Openplexus: Distributed Knowledge-Sharing in Virtual Teams

Robert Giacinto¹, Ana M. Lara Palma²

Cologne University of Applied Sciences¹ Steinmüllerallee 1, 51643 Gummersbach, Germany University of Burgos² Hospital del Rey, 09001 Burgos, Spain

Abstract. In this paper we describe a new distributed knowledge sharing framework that supports knowledge discovery and context-aware knowledge sharing in virtual teams by introducing a 3-tier knowledge sharing architecture. The three layers reflect different facets of knowledge: the personal knowledge that might only have a relevant meaning for its author, the shared vocabulary and experiencebased knowledge of teams and the persistent knowledge of the organisational memory. The prototype was designed by following a user-centred design approach in which the involved users gave important insight for the definition of relevant use cases and scenarios that describe common tasks in virtual teams.

Key words: Knowledge Sharing, Semantic Web, Distributed Systems, Virtual Teams

1 Introduction

One main concern with knowledge work in virtual teams is the externalisation and reusability of knowledge that is created in projects. Virtual teams can be described as "groups of geographically, organizationally and/or time dispersed workers brought together by information and telecommunication technologies to accomplish one or more organizational task" [5, p. 7]. Many artefacts are often buried in folders or email attachments of a single computer and are therefore not available to others. Enterprise Information Systems that focus on archiving documents in a central repository solve the availability issue but do not support easy discovery and reuse in virtual team settings since finding the right document or at least a good starting point for a search activity can be time-consuming and is often context-related rather than task-related. Our tool tries to bridge the gap between existing search-oriented information systems and the need for context-aware recommendations and the creation of informal knowledge spaces that can be used to make the tacit and distributed knowledge of a virtual team explicit and accessible to the organisation.

The remainder of this paper is organised as follows: In section 2 we describe our approach to a context-aware system called Openplexus. Section 3 gives an overview of the current architecture of the system and we conclude this paper with a summary of the current state and future work in section 4.

2 Approach

With the growing amount of available data the incorporation of contextual information into the selection and filtering process can help cut down the number of candidates in search results and therefore reduces the effort for the individual ([1], [3]) to find the right document for a task. The use of context models can be found in several implementations of semantic information systems (e.g. [4], [2]) and mainly differs in the number of context dimensions that are taken into account. These systems focus on local events that affect the context of an individual. In virtual teams, the context that influences the importance or meaning of a document can change even in the absence of a user.

Many definitions of context have been proposed in the literature over the years coming from different research areas. For our system, we use the definition by Shkundina et al. [6] where a context is comprised of six dimensions: (1) information, (2) organisation (organisational structures or persons), (3) behaviour (performed actions), (4) operation (used tools or applications), (5) cause (task and user goals) and (6) chronology (timeline of events).

To model different layers of available knowledge in virtual teams, a 3-tier knowledge architecture is implemented in Openplexus:

- 1. *Personal layer* that holds the individual knowledge of the owner of this instance. The knowledge can only be accessed by the owner.
- 2. *Team or Knowledge Sharing layer* that holds the knowledge that belongs to a team or that was shared with other individuals.
- 3. *Organisational layer* that holds the knowledge that should be accessible to all users in the organisational network.

To enable the user define a personal mental model and support the sharing of it, a shared base vocabulary is necessary from which the personal knowledge space is derived. These shared concepts are described in the Openplexus Upper Level Ontology and come from interview sessions with stakeholders and contextual enquiries. Where possible, existing ontologies (e.g. Prov-O¹, DUL², FOAF³ and Vocab-Org⁴) were used to derive sub-properties and sub-resources in the Openplexus namespace to create a clean and understandale schema for the user while relying on the expressiveness of existing vocabularies.

3 Openplexus Architecture

Fig. 1 depicts the architecture of the Openplexus system. It is designed as an event-based middleware system built from a set of loosely-coupled and extendable components. The system gathers and aggregates local and distributed event information created by different processes to model the current context. In its current implementation the system

¹ http://www.w3.org/TR/2013/REC-prov-o-20130430/

² http://www.ontologydesignpatterns.org/ont/dul/DUL.owl

³ http://xmlns.com/foaf/0.1/

⁴ http://www.w3.org/TR/vocab-org/

16 Giacinto & Lara Palma



Fig. 1. Conceptual overview of essential components implemented in an Openplexus node. Dashed arrows symbolise the use of the event bus, solid arrows are direct communications between two components.

focuses on the dynamic nature of collaborations in teams which are supported by the possibility to create adhoc peer-to-peer networks with nodes of other team members in which shared activities and events are distributed and kept synchronised.

Local Event Providers Local Event Providers are used to push context-related events from user interactions with the operating system and external applications to Openplexus. Here, two different providers are implemented: (1) Plugins for third-party applications like email clients or web browsers and (2) native event listeners to capture interactions with the operating system or watched folders. Additional events are fired by tools provided by Openplexus that help create new team spaces, let the user join active team spaces or share documents with team members. Each interaction with the system that influences a user's context is associated with an event that is fired and processed by the Openplexus system.

Local Content and Information Extraction This component listens for filesystem and files-related events and implements content extractors for the most common filetypes. The extraction can happen automatically if the related file resides in a watched folder or manually if the user passes a file to Openplexus (adding it to the personal knowledge space or sharing it with a team).

Organisational Services This is a set of components that can provide access to services that are only available to members of an organisation (e.g. access to the organisation-wide knowledge-base which can be comprised of semantic services or other (legacy) databases where queries are handled by an integration layer, or internal authentication services). The current implementation holds the ontology that acts as the organisational knowledge layer and directories of currently active team activities that a user can join.

(Local & Shared) Context Processing For every team a separate process handles the distributed events that come from the network of connected peers. The remote context of the team and the local context of the user are combined and enriched with information from past contexts that are similar to the current one. All context-related events are used as input for a multi-layer perceptron that is trained and used to classify the type of the current context where the identified type defines the way how the recommendations are calculated.

(Local & Shared) Knowledge Space A local knowledge space belongs to exactly one user and is only accessible from the local Openplexus node. Here, the user is able to introduce new concepts to the local ontology. In its current state, subclassing of existing concepts is supported. The new concept is then added to the ontology of the knowledge space if it is the private knowledge space of the user. In a distributed knowledge space, this change is proposed to all members of this shared space and is only applied if the majority of this team accepts the change.

Distributed Teams Handling Interactions between team members are related to a *Team Activity* which is the temporal event when members of a team collaborate. These team activities can happen in adhoc peer-to-peer (P2P) networks that are used for communication and knowledge sharing tasks. Openplexus handles a shared knowledge space for each team. It holds the shared knowledge and context information that is created during collaboration activities and maps to the second tier of our knowledge hierarchy.

Context-Aware Dashboard The dashboard is the UI of the system and serves as a single point of entry to all services of the platform. The current focus for the implementation of this component is the context-based visualisation of the available data by adjusting the displayed content and resources depending on the current context of the system. Fig. 2 depicts a mockup which emphasises the proactive aggregation of relevant information. In this example, Openplexus identifies a phone call context and deligates the rendering of the UI to a context-specific visualisation handler.

4 Conclusion and Future Work

In this paper, we presented an architecture of a context-aware information system that aggregates and processes information from different knowledge tiers of an organisation, offering a fuller view on the available knowledge. The next steps include formalising and extending the context similarity measure in use and the evaluation of the quality of the recommendations by evaluating precision and recall in a given scenario and the usability of the system. Additionally, the handling of distributed changes and the weighting of the tiers requires further investigation.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

18 Giacinto & Lara Palma



Fig. 2. Mockup of the context-aware dashboard, presenting information for an ongoing phone call.

- Alper Aksac, Orkun Ozturk, and Erdogan Dogdu. A novel semantic web browser for user centric information retrieval: Person. *Expert Systems with Applications*, 39(15):12001–12013, 2012.
- Susan Dumais, Edward Cutrell, and Hao Chen. Optimizing search by showing results in context. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 277–284. ACM, 2001.
- Tudor Groza, Siegfried Handschuh, Knud Moeller, Gunnar Grimnes, Leo Sauermann, Enrico Minack, Cedric Mesnage, Mehdi Jazayeri, Gerald Reif, and Rosa Gudjónsdóttir. The nepomuk project - on the way to the social semantic desktop. In *Proceedings of I-Semantics'* 07, pages 201–211. JUCS, September 2007.
- 5. Anne Powell, Gabriele Piccoli, and Blake Ives. Virtual teams: a review of current literature and directions for future research. *ACM Sigmis Database*, 35(1):6–36, 2004.
- 6. Roza Shkundina and Sven Schwarz. A similarity measure for task contexts. In *Proceedings* of the Workshop Similarities-Processes-Workflows in conjunction with the 6th International Conference on Case-Based Reasoning, 2005.