DESIGN OF THE SEEWOP SOFTWARE PROTOTYPE

Thomas Kern[(a)], Tanja Peschetz[(b)], Gerald Petz[(c)]

[(a,b)] University of Applied Sciences Upper Austria, School of Informatics/Communications/Media, Softwarepark 11, A-4232 Hagenberg, AUSTRIA
[(c)] University of Applied Sciences Upper Austria, School of Management, Wehrgrabengasse 1-3, A-4400 Steyr, AUSTRIA

[(a)] thomas.kern@fh-hagenberg.at, [(b)] tanja.peschetz@students.fh-hagenberg.at, [(c)] gerald.petz@fh-steyr.at

ABSTRACT

E-mobility in the sense of virtual mobility – the performance of tasks via electronic networks by so-called e-workers – reduces the necessity to commute physically to a workplace. It represents a working method, from which both companies and employees expect benefit in terms of flexibility, productivity and cost. Existing micro-job platforms already offer an easy possibility to tender small work assignments via social-web in order to be processed by an e-worker community. However, dealing with complex task settings is difficult. In this paper we describe the prototype of the Semantically Enriched E-Working Platform (SEEWOP) which enhances the capabilities of an existing micro-job-platform by Event-driven Process Chains (EPC) and Semantic Web Technologies for the design of business processes and the automatic division of tasks within the e-worker community.

Keywords: e-mobility, e-work, crowdsourcing, semantic web, Event-driven Process Chains

1. INTRODUCTION

E-working platforms can be used as an efficient method for enterprises to support the management and execution of tasks over the Internet. They can be very useful in order to improve communication and increase working efficiency between companies (requesters) and their employees (as e-workers). The biggest challenge for requesters is to divide complex workflows into separate *Human Intelligence Tasks (HITs)* so that they can be distributed and processed through the crowdsourcing system. Especially workflows, which often have to be performed in a similar, but infrequently in the same way within a company – e.g., the creation of a monthly advertising catalogue for a company by commercial graphic designers – cannot yet be assigned via crowdsourcing, because of the fact that the effort to define, manage and control the HITs and the merging of the individual results would be too big (Kern and Petz 2012). Therefore, the *Semantically Enriched E-Working Platform (SEEWOP)* is in development, combining a classic Web 2.0 micro-job platform with Semantic Web Technologies to provide a novel method of intelligent e-work management.

The actual development step of the SEEWOP prototype presented in this paper provides the ability to model business processes as ontologies to be used as templates for generating HITs within a micro-job platform automatically. It simplifies the distribution and management of tasks as well as the management of e-workers.

2. STATE OF THE ART

2.1. Micro-Job Platforms

Similar to the job-marketplace *Amazon Mechanical Turk* described in (Kern and Petz 2012) the micro-job platform *Klickwork* (http://www.klickwork.com) allows its clients (requesters) to tender small jobs (HITs) and process them via a user-friendly web platform (see Figure 1). Requesters can determine the amount of work, set prices, and check the results to approve them. Additionally, each job is provided with a number of points for payment (corresponding to the effort) and a deadline.

Figure 1: Klickwork Website in German.

E-workers can select from a list of offers and apply for a job according to their interests and talents. After surcharge the job can be processed by the selected e-workers and finally, after completion, the results can be submitted via the platform. After approval by the requester payment is initiated.

Typical areas of application are digitally producible services like online marketing, viral marketing, internet information retrieval, copywriting, market and competitive studies, data analysis and other small jobs (CYBERhouse 2013). As part of our research collaboration with Cyberhouse Ltd. we use this platform as a test front end for SEEWOP.

2.2. Event-driven Process Chains

An *Event-driven Process Chain (EPC)* is a modelling language which can be used to describe business processes and workflows (ARIS 2013). This semi-formal graphical representation is commonly used in modelling frameworks – e.g., Architecture of Integrated Information Systems (ARIS) – or as a component in enterprise resource planning (ERP) systems – e.g., SAP R/3.

2.3. Web Ontology Language (OWL)

The *Web Ontology Language (OWL)* is specified by the *World Wide Web Consortium (W3C)* to create, publish and distribute ontologies based on a formal description language that even software is able to understand and process the meaning of concepts and their relationships among themselves (World Wide Web Consortium 2013d). It extends the *Resource Description Framework (RDF)* which enables the description of (true) statements about resources in the form of a directed graph and *RDF Schema (RDFS)* which provides a basic vocabulary for RDF.

3. REQUIREMENTS

Currently only small tasks can be distributed and processed easily via micro-job platforms. To provide a solution for complex tasks it is intended to enrich these platforms with EPCs.

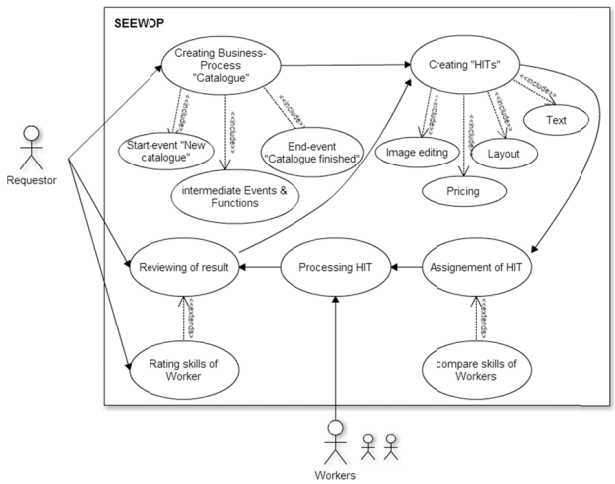

Figure 2: The use case "Creation of a Catalogue".

The main idea is to generate HITs within an e-working platform no longer by (the requester's) hand but automatically on demand based on a previously modelled common EPC-workflow description according to the respective actual parameters, process-states, events and results. To show the required functionality of the SEEWOP prototype a very simple use case named "Creation of a Catalogue" is shown in Figure 2.

1. *Business process creation:* The requestor once models a new business process which defines the common way of creating a catalogue according to the rules of the EPCs. This model is stored in the SEEWOP system and serves as template for future process instances.
2. *Business process instantiation:* To start a new catalogue project, the requestor selects a suitable process template and instantiates the process. After parameterizing, the process can be started.
3. *Business process consistency check:* According to entailment rules defined in RDF/OWL and ontology constraints SEEWOP checks the business process model for validity.
4. *Task generation:* SEEWOP begins to work off the business process by creating the first HITs based on the defined business process. For navigation through the control flow user-defined rules are evaluated by a reasoner.
5. *Task distribution:* The generated HITs are distributed to a selected micro-job platform automatically.
6. *Task processing:* The e-workers select and work up the HITs as usual within the micro-job platform and return the results. According to these results and the actual state within the business process new HITs are generated iteratively (step 3) until the end of the process.
7. *Enhanced functionality:* The use case also shows some possibilities of functional enhancements like the assignment of tasks to suitable skilled workers or rating the quality of working results.

3.1. The EPC Ontology

The fact that EPCs commonly just provide a semi-formal graphical representation of business processes results in a lack of processability by machines. Therefore as a first step a machine-interpretable representation – an ontology – for EPCs has to be developed.

The EPC Ontology explicitly specifies the concepts, their properties, and the necessary constraints to enable the modelling of valid EPCs in OWL. This formal semantic description and additional rules enable the SEEWOP-prototype to reason new facts by an inference mechanism at query time.

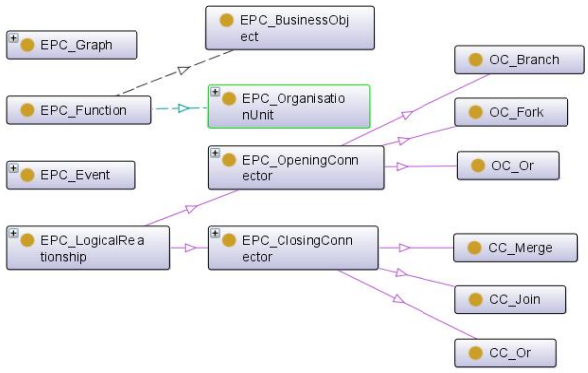

Figure 3: EPC Ontology Key Concepts.

The semantics of the key concepts shown in the simplified diagram in Figure 3 are:

- *EPC_Graph:* An EPC graph consists of finite sets of EPC elements. It starts and ends with an event.
- *EPC_Element:* Elements within an EPC graph are functions, events and logical connectors.
- *EPC_Event:* Events are passive elements in EPC. An event is a state occurring before or after a function.
- *EPC_Function:* Functions are active elements in EPC. A function is an action or task that follows an event.
- *EPC_OpeningConnector:* Opening connectors are used for logical relationships like "Branch", "Fork" and "OR". They may have one incoming and two or more outgoing control flows. A "Branch" is symbolized by an opening XOR connector. Depending on a condition exactly one of the outgoing control flows is activated. It is closed by a "Merge". A "Fork" is represented by an opening AND activating all outgoing control flows concurrently if the condition is fulfilled. It is closed by a "Join". An "OR" relation activates one or more outgoing control flows depending on the condition.
- *EPC_ClosingConnector:* Closing connectors are the corresponding counterparts of the logical relations described above and are also represented by the symbols XOR (for "Merge"), AND (for "JOIN") and OR. They may have two or more incoming and one outgoing control flow.
- *EPC_BusinessObject:* Business objects hold input or output data of functions.
- *EPC_OrganisationUnit:* Organisation units determine responsible persons or departments for a specific function.

4. SEEWOP SYSTEM DESIGN

The SEEWOP system design shown in Figure 4 consists of three layers based on the Semantic Web framework *Jena*.

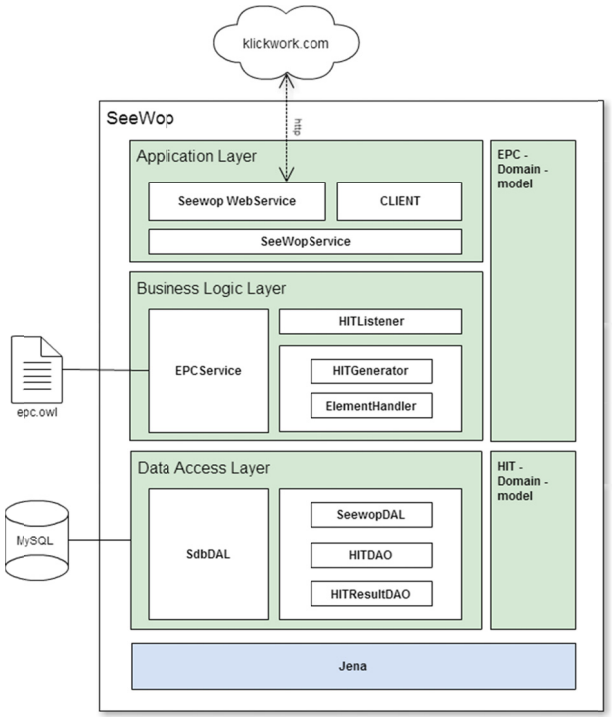

Figure 4: SEEWOP System Design.

4.1. Jena

Jena is a Java-based framework for the development with Semantic Web Technologies (Apache 2013a). It includes an API for reading, processing and writing RDF data, an API for handling OWL and RDF(S) ontologies (World Wide Web Consortium 2013d, 2013a), an inference engine for reasoning with RDF, OWL sources and Semantic Web Rule Language (SWRL) rules (World Wide Web Consortium 2013c), a data store for storing RDFS/OWL graphs, and a SPARQL query engine (World Wide Web Consortium 2013b).

4.2. Data Access Layer

The *Data Access Layer* enables the persistent storage of business processes and HITs and holds the underlying domain models.

- *EPC-Domain model:* This domain model describes all elements of the Event-driven Process Chains as domain objects.
- *HIT-Domain model:* The domain objects *HIT* and *HITResult* are representations of a Human Intelligence Task and their results with their corresponding properties. HITState currently defines the valid states of a HIT as "UNFINISHED" or "FINISHED".

The *SEEWOP Data Access Layer (SeewopDAL)* is responsible for storing and updating the domain objects HITs and HITResult (HITDAO and HITResultDAO). For realising the SeewopDAL the Java Persistence API (JPA) has been used. The JPA is a framework to bind relational databases in Java-based applications.

The generated business processes of the SEEWOP platform are represented as RDF graphs. For storing these graphs in the SQL Database the *Data Access Layer (SdbDAL)* uses the Triple Store *SQL Database (SDB)*. This Triple Store maps all triples of an RDF graph to a relational database and is provided by Jena for enabling the construction of persistent Jena models. An SDB store can be accessed and managed via command line scripts or the Jena API (Apache 2013b).

4.3. Business Logic Layer

The *Business Logic Layer* provides services for modelling and processing business processes. It is responsible for the generation of HITs at runtime.

The *Event-driven Process Chain Service (EPCService)* is responsible for all services which are dedicated to the modeling of business processes. These functions are:

- modeling of business processes based on the ontology Event-driven Process Chains,
- instantiation of business processes with concrete values (so called Individuals),
- adding SWRL Rules to concrete business processes,
- consistency check of produced business process ontologies, and
- storage of business processes.

The *HIT-Generator* creates HITs at runtime based on a given instantiated business process. Whenever an element of the type "Function" occurs in the process a HIT will be generated out of the information of the function. Visited elements of the type "Event" are added to the progress list of the workflow. For special logical connectors like a branch, a join, or a merge the *HIT-Generator* delegates to the *Element-Handler*. Figure 5 shows the generation of tasks out of an instantiated order process at runtime.

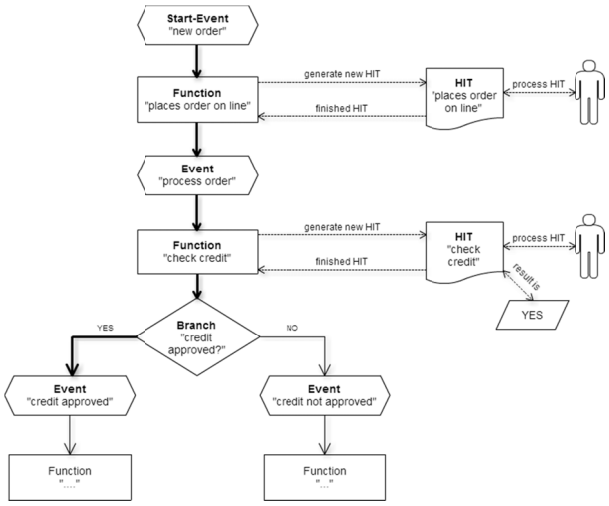

Figure 5: Functionality of the HIT-Generator.

All logical connectors are processed by the *Element-Handler*. If a SWRL-Rule is added to a connector, it will be evaluated and the inferred result will be stored in the knowledge base as new fact. According to these new facts the *Element-Handler* can return the next element to be processed by the *HIT-Generator*. As described in section 3.1 the connectors are categorised in opening and closing connectors for various logical relationships (shown in Figures 6a and 6b).

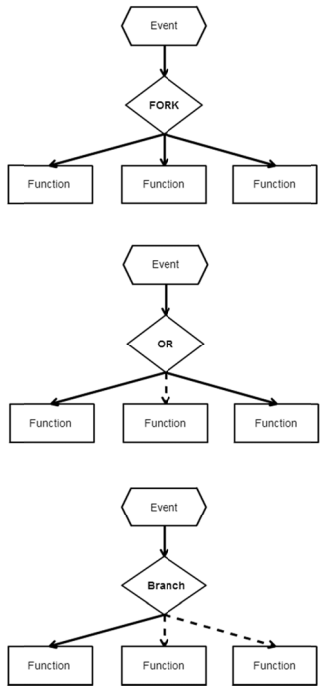

Figure 6a: Opening Connector types.

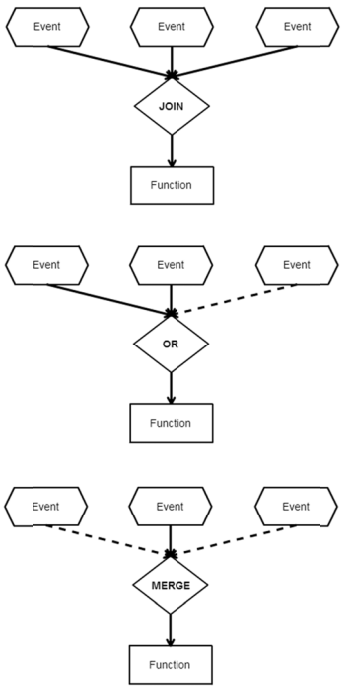

Figure 6b: Closing Connector types.

The *HIT-Listener* reacts to changes of HITs (see Figure 7). Depending on the modifications of the HIT the *HIT-Listener* delegates to other components which process the HITs. Currently the *HIT-Listener* only communicates with the *HIT-Generator* . If the state of a HIT changes to FINISHED, the Listener informs the *HIT-Generator* for processing the next elements of the business process chain and creates new HITs.

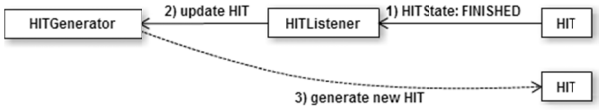

Figure 7: Functionality of the HIT-Listener.

4.4. Application Layer

The *Application Layer* includes all necessary functions for clients. The *Seewop Service* is the interface between the Business and Application Layer. It combines all necessary functionality for clients and web services out of the features from the *EPC-Controller* and *HIT-Listener*. These include above all the modelling, storing, instantiating, and consistency checking of business processes. Figure 8 shows a first implementation of the web-based SEEWOP user interface.

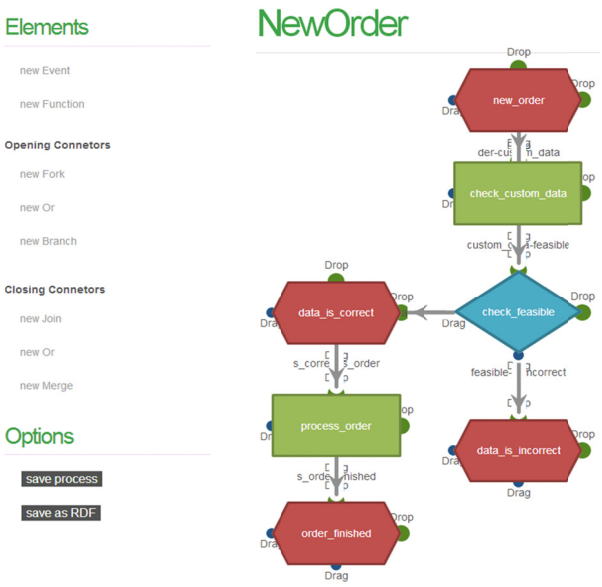

Figure 8: SEEWOP GUI for business processes.

The *SEEWOP-Web-Service* makes the SEEWOP features available for web-based clients such as the Micro-Job platform klickwork.com. It can get new HITs and stores the retrieved information. Afterwards, it is possible to process the HITs and inform the prototype SEEWOP about updates.

5. SUMMARY

The paper describes the design and architecture of the SEEWOP platform. It states the use of ontologies for describing business processes and individuals. By the integration of SWRL rules and inference it is possible to generate new facts. The prototype can use this knowledge to generate HITs at runtime and distribute it to workers. The aim of SEEWOP is to provide an intelligent task manager based on the knowledge of business processes. The prototype e-working platform simplifies the distribution and management of tasks, as well as the management of e-workers.

ACKNOWLEDGEMENTS

The described research activities are part of the research project 4EMOBILITY, carried out at the University of Applied Sciences Upper Austria (UAS), School of Management in Steyr together with the School of Informatics, Communications and Media in Hagenberg. The project is financed by the European Regional Development fund (EFRE/Regio 13) as well as by the Federal State of Upper Austria.

REFERENCES

Apache, 2013a. *Jena.* Available from: http://jena.apache.org/ [accessed 20 Feb 2013]

Apache, 2013b. *SDB - persistent triple stores using relational databases.* Available from: http://jena.apache.org/documentation/sdb/index.html [accessed 20 Feb 2013]

ARIS, 2013. *Event-driven process chain (EPC).* Available from: http://www.ariscommunity.com/event-driven-process-chain [accessed 25 Mar 2013]

CYBERhouse, 2013. *Klickwork.com.* Available from: http://www.klickwork.com/ [accessed 15 Apr 2013]

Kern, T., Petz, G., 2012. Basic Architecture of the „Semantically Enriched E-Working Platform“. *Proceedings of the International Conference on Harbor Maritime and Multimodal Logistics M&S, 2012,* pp. 153–156. September 19-21, Vienna (Austria)

World Wide Web Consortium, 2013a. *RDF Vocabulary Description Language 1.0: RDF Schema.* Available from: http://www.w3.org/TR/rdf-schema/ [accessed 20 Feb 2013]

World Wide Web Consortium, 2013b. *SPARQL Query Language for RDF.* Available from: http://www.w3.org/TR/rdf-sparql-query/ [accessed 15 May 2013]

World Wide Web Consortium, 2013c. *SWRL: A Semantic Web Rule Language.* Available from: http://www.w3.org/Submission/SWRL/ [accessed 20 Feb 2013]

World Wide Web Consortium, 2013d. *Web Ontology Language OWL / W3C Semantic Web Activity.* Available from: http://www.w3.org/2004/OWL/ [accessed 15 Apr 2013]

AUTHORS BIOGRAPHY

Thomas Kern is head of the Research Center Hagenberg, School of Informatics, Communications and Media, University of Applied Sciences Upper Austria. In this research facility approximately 40 R&D projects per year are carried out in the following three research areas Software Technologies and Applications, Information and Communication Systems, and Media and Knowledge Technologies. He finished his studies in Software Engineering in 1998. After industrial work experience he started as research associate and lecturer for the University of Applied Sciences Upper Austria in autumn 2000. Since 2003 he conducted several application oriented R&D projects in the fields of Bioinformatics and Software Engineering focused on design and implementation of knowledge-based systems. Since 2010 he is member of the 4EMOBILITY research project.

Tanja Peschetz is currently a student in the master degree programme "Software Engineering" at the University of Applied Sciences Upper Austria in Hagenberg. She finished her Bachelor degree in "Software Engineering" at the same University in 2011. Tanja Peschetz works for the R&D project SEEWOP and has implemented the first prototype.

Gerald Petz is director of the degree course "Marketing and Electronic Business" at the University of Applied Sciences in Upper Austria, School of Management in Steyr. His main research areas are Web 2.0, electronic business, electronic marketing and opinion mining; he has also conducted several R&D projects in these research fields. Before starting his academic career he was project manager and CEO of an internet company.