O'Reilly, T. (2005): What Is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (2007-09-04).

[18] Searle, J. R. (1980): Minds, Brains and Programs. The Behavioral and Brain Sciences, 3 (3).

[19] Shneiderman, B. (2003): Leonardo's Laptop. The MIT Press.

[20] SoapUI (2006): http://www.soapui.org (2007-09-04).

[21] The Mechanical Turk (2007): http://www.mturk.com/mturk/welcome (2007-09-04).

[22] Turing, A. M.: (1950): Computing machinery and intelligence. Mind, 59 (236), 433-460.

[23] Papazoglou, M. P., and Georgakopoulos, D. (2003): Service-Oriented Computing. Communications of the ACM, 46 (10), 25-28.

[24] Papazoglou, M. P., and Ribbers, P. M. A. (2006): e-Business: Organizational and Technical Foundations. Wiley. 2006.

[25] Zhong, N., Liu, J., and Yao, Y. (2007): Envisioning Intelligent Information Technologies through the Prism of Web Intelligence. Communications of the ACM, 50 (3), 89-94.

Author:

Karl Flieder, Dipl.-Ing. (FH)
Fachhochschule Campus 02, graduate – degree program in IT and IT-Marketing
Fachhochschule Joanneum, visiting lecturer – degree program in Information Management
8010 Graz, Austria
E-Mail: eai@karlflieder.at