Collaboration Language for Social Information Engineering

Kurt Englmeier
Faculty of Computer Science,
P.O. Box 10 04 52, 98574
Schmalkalden, Germany
k.englmeier@fh-sm.de

Josiane Mothe
Institut de Recherche en Informatique de Toulouse,
UMR5505, CNRS, Université de Toulouse, IUFM
Toulouse, France
josiane.mothe@irit.fr

Fionn Murtagh
Dept. Computer Science, Royal Holloway,
University of London, Egham, Surrey TW20 0EX, UK
fmurtagh@acm.org

Javier Pereira
Engineering faculty, Universidad Diego Portales,
Av. Ejército 441, Santiago, Chile
javier.pereira@udp.cl

Duska Rosenberg
iCOM Centre for Research in Information, Computing and Communication, Royal Holloway Univ. of London
London, UK
duskarosenberg@me.com

***Abstract*— In this paper, we present a technology platform for personalised learning environments and discuss the key features that will enhance user experience. In particular, we consider elements of social software, integrated tools and user modelled services as they converge on the development of folksonomies. We base our discussion on design principles for accessing and sharing a range of different resources, tools and services for both individual and group learning. In the paper, we concentrate on the language model that supports the development of process-related folksonomies. The model emerges from both the functionality of the learning environment and the language applied in the specific learning environment, thus providing a common platform that users can themselves adapt to their specific learning needs and circumstances.**

Keywords- model of query language; social participation.

I. INTRODUCTION

Folksonomies or collaborative tagging became popular on the Web through social bookmarking and photograph tagging. This facet of social computing demonstrates that social activities on the web can form information landscapes. We think that folksonomies add a new quality to social computing: social information engineering. Our project Co-lingua reflects work-in-progress that fosters social tagging not only for particular information items but also for the information integration processes applied to generate information.

Information integration applications are prime candidates among systems that encourage active user engineering. Their requirements, wants, needs, ideas, and skills are so manifold and diverse that no traditional software engineering approach can capture them all in a satisfactory way. We therefore have to harness the power of user communities in order to share information.

With Co-lingua, we take up the rationale of technology-mediated social participation (TMSP) [3] and develop an open interaction language. "Open" means that users can adopt it as their commodity toolset, and also that its facets accommodate their domain-specific jargon. A recent study, conducted by Forrester Consulting [8], found that knowledge workers will significantly *benefit from a blended solution* that would allow them to fulfil most of their information requirements, while requiring minimal support from Information Technology (IT). While Forrester does not explain how such a blended solution may work, we are at least told that some minimal support from IT is one key ingredient of TMSP in the workplace. The users take the role of "prosumers", i.e., users frequently changing between the role of information consumers and information producers. Prosumers produce goods they and/or others consume.

Prosumer computing represents a blended engineering concept that emphasises collaboration between IT and users, where IT provides minimal support to users that develop self-service solutions autonomously. Co-lingua translates this rationale of prosumer computing into a new paradigm for information access and mining software: under-designed software fine-tuned by information access and mining prosumers using a controlled language interaction layer.

Within this rationale, the objective of Co-lingua is to transform natural language statements into machine-processable instructions [8]. These instructions control the *underdesigned software that provides generic features for search, access to structured and unstructured data, data cleansing, and data merging*. On top of that resides the controlled interaction language layer that lets users fine-tune the generic features using human language in all sorts of learning environments that direct user to develop suitable answers for questions like [11]:

- "What are the actual sales figures for the automobile industry in Greece?"
- "How can I build an indicator for the economic performance of the European industry?"
- "What kind of cancellation policy applies for booking BY2TF6?"

- "How can I join credit card information and customer demographics?"

In the second section, we present the rationale of our language model. Section 3 explains the semantics of the interaction model. Section 4 outlines how named entities and expressions of communicative acts are integrated through a domain-specific ontology. Section 5 concludes the paper.

II. RATIONALE AND RELATED WORK

A. Scenario

We take, for instance, an economic indicator system that helps people to learn more about indicators and to set up their own indicator system. Let us assume the user wants to compare the actual economic situation of the industry in the Maghreb states with that in the states of Southern Europe. The indicator is the product of a number of processing steps. From each country of the regions addressed three time series need to be retrieved: "business climate", "incoming orders", and "six months business expectations". According to the economists' advice, the user shall create composite time series for both regions. In the composite, each country has a different weight, of course. After developing moving averages (over 3 periods) from each time series, the user can compare the two data sets.

B. Rationale

Our concept of folksonomies covers three practical aspects that merge into a unified, domain-specific interaction language:

- The **service aspect** comprises first generic data access and processing services, the target of the learning environment. The huge diversity of economic indicators makes is impossible to fix all conceivable user requirements for the access and processing services. These services have to be generic, by nature, in order to accommodate the broad variety of their later use. The first part of the folksonomy results from "translation" of service signatures into expressions of the user community. This part can be a simplified command language similar to macro-languages. Moving averages over three periods of time series may result from an instruction like "moving_averages.time_series.3".
- Generic services need to be fine-tuned before they can be used. This can be achieved by **instructional folksonomies**. These come closer to the language action perspective explained below. They instruct, for instance, that "in any aggregate of Maghreb industrial data, Moroccan data enter with a weight of 1.8". The language applied to cover instruction aspects can also be a simplified command language like "economic situation = business climate + incoming orders".
- The **action language aspect** addresses primarily the users. Their actions reflect their navigation in the learning environment and the use of the tools provided. Their statements, thus, reflect directives, rejections, requests, and the like: "for aggregation apply 6-period moving averages", "send me the latest figures on business climate in Turkey when available", etc.

C. Related work

User statements, emerging from the instructional or language action aspect, usually contain precise and clear descriptions of resources and activities required to complete tasks. Merging language elements from the three aspects mentioned above lead to a controlled vocabulary [7] that semantically covers all user interaction possible in this particular environment [2].

Communicative acts performed during interaction reflect the users' domain expertise and their intentions. Our language model follows the rationale of *Language Action Perspective (LAP)* as developed in [4], for instance. LAP has roots in *speech act theory* [12] and the *conversation-for-action (cfa) schema* introduced by Winograd and Flores [14] (see also [1]). There were a number of approaches emerging from this schema, the *Action Workflow* [14], for instance, or the *BAT (Business interAction and Transaction) model* [4].

The rationale "is to get a model of how people, through conversation, coordinate their work" [13]. The language used in the communicative acts represents the language for the definition and coordination of processes. A storybook serves as a means for editing, categorising, and linking of the language elements. Communicative acts are developed around the storybook metaphor and the principles of narration. It forms the blueprint for Co-lingua's interaction model.

III. SEMANTICS OF THE INTERACTION MODEL

Folksonomies in Co-lingua are constrained by semantic representations of the functionality of the respective learning environment. The controlled language thus covers all use patterns related to this functionality. Conversely, all variants of interactions and objects that can be associated with the functionality form the semantic space the controlled vocabulary has to cover. In principle, an ontology for the specific learning environment could constitute the centrepiece of this interaction language. The ontology needs to be grounded sufficiently in the signature of the services. It includes the roles of the actors involved, the objects addressed, and the objects' collaboration model. In combination with the ontology, the interaction model defines and controls not only user dialogues but also service choreography and thus aligns work flow and flow of information.

The challenge is the open definition of the interaction language that maps to information integration functionality. Definitions must be "open" in terms of language elements being individually definable and adaptable by the users. The learning environment, in turn, defines the functionality. The target language is the language users expect when interacting in natural language with a learning environment. The development of the target language follows first a bottom-up approach. The starting point is the source language emerging from generic services. Through gradual enhancement, the source language develops into the simplified command language.

The *service interfaces* provide an initial set of elements for the source language. Each service hosts a

number of actions referring to a particular theme of task. A *search* action, for instance, looks for one or more *items* in one or more *containers*, while each item to be included may have one or more characteristic *attributes*. This stereotypical search pattern appears in SQL statements as well as text search. The source language is thus a unified command language the services can interpret. In a very simplified version such a command language can take the form

search.archive_of_persons.person.age > 24,
country.residence=greece,

for looking for persons older than 24 years and living in Greece. By adapting the nomenclature that gains popularity in programming languages like Ruby we say that the plural of a noun automatically indicates the archive containing items named by that noun. For example, the archive "persons" contains items of type "person". This reduces the command to search.persons.age > 24, country.residence=Greece.

We emphasise that this language is certainly an oversimplified version for a stereotypical action pattern that is probably better represented in WSDL (Web Service Definition Language). However, it suffices to explain the first level of the source language.

At a first glance, this command language is already a standard language IT people should agree upon. However, in many cases such a standard exists in IT. In particular when IT departments apply semantic web standards, they enforce standardised interface and object representations. The collaboration already in place in many IT departments and the enforcement of common standards supports the take-up of the collaborative management of a standard command language.

Our *simplified command language* is first a proposal for an intuitive language on the interface level of services. It is intuitive, because it follows commonly known syntax patterns of programming languages. Although quite simple, this command language is powerful and easy to implement. A parser uses a series of regular expressions to interpret the command. The assignments of command terms to action patterns can be explicit or implicit through nomenclatures. Users define assignments by explicit statements like container=archive_of_persons, persons, employees or country. residence="RESIDENCE" (to adapt different column names to the command languages).

Co-lingua's determination of language concepts uses grammar-free text analysis. It applies methods for automatic text analysis such as stopword elimination, identification of composed terms, and named entity recognition. Recursive pattern analysis develops a hierarchical structure of language concepts:

- It starts with determining basic elements like word, numerical value, date, address, value range, booking code, zip-code, email address, etc. through Regular Expression analysis.
- Next, we repeat pattern analysis on the term level to determine compounds of basic elements reflecting minimal semantics like composed terms including combinations with numerical values ("older than 24", "weight: .35", etc.). Composed term determination precedes this analysis step.
- Minimal semantic expressions are input to further pattern analysis. Like in determining composed terms we repeat the determination of terms frequently appearing in close proximity. We expect to reveal superior concepts like "birth" identified in patterns like "[name], born in [name] on [date]". This determination process will see human intervention to assign concept names to term patterns.
- The next step is the categorisation of the concepts identified in the previous step. Categorisation here means both structuring concepts along generalisation-specialisation as well as hierarchical annotation of text documents according to the hierarchical organisation of the text (along titles, subtitles, paragraphs, etc.).

The end result is a thesaurus developed along categorised named entities with ramifications into text documents and into command language expressions. It is important to note, that the first candidates for named entities are the parameters of the service together with their names (i.e., the terms applied to express their signatures.). This thesaurus represents the named entities of the interaction language.

IV. METALANGUAGE LAYER FOR SOCIAL INFORMATION ENGINEERING

It is certainly reasonable to model social information engineering along an ontology that processes (task ontology) and actors and objects (onto terminology). The conceptual model is the blueprint for domain-specific meta-models. The Co-lingua metalanguage should preferably be relatively simple and precisely specified so as to be amenable to processing by the generic information integration services. In principle, this is the approach taken in the semantic web, where ontologies are used to provide extensible vocabularies of terms.

The interaction language shall be powerful enough to enable users, by following their intuition, to define classes, properties and relationships while the system checks for implicit subsumption relationships and gives feedback about the logical implications of their design, including warnings about inconsistencies and synonyms. Ontologies allow the representation of semantic memory, but a narrative is a way in which a story is told. Finally, the dialogue system managing the storybooks will be able to explain inferences; without explanations, users may find it difficult to repair errors in the ontology and may even start to doubt the correctness of inferences. Argument schemas may contribute to provide explanations by analysis of arguments for and against some claim (e.g., a proposition, an action intention, a preference, etc.). An argument schema consists of a claim, a set of premises, i.e., a number of supportive and possible denying reasons, and an aggregation mechanism to reach a global conclusion concerning the validity of that claim [10].

By applying the social computing paradigm it is possible for user communities to collaboratively create simple forms of ontology via the development of action

signatures and object/actor tags organised in folksonomies [13]. The ontology plays the following roles: provides a formally defined extensible vocabulary for semantic annotations; describes the structure of domain sources and the information they store; and provide detailed model of the domain against which queries are formulated. The semantics of the vocabulary is mainly captured informally in textual and contextual descriptions of each term and procedurally interpreted by information integration services. This informality reduces the complexity of reasoning without limiting the ability of the services to "understand" vocabularies. It enables, in any case, logic for which query answering can be implemented using rule-based techniques.

OWL, the Web Ontology Language, has been defined to be used for web resource indexing. Thanks to OWL, it is possible to represent the lexicon in the form of which a concept will occur in a document or the user's interaction. To model the terms used for this concept, OWL associates with the associated class one or several sequences of characters by the means of RDF-S.

IRIT laboratory developed a model that aims at considering context, content and communicative acts in information search and navigation related to learning. We propose to model two main aspects of context: the themes of the user's problem and the specific information the user is looking for, to achieve a particular learning task. Both aspects are modelled by means of ontologies. Documents are semantically indexed according to the context representation and the user accesses information by browsing the ontologies. OntoExplo is an interface we developed based on this model. It has been applied to a case study in the domain of astronomy that has shown the added value of such a semantic representation of context [5] as well as in the domains of e-learning [6] and security.

Figure 1 presents the OntoExplo interface. The documents that are handled are scientific papers in the domain of public health. The task ontology is presented on the left hand side of the figure. It corresponds to meta-data associated with the documents (authors, title, language, etc.). This description follows the Dublin Core. On the right hand side of the figure, the security domain-ontology is presented. It can be browsed to see the various concepts. In addition, the user can query the ontology to access a specific concept.

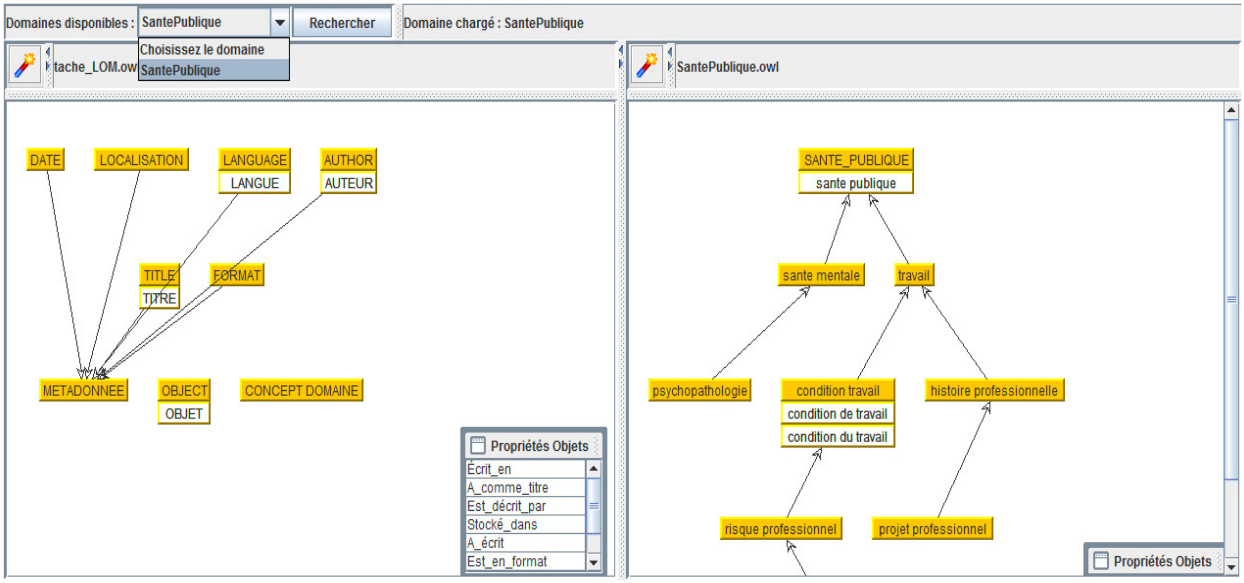

Figure 1. OntoExplo interface, task and domain ontologies.

One of the documents is more specifically visualised on the left hand side of Figure 2. Its description is visualised. From this, it is then possible to interactively visualise the concepts from the domain ontology that occur in this specific document. Task and domain ontologies can be browsed the other way around: the user can visualise the documents that are indexed by a specific domain concept that is chosen. Alternatively, she can select an author name and dynamically visualise the set of documents this author wrote.

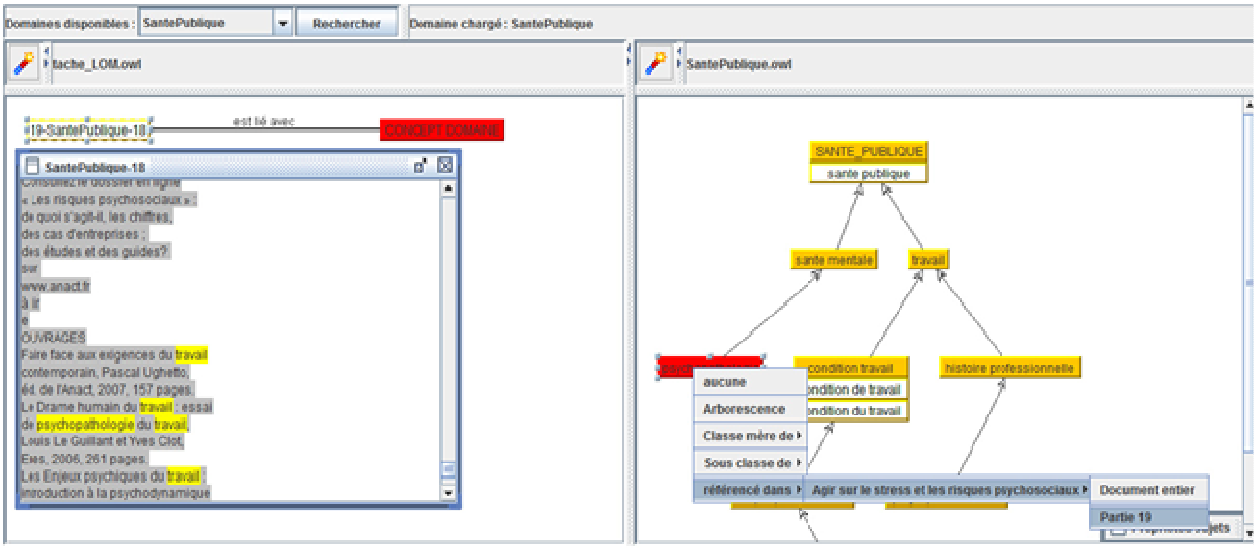

Figure 2. OntoExplo interface, browsing task and domain ontologies.

V. Conclusion and Future Work

As stated above, this paper presents work in progress, representing the development of technology-mediated social participation approach suitable for user-centred design of personalised learning environments. Both education and industry have recognised the benefits of platforms supporting social activity in order to enrich our everyday work and to make our work processes more efficient. This demonstrates the need for stronger focus on social computing in the design of collaborative environments for learning and work. However, the successful development of new applications along the social computing paradigm also demonstrates that the community of "engineers" is not limited by IT skills alone. Users are excellent "engineers" when it comes to organising their information processes. The first step is to endow them with an information engineering environment that enables them to express through folksonomies how information integration should behave in their individual domains. Our future work will illustrate a way in which folksonomies guide the entire process of developing a technology platform that supports both individual and social learning.

VI. References

[1] Brown, P.C., Succeeding with SOA: Realizing Business Value Through Total Architecture, Addison Wesley Professional, Boston, MA, 2007.

[2] Dietz, J.L.G., The Deep Structure of Business Processes, Communications of the ACM, vol. 49, n. 5, 2006, pp. 59-64.

[3] Fruchter, R, Nishida, T., and Rosenberg, D., Understanding mediated communication: the social Intelligence Design approach, AI & Society-Social Intelligence Design for Mediated Communication, vol. 19, n. 1, 2005, pp. 1-7.

[4] Goldkuhl, G., Action and Media in Interorganizational Interaction. Communications of the ACM, vol. 49, n. 5, 2006, pp. 53-57

[5] Hernandez, N., Mothe, J., Chrisment, C., and Egret, D., Modeling context through domain ontologies, Journal of Information Retrieval, vol. 10, n. 2, 2007, pp. 143-172.

[6] Hernandez, N., Mothe, J., Ralalason, B., Ramamonjisoa, A., and Stolf, P., A Model to Represent the Facets of Learning Objects, Interdisciplinary Journal of E-Learning and Learning Objects, vol. 4, 2008, pp. 65-82.

[7] Huijsen, W.-O., Controlled Language—An Introduction, Proc. 2nd Int'l Workshop Controlled Language Applications, Carnegie Mellon Univ., 1998, pp. 1-15

[8] Englmeier, K., Pereira, J. Mothe, J., Choregraphy of web services based on natural language storybooks, international conference on Electronic commerce, 2006, pp. 132-138.

[9] Lean Business Intelligence, Forrester Consulting, October 2, 2009. http://www.slideshare.net/findwhitepapers/lean-business-intelligence-how-and-why-organizations-are-moving-to-selfservice-bi, 31 march 2011.

[10] Ouerdane, W., Maudet, N., and Tsoukias, T., Argument Schemes and Critical Questions for Decision Aiding Process. In Proceedings of the 2nd International Conference on Computational Models of Argument, 2008, pp. 285-296, IOS Press.Toulouse - France,.

[11] Ravat, F., Teste, O., Tournier, R., and Zurfluh, G., Graphical, Querying Multidimensional Databases. 11th East-European Conference on Advances in Databases and Information Systems, Springer-Verlag, LNCS 4690, 2007, pp. 298-313, Varna (Bulgarie).

[12] Searle, J.R., Speech Acts – An Essay in the Philosophy of Language, Cambridge University Press, Cambridge, MA, 1969.

[13] Spyns, P., de Moor, A., Vandenbussche, J., and Meersman, R., From folksologies to ontologies: How the twain meet. In Proceedings of On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, Lecture Notes in Computer Science 4275, Springer, 2006, pp. 738–755.

[14] Winograd, T. and Flores, F., Understanding Computers and Cognition: A New Foundation for Design. Ablex, Norwood, 1986.