Teaching Business Informatics with Web 2.0 Technologies

Karl Flieder[1]*, Felix Mödritscher*[2]

[1]Academia Nova, Schwechat, Austria,
[2]Vienna University of Economics and Business (WU), Vienna, Austria

Key words: *Business Process Management (BPM), Collaborative Learning, e-Business, Web 2.0, Web Service*

Abstract:

In this paper we report on a collaborative learning approach for teaching Business Process Management (BPM) with a focus on technologies and concepts of the Web 2.0. We explore the virtual continent – the cloud – for web service interfaces and applications related to e-Business in a broader context by following the core definitions of the Web 2.0. In doing so, we strive for an interdisciplinary challenge and investigate an application based on a historical idea. By following the evolution of the Web until we arrive at what is known as the Web as platform. With this metaphor in mind, we aim at interlinking the evolution of the Web and its applications with traditional aspects of computer science.

1 Introduction

A current trend in practical Internet computing as well as in research and development is the strategic focus on how to use the World Wide Web (Web) best in different application areas. As a result, applying the benefits provided by the Web 2.0 has become commonplace in higher education (Gillet, Law, & Chatterjee, 2010; Safran, Helic, & Gütl, 2007). In this chapter we report on our pedagogical approach to teaching Business Process Management (BPM) at a university of applied sciences in Austria.

Amongst others, we make extensive use of the achievements of the Web 2.0: the Web as platform, interconnected services, and applications of social software. For interacting with the virtual continent, we put emphasis on publicly available web services. Their advantages are manifold: They provide well-proven solutions, act as a technical reference for our students, and give insights into applications that support a collaborative learning environment. By applying an adapted model of Active Learning we aim at a broader understanding of complementary disciplines, at superior collaboration skills, and at an increased awareness of the social and cultural context in which technology is applied. We believe that this can be achieved best by utilizing – or at least looking at – interesting applications available on the Web. Moreover we focus on Web 2.0 technologies that are usually invisible but provide interoperability and flexibility for users as well as for software applications that rely on an effective machine-to-machine communication in the background.

In this paper, we introduce web services as machine-to-machine communication technology also suitable for the interaction with collaboratively created content, and – of course – for teaching e-Business integration techniques. The following section provides an overview of the background and work related to e-Business and Business Process Management. Moreover, we introduce our pedagogical setting, which is situated in an interdisciplinary context and incorporates a collaborative learning approach. In the main part we put our model into practice. We explore the Web 2.0 for interesting web services in order to interconnect with them and we make use of applications based on a reliable machine-to-machine communication. Some of the examples introduced are related to language acquisition in a broader context, others have the ability to satisfy post-industrial human needs by building applications with them (Zhong, Liu, & Yao, 2007). The Mechanical Turk is an example for managing “natural”, i.e. human intelligence (Flieder, 2007a) by means of a web application. In this way, different domains such as artificial intelligence, the history of automation or knowledge management can be interlinked. As a matter of fact, technology is being mapped according to human needs.

2 Background and Technical Context

Web 2.0 technologies represent a quite revolutionary way of managing and repurposing/remixing online information and knowledge repositories (Boulos & Wheeler, 2007). In order to understand the functionality behind interactive content generation and its consumption it is also important to have a closer look at the underlying technologies. Web services are a basic technology of the Web 2.0. Unfortunately, the term “web service” is subject to ambiguous interpretation. In the context of this paper we use this term for an interface technology that is related to Service Oriented Architectures (SOA) consisting of an XML-based service description and SOAP messages, which can be generated on the basis of the service description (Web Service Description Language, WSDL). In workflow scenarios, which implement automated and manual business process activities, web services act as entry points – as interfaces – to heterogeneous system environments. Optionally, an UDDI directory (Universal, Description, Discovery, and Integration) can be used to find web services dynamically. Figure 1 shows the basic interaction model of a web service in the context of service oriented architectures.

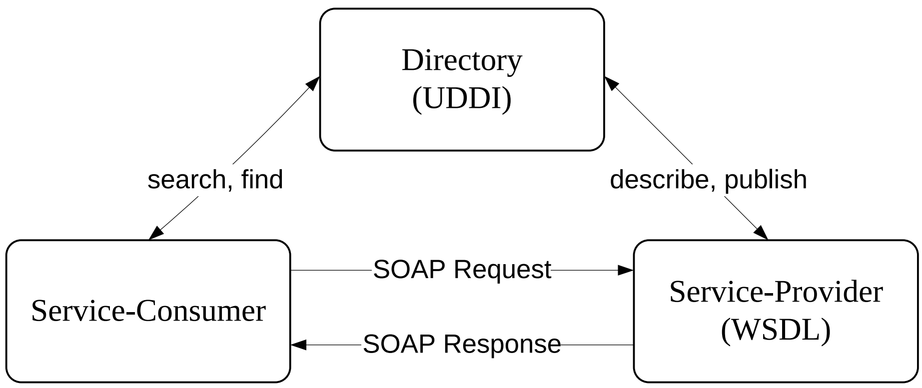

Fig. 1. Web service interaction model.

SOAP specifies the XML (eXtensible Markup Language) structure for the messages exchanged between the service provider and its consumers. A SOAP message acts as an envelope for domain-specific XML business documents that include the semiotic structure for the information exchanged. Due to the modularity and flexibility of web services, SOAP messages are ideal for e-Business application integration because of their ability to (a) complete tasks, (b) conduct business transactions, and (c) solve complex problems (Papazoglou & Ribbers, 2006).

Business Process Management (BPM) is embedded in the context of e-Business and includes concepts as well as technologies for integrating an organization's core business applications such as Production Planning System (PPS), Manufacturing Execution System (MES), Enterprise Resource Planning (ERP), Supply Chain Management (SCM), and legacy systems. In this way, EAI also enables connectivity to suppliers, partners, and customers. The evolution of BPM technology and related concepts such as service oriented architectures that use web services as inherent interface technology (Erl, 2005; Legner, 2007) as well as the recently emerging Web 2.0 paradigms ("the Web as platform") are shifting attention to web services publicly available. From a historical viewpoint, the intra-enterprise perspective is most common. However, SOA and web services constantly shift the focus of attention towards inter-organizational process integration (Legner, Vogel, Löhe, & Mayerl, 2007) and towards the integration of public services. This evolution shows parallels to the evolution of the Web. As a matter of fact, we can find characteristics in contemporary EAI that match O'Reilly's (2005) definitions of the Web 2.0:

- The Web as platform
- Open source development
- Interconnected services
- Architecture of participation, incorporating social software

Web services serve as an ideal starting point for teaching enterprise e-Business and application integration techniques as well as concepts, because they cover two of the four most important service design principles of service oriented architectures: interface orientation and interoperability (W3C, 2004). Apart from classical Web 2.0 technologies – like wikis, blogs, mashups, podcasts, social book marking, collaborative tagging and RSS feeds – web services are an interface technology for machine-to-machine communication. They form the basis for using the Web as a platform and foster interactivity between heterogeneous systems as well as between web applications. Moreover, applications for the exchange of learning and business content can be established by means of messaging middleware based on web services. Concerning interoperability that primarily deals with standardization issues, a number of fundamental artifacts describing technical and business interoperability can be derived (Legner & Wende, 2006). The business content varies depending on the domain of application, whereas the technical standards – the message types and their formal structure – should be uniform in order to enable machine-to-machine integration between different application domains.

3 Pedagogical Setting

According to Böszörmenyi (2007a) "*the real goal of learning is not to agglomerate knowledge, but to keep fresh our capacity to learn new things, to receive new ideas.*" In this sense, our pedagogy aims at a collaborative learning environment through *Active Learning* in which the learner plays an active role in collecting, analyzing, composing, and evaluating information, knowledge, and artifacts (Wade & Ashman, 2007). Consequently, our pedagogical setting affects the learning process as well as the learning success of the students. One particular artifact result is the usage of publicly provided web services related to different application domains such as knowledge management, language acquisition through dictionaries as well as the use of search engines by means of Application Programming Interfaces (APIs). In addition to our technical goals, we strive for an interdisciplinary context and provide the opportunity to gain learning experience in cross-disciplinary fields by incorporating applications of the Web 2.0.

3.1 Active Learning

In order to support critical thinking, analytical strategies, and working with classmates we adopted Active Learning, originally introduced as a framework for mega-creativity (Shneiderman, 2002; see also Coughlan & Johnson, 2006). This collaborative learning approach promotes the idea that learning through practical work is more than a repetitive application of rules – it is rather a reinforcing process. Active Learning comprises four parts, each of which helps us to achieve our goals: (a) *collect*, (b) *relate*, (c) *create*, and (d) *donate*.

- **Collect.** By learning and working with an enhanced script, slides and appropriate papers as well as by doing selective homework, our students gather the basic information related to the subject EAI. In order to teach the basic concepts in a system-independent way we investigate, amongst other sources, enterprise integration patterns (Hohpe & Woolf, 2004) and workflow patterns (van der Aalst, ter Hofstede, Kiepuszewski, & Barros, 2003)

- **Relate.** It can be observed that group work often helps to improve the learning process. Working in teams facilitates motivation, communication, knowledge sharing, social skills and the development of human and social intelligence. During our exercises, the students usually work together in teams of three. As a result, each student benefits from the collaboration with like-minded colleagues and from the exchange of know-how within the group.

- **Create.** To bring a fusion between the theory and its practical application we also work on exercises based on the web service interfaces available via Internet. Besides a professional business integration server, we make use of different web services and open source tools. Global technology leaders such as Google, Amazon, Aonaware, and others provide well-proven implementations of web services. Selected open source tools help us to successfully acquire the abilities for integrating heterogeneous systems and applications at low costs. Moreover, our students get prepared for the semester project, i.e. implementing a real world e-Business problem.

- **Donate.** The ability to solve complex real world problems is obviously superior to simple experiments. This is the reason why our students have to work collaboratively on a so-called semester project. The main goal is that the students apply the knowledge gathered and the abilities acquired during two semesters to a real world project. Planning and implementing their own solution is one of the core requirements, mastering the scalability and complexity of this project during all stages is another factor for success.

By applying Active Learning we aim at a broader understanding of complementary disciplines, at superior collaboration skills, and at an increased awareness of the social and cultural context in which technology is applied. With this constructivist approach we are able to provide an active, stimulating, and authentic learning environment. The adoption of Active Learning as an innovative pedagogic strategy combined with our taxonomy of learning objectives results in a more learner-oriented way of learning. In daily practice, learning-by-doing and different problem-solving strategies complemented declarative knowledge successfully. Table 1 shows the four steps of Active Learning together with correlating didactical questions, the resources used, learning paradigms and Durand's (1998) adapted competences model. Additionally, we mapped our *taxonomy of learning objectives* (Bloom, Engelhart, Furst, Hill, & Krathwohl, 1956) to our learning model and complemented it with estimated degrees of difficulty.

Didactical Questions	**Active Learning**	**Resources / Media**	**Learning by**	**Competences (through)**
How to teach?	*Collect*	Domain-specific information (script, slides, papers)	Recalling learning data	Know-how / skills (information)
For which purposes?	*Relate*	Resources provided by the Web 2.0	Analyses, sharing ideas	Attitude / behavior (communication)
How to improve?	*Create*	Building own workflows	Syntheses, doing	Abilities / skills (innovation)
How to measure learning?	*Donate*	Mastering a collaborative project	Problem-solving	Competence (dissemination)

Table 1. Comprehensive overview of our learning model.

3.2 Interdisciplinary Challenge

Related to innovation issues, 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"*. For this reason we argue in favor of an interdisciplinary context in e-Business integration. Historical ideas that lead to applications of Computer Science (CS), for example, should be part of the theory on which tertiary education in informatics is built. We strive for this challenge and believe that the application of integration technologies in an interdisciplinary context will be the better choice in order to teach a broader understanding of the related ideas, concepts, and technologies.

At first glance, it seems that informatics in general and Business Process Management (BPM) in particular has nothing to do with language acquisition. Informatics as a science is an offspring of mathematics, electrical engineering and physics as well as of economics, operational research and cybernetics. Etymologically, informatics can be derived from *inform*ation and auto*matic*. Algorithms and data structures, their implementation in informatics systems, and the theory associated with them constitute an important (and undisputed) part of informatics (Humbert, Micheuz & Puhlmann, 2007). This viewpoint reduces informatics to the aspect of data processing. However, there are more dimensions of informatics.

According to the philosopher Emanuel Kant, a human being is an autonomous individual, free to make its own decisions. We formulate and use specific words and phrases in order to express our (individual) thoughts, we express wishes, desires, urges and we enunciate ideas and theories – mental actions autonomously done by our mind (Selzer, 2007). In any case, nobody is programming us; instead technology needs to be adapted to human requirements.

4 Practical Application through Interconnected Services

The course Business Process Management covers a wide range of topics. The ongoing convergence of intra- and inter-enterprise application integration was one of the reasons why we put emphasis on web services and their related application in the Web 2.0. We make use of mature web service interfaces that provide access to the "cloud" This term represents the

Web as platform, applications of social software and many others. Other reasons for relying on achievements of the Web 2.0 in the classroom are their cost effectiveness, and most important, their well-proven status. These features fit well into our surrounding conditions and match with our taxonomy of learning objectives. In order to get acquainted with the complexity of web service techniques and to evaluate alternatives it is important to support our learning approach by implementations that have the ability to serve as a reference and that offer useful benefits.

Interaction with resources of the Web 2.0 typically can be performed via web sites, which are built upon Application Programming Interfaces (APIs) responsible for the communication with a data provider in the background. The portal 24hnews.net, for example, provides an overview of many different types of practical and popular APIs that can be used for interacting with Web 2.0 applications. According to our learning objectives, we aim at providing examples of web services that support language learning. Amongst others we used the Google search API and the Aonaware API, each of which is central to our learning model, for incorporating public web service interfaces in the classroom.

One of the main concepts for integrating heterogeneous systems and applications is the principle of unification, typically reached through standardization on different levels: common domain-specific data models, based on the enterprise integration pattern *canonical data model* (Hohpe & Woolf, 2004) to minimize dependencies between applications that use different data formats. A unique message format such as SOAP supports interoperability between heterogeneous systems and a common public process model for integration requirements on process level (Legner & Vogel, 2007). Unfortunately, some web service providers – amongst them a very well-know company – decided *not* to follow common standards. For the students it is important to know how to handle both situations: integration based on common standardized SOAP messages as well as integration based on proprietary solutions that do not support common standards. For showing the different effects, we compare the Google search API (standard) with the Aonaware API (non-standard). In the following sub-sections we provide an overview of the capabilities of both.

4.1 A Web Service for Dictionary Lookup Services

Dictionaries are often accessed electronically via web sites, but human interfaces are not the best choice for automating workflows that include access to dictionaries. Web services offer a solution to this problem. The Aonaware API, for example, provides access to many different dictionaries, amongst others to The Collaborative International Dictionary of English, also referred to as CIDE. This dictionary was derived from the Webster's Revised Unabridged Dictionary Version and from WordNet, a semantic network created by the Cognitive Science Department of Princeton University. It is updated and supplemented by an open coalition of volunteer collaborators from all over the world.

One of the reasons for working with this particular API in our class is the non-standardized character of its SOAP message, manifested in the interface description (WSDL). In contrast to the standardized Google search API, we simply aim at showing the difference. The SOAP operation “Define” (Figure 2) provides lookup functionality for querying a particular dictionary or – in case of the submitted parameter “any”– all the available dictionaries. In contrast to standardized SOAP operations, the name of the operation has to be supplied by means of the BPEL control “variable” that is sent to the server outside the SOAP message as encoded URL (Uniform Resource Locator) within the Hyper Text Transfer Protocol (HTTP).

In our example (Figure 2), we launched a lookup in the dictionary CIDE for the word “lilac”, the English translation of the first author’s surname (Flieder).

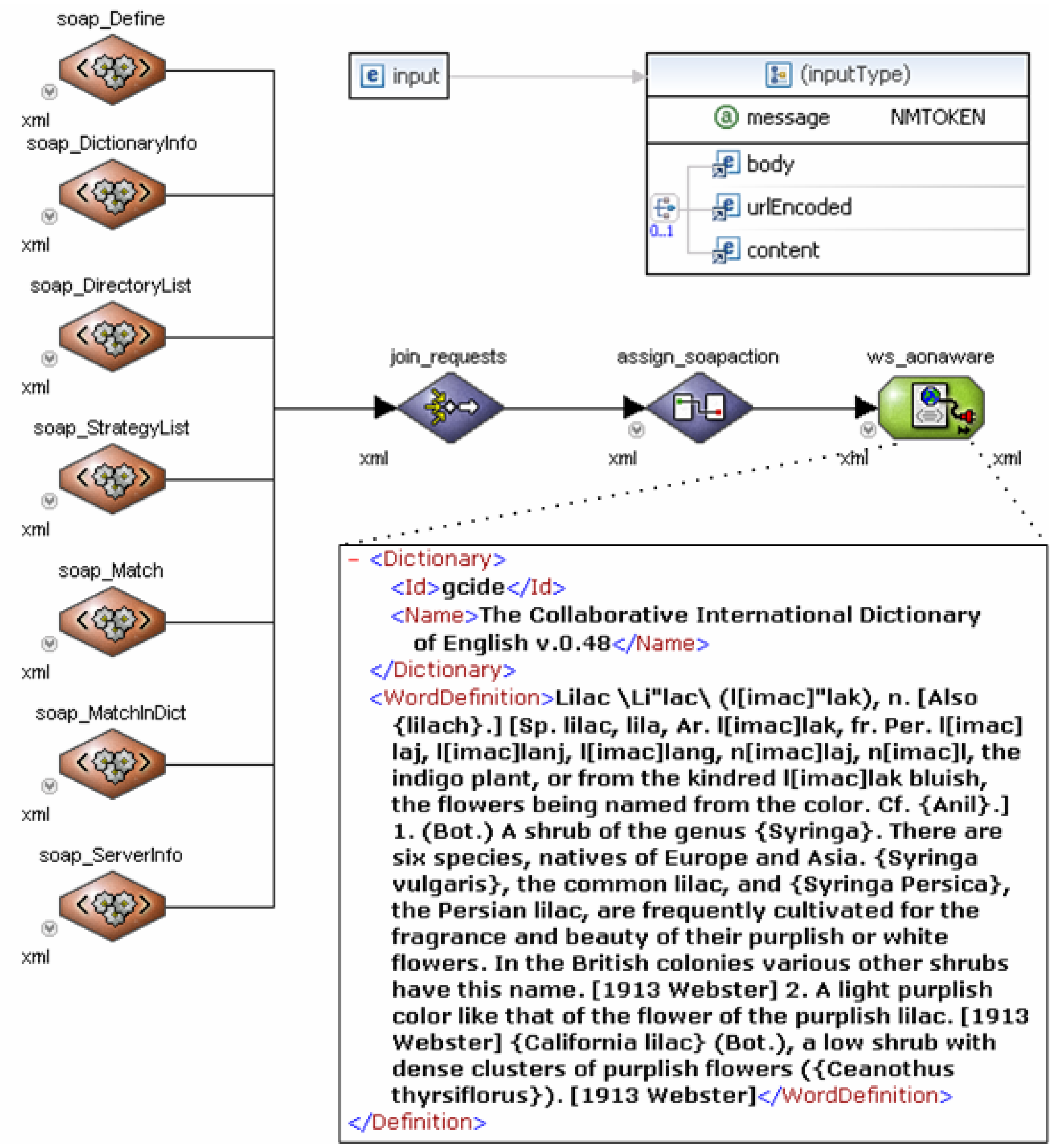

Fig. 2: A workflow based on the Aonaware web service API with a parameter setting for a SOAPAction and the schema of a SOAP request.

With the Aonaware web service application programming interface, our students have to launch the requests to the web service in a particular order for finally getting the desired information. For example: In order to find the parameter for the operation “MatchInDict” they firstly have to request the “Directory List” and the “Strategy List”. The responses provide the students with possible parameters for other operations such as “Match”. For students who are more interested in technology than in linguistics this API provides the operation “Server Info”. The correlating response shows key performance indicators of the hosted dictionaries.

4.2 Amazon.com's Simple Queue Service

Within the domain of Business Process Management and related topics we cannot solely rely on synchronous request-response interactions for interacting with the Web 2.0. Instead, we need asynchronous messaging solutions that also support persistent messaging. In the past, there had already been attempts to provide simple messaging systems as well as applications related to artificial intelligence (O'Leary, 2003). However, comfortable infrastructures and interesting applications emerged with the advent of the Web 2.0. From a technical point of view, the Web as platform became reality with Amazon's Simple Queue Service (Amazon SQS). This messaging backbone 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. The Amazon SQS offers a reliable and highly scalable, virtually hosted message queuing system for the handling of messages as they occur in distributed computing. This system is responsible for (a) virtualized configuration, (b) virtualized message store, (c) hosted message routing, and (d) virtualized software repository.

By taking advantage of this virtually hosted messaging backbone, our students have the tool to build loosely coupled and message-based workflow applications for combining human and automated tasks without the need to install a cost-intensive commercial messaging system (Flieder, 2005). Hence, this innovation is a powerful alternative to proprietary and expensive messaging solutions currently used in many enterprises. Implementing e-Business solutions that work over the public and insecure Internet requires security efforts on different levels (Flieder, 2007b). This is the reason why the Amazon SQS offers basic security features such as authentication. To use this and other web service APIs securely, each request must carry information about the identity of the sender to ensure that the requests for a certain resource were made by authorized senders. Authentication is accomplished by means of a secret access key. 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 integration scenarios with our 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.

By means of the Amazon SQS as an infrastructure and other APIs, we can develop applications of the Web 2.0 that have the ability to satisfy post-industrial human needs: (a) information empowerment, (b) knowledge sharing, and (c) virtual social communities (Zhong, Liu, & Yao, 2007). An excellent example is the application Mechanical Turk, described in the next section.

4.3 The Mechanical Turk Application

Although we introduced web services for automating machine-to-machine interactions, Amazon's implementation of the Mechanical Turk at last shatters the notion that web services have nothing to do with people. The purpose of the "Amazon Web Service Mechanical Turk" (AWSMT) is to provide software developers with a programmable interface to a network of humans in order to solve problems and to incorporate human intelligence into their applications. Developers use the AWSMT interface to submit tasks programmatically to the

Mechanical Turk application, to approve completed tasks, and to incorporate the answers into their software application.

The idea for the interface and the web application is based on a historical illusion manifested by the 18th-century inventor Wolfgang von Kempelen (Flieder, 2007a). He was responsible for creating one of the most famous illusions in history – the Turk. It was constructed in 1769 and presented at the court of the Austrian Empress Maria Theresia: A figure dressed in a Turkish costume sat at a desk on which a chessboard was placed. After demonstrating that no one was concealed inside the automaton, von Kempelen proceeded to wind up the mechanism and get the machine going. Against all expectations it would invariably win the chess game. Publicly promoted as an automaton, the Turk was in fact a mechanical illusion that allowed a human chess master to hide inside and operate the machine. Nevertheless, due to his numerous inventions, such as using the principle of magnetism for information transfer or his mechanical language generation, he is regarded as one of the principle forerunners of information science (Böszörmeny, 2007b). From Kempelen’s illusion we can learn that human judgements and intelligence are often much more effective than machines in solving certain types of problems. Examples in computing are – amongst others – finding specific objects in pictures, evaluating beauty, or translating text.

Amazon’s implementation of this historical idea as a web-based application for managing human knowledge by means of web services helps our students to follow the metaphor “Turk” top-down from the idea to the coding level. Embedded in an interdisciplinary context, this application works well as an example for our learning approach. In computing a human being usually requests a task while the computer completes it and provides the results. The Mechanical Turk reverses this process and asks a human being to perform a task. The automated system mediates between the computer and the people doing the work. The requesters interact with the Mechanical Turk by means of its web service interface (Figure 3). Human beings perform fine-grained work units, so-called HITs (Human Intelligence Tasks). The man behind the technology returns the result and completes the task. Typically these tasks are difficult for computers, but simple for humans to answer. Translating paragraphs of text from one language to another is a common example. Amazon identified a number of internal tasks, amenable to high-volume processing by a workforce composed of individuals with particular skills. This was the main reason for implementing this application. Japanese text, for example, can be written left to right or top to bottom. As part of Amazon’s effort to create searchable indices from scanned images of book content, the text recognition system must be informed of the text direction (Barr & Cabrera, 2006). Against some expectations, humans can do this better than computers.

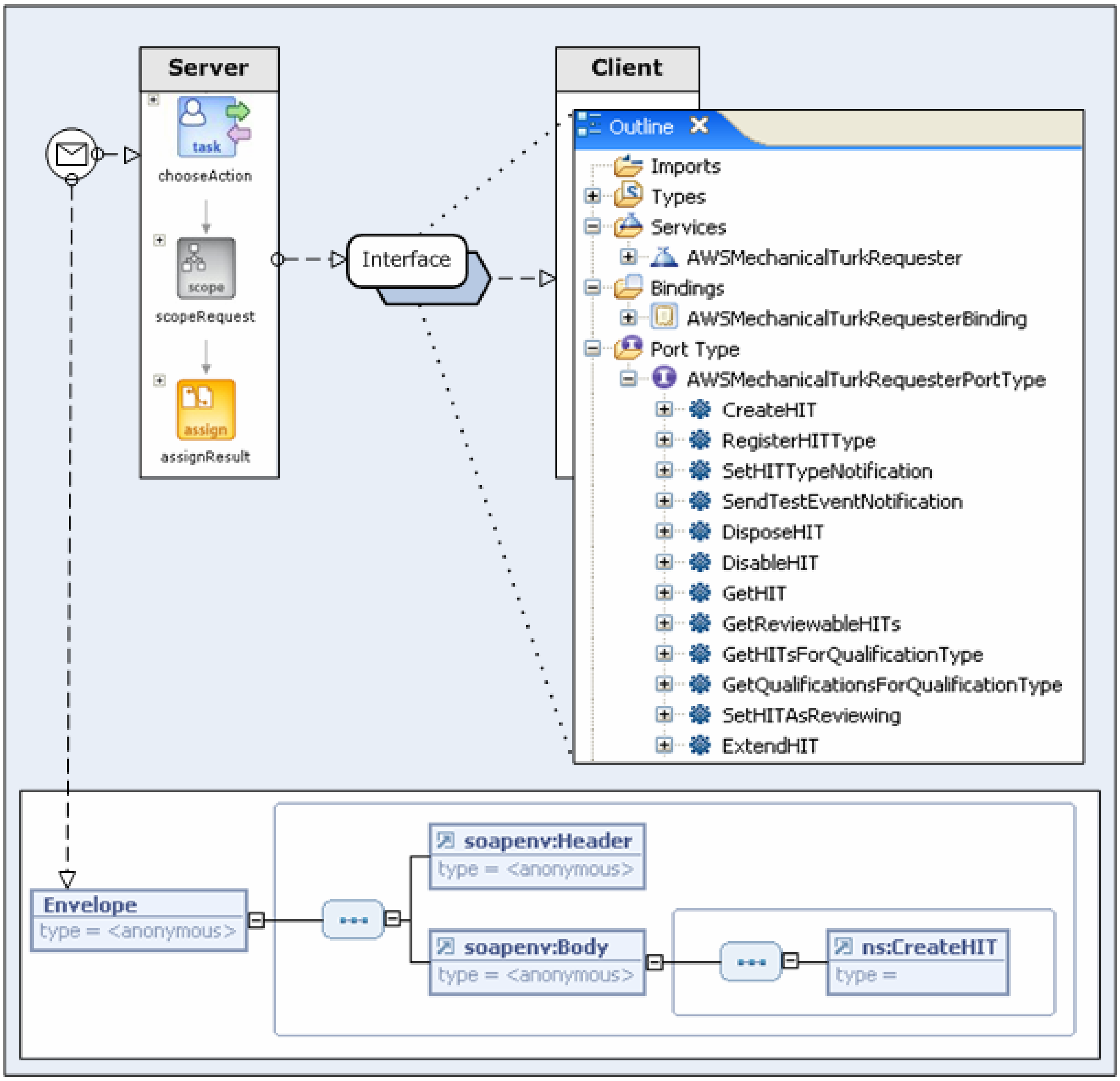

Fig. 3: The Mechanical Turk web service interface exposing the method CreateHIT.

4.4 Evolution towards the Web as Platform

From the perspective of a structured invention process using patent databases such as the Theory of Inventive Problem Solving (TRIZ), evolution follows a certain structure (Mann, 2002). In TRIZ, about 40 inventive principles serve as patterns for the evolution of innovation, for example:

- **Principle of mechanics substitution.** Mechanical means get replaced with sensory means (optical, acoustic, or via radio frequency) – e.g. wireless data transmission between computer systems (Mann & Catháin, 2001).

- **Principle of extraction.** In software: Text in images gets extracted to eliminate background image information and to highlight or identify the regions of the image that contains text (Rea, 1999).

- **Principle of segmentation.** Moveable fields follow static ones, the degree of fragmentation or segmentation gets increased (e.g. multi-pane windows, mashups). The surface segmentation trend, for example, shows the evolution from *immobile* to *single joint* to *multiple joints* to *completely flexible* to *liquid/gas* to *field* (Mann & Catháin, 2001).

From the principle of segmentation we can also derive simplified evolutionary steps related to distributed computing in general and to EAI in particular: Evolution follows a pattern from *single systems* to *multiple (distributed) systems* to *systems within a field*. Replacing the word "field" by Web we arrive at what we know as *the Web as platform*. In the following sub-section we present a web service based messaging infrastructure, representing the Web 2.0 as platform.

5 Values Perceived and Future Challenges

By adopting the concept of Active Learning, we were able to address many of our learning objectives. In daily practice, learning-by-doing and learning-by-problem-solving successfully complemented declarative knowledge. Working in teams increased enthusiasm, communication and commitment that help us to develop a shared vision of our aims. Moreover, we found out that the application of this framework also strengthens mega-cognitive skills: the capacity of the students to reflect on what they know and what they don't know. After two years of experience, our shared assumption is that teaching Business Process Management can be supported very well by incorporating achievements of the Web 2.0. Its flexibility in a technological sense is also an advantage from an educational viewpoint. By working with several web service APIs as well as with open source tools in addition to our business integration server, many technical concepts related to e-Business integration could easily be demonstrated.

From an economical perspective, the big advantage of virtually available web service interfaces such as the Amazon Simple Queue service is that the principles of asynchronous messaging, a key to success in EAI and e-Business integration projects, can be demonstrated without establishing a local and therefore expensive messaging infrastructure at our university. As a matter of fact, this kind of technology enables our students to get involved with complex scenarios. Without the benefits of the Web 2.0, some of the students would never have got in touch with this kind of technology. Additionally, economic concepts such as Value Added Networks (VANs) can be explained my means of Amazon's web service infrastructure and its related business models.

When integrating applications on process level by means of the Internet, the security needed differs significantly from that of normal networks (Flieder, 2007b). Furthermore, for using the Web as platform – or as infrastructure – additional security is necessary for the exchange of data and information. For that reason our course concept for next year puts greater emphasis on WS Security and XML Security in the context of process integration and by means of web services. The usage of private and public LDAP (Lightweight Directory Access Protocol) directories for the application of authentication and authorization is another future challenge.

6 Conclusion

In this paper we outlined the advantages of public web service interfaces that provide interoperability and flexibility for interacting with the virtual world. With our examples we presented an overview of how to use the achievements of the Web 2.0 for teaching integration issues in an interdisciplinary context. By adopting the pedagogical framework Active

Learning, we were able to structure many of our learning objectives defined in an additional learning taxonomy. The work in small teams not only increased enthusiasm, communication and commitment among the students, it also helped a lot to agree on a shared vision about the students' future careers. We believe that they are on a good way to become Business-to-Business (B2B) specialists with a high core competency in protocols, formats and intra- as well as inter-enterprise integration concepts.

Knowing how to use Web 2.0 technologies is not the primary concern within this course. Our goal was rather to foster competence development on how to model a "world of services" and how to find one's role in a global economy. After finishing their studies, some of our students will integrate, automate, or optimise internal and external business processes within organisations, while others will be actively engaged in business development or even in BPM research. For all of them, the ability to handle the technological achievements of the Web 2.0 related to e-Business and enterprise application integration is a good preparation for their daily business in the future.

Acknowledgement

The research leading to these results has partially received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 231396 (ROLE project).

References

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

Ahn von, L., Blum, M., & Langford, J. (2004). Telling Humans and Computers Apart Automatically. *Communications of the ACM 47* (2), 57-60.

Barr, J., & Cabrera, L. F. (2006). AI Gets a Brain. *Queue 4* (4), 24-29.

Böszörmenyi, L. (2007a). On Methodic and Didactics of the History of Informatics. In L. Böszörmenyi (Ed.), *MEDICHI 2007 – Methodic and Didactic Challenges in Informatics* (pp. 29-33). Vienna: Oesterreichische Computer Gesellschaft.

Böszörmenyi, L. (2007b). *Wolfgang von Kempelen Prize for Computing History*. Retrieved August 9, 2010, from http://www.uni-klu.ac.at/tewi/inf/itec/dms/2814.htm

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of Education Objectives: The Classification of Educational Goals*. New York: David McKay Company.

Boulos, M. N. K., & Wheeler, S. (2007). The emerging Web 2.0 social software: an enabling suite of social technologies in health and health care education. *Health Information and Libraries Journal* (24), 2-23.

Chakrabarti, S., & Singbal, M. (2007). Password-based Authentication: Preventing Dictionary Attacks. *Computer 40* (6), 68-74.

Coughlan, T., & Johnson, P. (2006). Interaction in Creative Tasks: Ideation, Representation and Evaluation in Composition. *Proceedings of CHI 2006 Conference on Human Factors in Computing Systems*, 531-540.

Davydov, M. M. (2001). *Corporate Portals and e-Business Integration*. New York: McGraw-Hill.

Durand, T. (1998). The Alchemy of Competence. In G. Hamel, C. K. Prahalad, H. Thomas & D. O'Neal (Eds.), *Strategic Flexibility: Managing in a Turbulent Environment.* Chichester: John Wiley & Sons.

Erl, T. (2005). *Service-Oriented Architectures*. New York: Prentice Hall.

Flieder, K. (2007a). Kempelens Schachtürke: Seine Renaissance im Web 2.0. *Wissensmanagement – Das Magazin für Führungskräfte* (6), 10-12.

Flieder, K. (2007b). Sicherheit in der Prozessintegration mit Web Services und SOA. In P. Horster (Ed.) *DACH Security 2007* (pp. 83-97). Klagenfurt: Syssec.

Flieder, K. (2005). Testing and Visualizing a Message Queuing Infrastructure. *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI)*, 51-57.

Gillet, D., Law, E., & Chatterjee, A. (2010). Personal Learning Environments in a Global Higher Engineering Education Web 2.0 Realm. *Proceedings of the IEEE Engineering Education Conference (EDUCON).*

Hohpe, G., & Woolf, B. (2004). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Boston: Addison-Wesley.

Humbert, L., Micheuz, P., & Puhlmann, H. (2007). Why History matters in School Informatics. In L. Böszörmenyi (Ed.), *MEDICHI 2007 – Methodic and Didactic Challenges in Informatics* (pp. 161-172). Vienna: Oesterreichische Computer Gesellschaft.

Legner, C. (2007). Is there a Market for Web Services? – An Analysis of Web Service Directories. *Workshop Mashups 2007 at the 5th International Conference on Service Oriented Computing (ICSOC).*

Legner, C., & Vogel, T. (2007). Service-based Interoperability – Leveraging Web Services for Implementing Industry Standards. *Proceedings 20th Bled eConferenc, eMergence*.

Legner, C., Vogel, T., Löhe, J., & Mayerl, C. (2007). Transforming Inter-Organizational Business Processes into Service-Oriented Architectures: Method and Application in the Automotive Industry. *Proceedings of Fachtagung Kommunikation in Verteilten Systemen (KiVS), 37-48.*

Legner, C., & Wende, K. (2006). Towards an Excellence Framework for Business Interoperability. *Proceedings 19th Bled eConference eValues*.

Mann, D. (2002). *Hands-on systematic innovation*. Leper, Belgium: CREAX Press.

Mann, D., & Catháin, C.Ó. (2001). 40 Inventive (Architecture) Principles with Examples. *The TRIZ Journal* (7). Retrieved November 10, 2007, from www.triz-journal.com

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

O'Leary, C. (2003). Technology for automated assessment: The World-Wide-Mind. *Proceedings of the 4th Irish Educational Technology Users Conference (EdTech03).*

O'Reilly, T. (2005). What Is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software. Retrieved August 19, 2010, from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

Papazoglou, M. P., & Ribbers, P. M. A. (2006). *e-Business: Organizational and Technical Foundations*. Chichester: Wiley & Sons.

Rea, K. C. (1999). TRIZ and Software – 40 Principle Analogies, Part 1. *The TRIZ Journal*. Retrieved August 19, 2010, from http://www.triz-journal.com

Safran, C., Helic, D., & Gütl, C. (2007). E-Learning practices and the Web 2.0. *Proceedings of the International Conference on Interactive Computer Aided Learning (ICL).*

Searle, J. R. (1980). Mind, Brains and Programs. *The Behavioral and Brain Sciences 3* (3), 417-424.

Selzer, E. (2007). Why the Turing-Machine is not able to Play the Language of Game of Wittgenstein. In L. Böszörmenyi (Ed.), *MEDICHI 2007 – Methodic and Didactic Challenges in Informatics* (pp. 97-99). Vienna: Oesterreichische Computer Gesellschaft.

Shneiderman, B. (2002). *Leonardo's Laptop: Human Needs and the New Computing Technologies*. Cambridge, MA: MIT Press.

Turing, A. M.: (1950). Computing Machinery and Intelligence. *Mind 59* (236), 433-460.

W3C (2004). Web Service Architecture, W3C Working Group Note 11, February 2004. Retrieved August 19, 2010, from http://www.w3.org/TR/2004/NOTE-ws-arch-20040211

Wade, V. P., & Ashman, H. (2007). Evolving the Infrastructure for Technology-Enhanced Distance Learning. *IEEE Internet Computing 11* (3), 16-18.

Wilding-McBride, D. (2003). *Java on PDAs with J2EE*. Boston, MA: Addison-Wesley.

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

Author(s):

Karl Flieder, Dipl.-Ing.(FH)
Academia Nova, Wirtschaftsinformatik
Am Concorde Park 2, A-2320 Schwechat, Austria
eai@karlflieder.at

Felix Mödritscher, Dipl.-Ing. Dr.techn.
Vienna University of Economics and Business (WU)
Augasse 2-6, A-1090 Vienna, Austria
felix.moedritscher@wu.ac.at