Requirements of Affective Information Systems in the Industrial Domain

Joachim Baumeister[1,2]

[1] denkbares GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg
[2] University of Würzburg, Am Hubland, 97074 Würzburg

Abstract. Recently, semantic technologies improved the information access to massive amounts of data in various manners. Also industrial domains benefit from semantic approaches. Information now can be accessed more precisely and is directly linked with related data. However, the actual support for humans to access suitable content at time is still in its infancy. In this paper, we motivate a number of requirements for extending current state-of-the-art information systems in order to support human-aware information access. The requirements are based on the experiences we made during the introduction of industrial information systems over the last years.

1 Introduction

We motivate the application of semantic technologies and affective computing in the domain of Technical Service. The primary task of *Technical Service* is the troubleshooting and maintenance of advanced machinery. For instance, Technical Service has to identify the reason for a malfunction of a machine and in consequence has to replace and adjust faulty components. With the increasing complexity of machinery the Technical Service becomes more and more challenging even for well-trained service technicians. *Technical documentation* provides the service-related information to support technicians during their work. Having complex machinery, however, the information typically covers some thousand pages only for a single machine. Consequently finding relevant information units in a problem situation is difficult, time-consuming, and often not successful. Many machinery companies report, that their globally distributed service technicians are often not able to find helpful information during their operations.

Recently, semantic technologies advanced information systems to improve the search and navigation in the information data bases [5]. In contrast to full-text search systems, semantic metadata are attached to information resources to precisely point to information. Then, for instance, a chapter within a technical documentation explicitly states its function as a *repair instruction* and names the concrete *components* that are assembled [6]. With these metadata, the search and navigation is improved dramatically. One prominent technology are semantic search systems [7], where the retrieval of information is based on semantic queries. With semantic search, a query can be stated much more precisely and search results can be reduced significantly. The introduction of semantic technologies improved the application of service information systems. However, a number of challenges are still left unsolved.

In this position paper, we discuss the requirements of the domain *Technical Service* in Section 2. We propose a conceptual model approaching these requirements in Section 3. We conclude the paper in Section 4.

2 Requirements for Affective Information Systems in Technical Service

Semantic technologies improve the accessibility of resources in information systems dramatically. In comparison the standard information systems with text-based queries and tags, semantic information systems are able to precisely point to resources based on concepts stated in the query. Further, they are able to rewrite queries so that results include semantically related or synonym resources.

Nevertheless, a number of requirements exist in Technical Service, which should be taken into account to simplify and refine the information access for the user. We claim, that current systems miss different aspects of *awareness* that in summary complicate the access to information. An overview of the aspects is depicted in Figure 1.

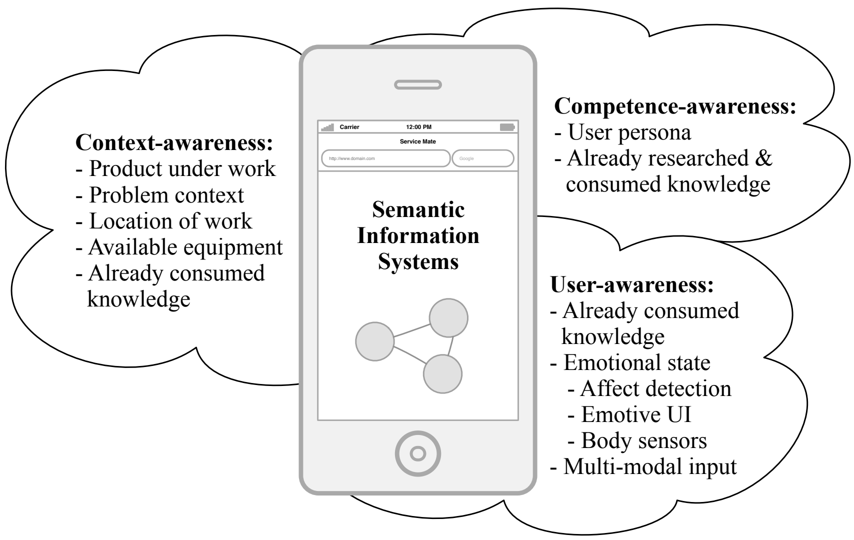

Fig. 1. Requirements of affective semantic information systems together with supporting technologies.

We discuss the aspects in the following:

1. **Context-awareness:** For large corpora, even semantic search yields large amounts of information resources and the selection of the most appropriate resource rarely can be done automatically. With the knowledge of the current user context, a *context-aware system* can filter the query results to the most appropriate ones. For instance, when looking for the replacement of a component, it is not necessary to also offer the operation manual for this component.
2. **User-awareness:** Stating a semantic query is not necessarily more user-friendly than a standard text query. Users have frequently problems formulating appropriate queries. With growing frustration of not finding an appropriate information resource, the users are again turning away from the documentation system missing important information for their work. A *user-aware system* will monitor the user's satisfaction during the information research and consumption. It will propose alternative ways of communication when the user's satisfaction is downgrading.
3. **Competence-awareness:** Technical information is typically written for well-trained technicians. In the context of world-wide service operations the technicians may not be well-trained in some regions. Then, the information resources are simply not useful because they miss important detail for less competent technicians. A *competence-aware system* will preferably present information with an appropriate level of detail to the user.

In the following section, we introduce a conceptual model for the realization of the requirements stated above.

3 Conceptual Model

We develop a conceptual model relating information resources from a semantic information model with concepts representing context-awareness, user-awareness, and competence-awareness. The semantic information model is fitted to the needs of information management in the domain of Technical Service and has been successfully used in a number of industrial projects.

The shown model comprises task-specific ontologies and it will be used as the blueprint for implementing smart service systems in the future. For the implementation and visualization of the ontologies we used the open-source version of the semantic wiki KnowWE [2].

Figure 2 shows the general concept of information resources, i.e., entities. We use the class `prov:Entity` from PROV-O [9, 11] to represent general information units. When needed, PROV-O allows for a standardized representation of provenance of entities, such as the creation date, author and version history. Each instance of `prov:Entity` is assigned to a mime type to identify the kind of information. Elements of metadata are represented as instances of `skos:Concept`. SKOS [10] is a standard ontology for representing knowledge structures, such as concept hierarchies. The properties `skos:broader` and the

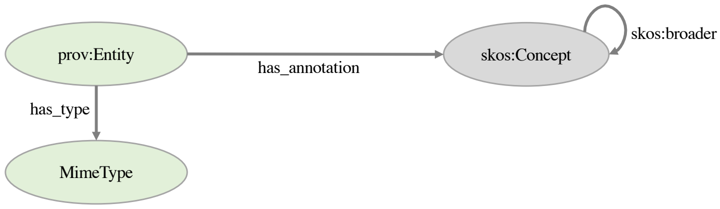

Fig. 2. Basic elements of an information systems: Information resources are assigned to concepts describing the metadata.

inverse `skos:narrower` are used with corresponding sub-properties to build a flexible hierarchy structure.

3.1 Information Representation in Technical Service

The basic ontology from above is extended to model the general artifacts of the domain Technical Service. An actual application–the entities of a concrete manufacture–will align to this upper ontology.

In Figure 3 we see the class `sm:Concept` sub-classing from the general class `skos:Concept` (the prefix `sm` stands for *service model*). The class `sm:Concept` represents general entities the can build of elements with the same type. For example, we usually define a hierarchy of `sm:Product` instances, where each products consists of hierarchically structured `sm:Component` instances. Usually, a component fulfills a specific `sm:Function`. Final instances of `sm:Component` can be referenced to `sm:Part` instances. Please notice, that a real-world ontology additionally describes more detailed types of components, for example, classes representing electrics/hydraulics, located components, and virtual components.

In Figure 4 we see an example with concrete instances (prefix `ex` is used for example instances). When describing the component *climatic system*, we see that this system (among others) consists of parts like *AC pump* and a *climate control module*. Specific components fulfill specific *functions* of the machine. In the shown example, the component *climatic system* is related with the function *cooling* and the function *heating*. Having represented the product world of the application, we are able to use the instances as annotation values. In the example, the concrete book `ex:book4711` consists of a chapter for replacing the climatic system (`ex:acReplacement`) and a chapter showing the wiring diagram (`ex:acWiring`). The first chapter is assigned to the pump of the climatic system, i.e., how to replace the referred pump. The second chapter depicts the wiring of the corresponding climatic control module.

In an information system, the query "give me information about the component `ex:climaticSystem`" will find both resources `ex:acReplacement` and `ex:acWiring`, since the reasoning will show that the resources are assigned to parts that are included in the climatic system. Also, when looking for electric

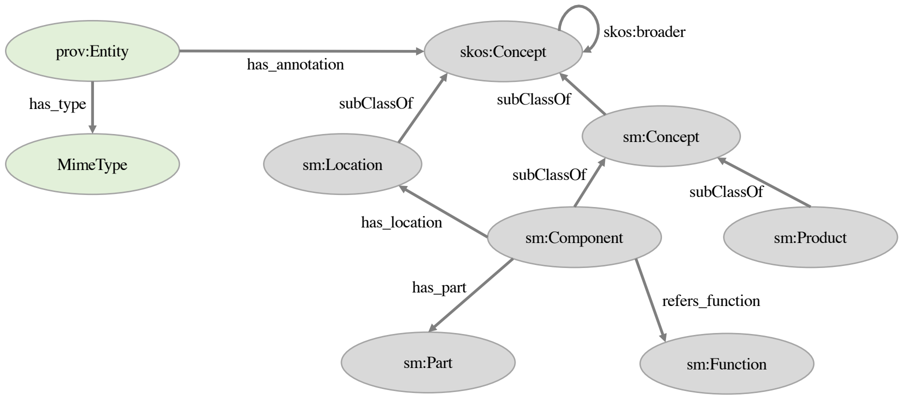

Fig. 3. Excerpt from a domain ontology describing the artifacts of Technical Service.

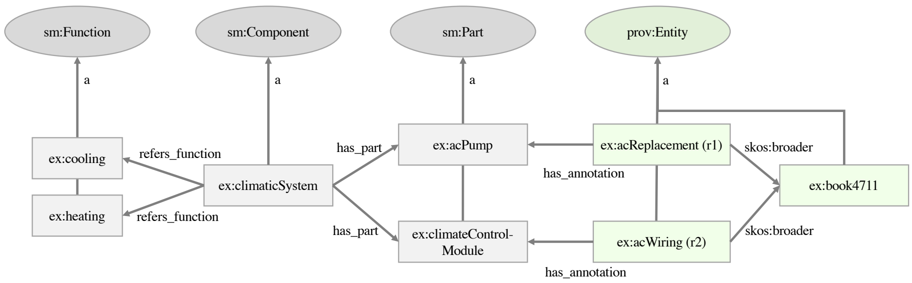

Fig. 4. Instances of the upper ontology describing the component *climatic system* with parts and information resources.

information for the function `ex:cooling`, the system will yield the information resource `ex:acWiring`, since the semantic reasoning will derive this resource as an appropriate (because close) match.

In the following, we briefly sketch how the ontology can be extended in order to fulfill the requirements of an affective information system. All these attributes can be used to further reduce the number of information resources shown as query results but also should be used to present the appropriate user interface depending on the context and state of the user.

3.2 Context-Awareness

In summary, the context of a technician comprises:

- **Product under work**, i.e., the actually considered product the technician is working with.
- **Problem context** describing the activity of the technician, e.g., assembly vs. diagnosis.

For larger problem contexts the current working step is also an interesting attribute of the context.
- **Location** of work, e.g., workshop vs. field.
- **Available equipment**, e.g., which level of tools/measurement devices are available
- Already **consumed knowledge**, i.e.. which information resources the technician already used during the current problem context.

Context-awareness should be primarily used to further filter the query results [1, 3]. Knowing the context of the user, the system can easily remove inappropriate/unhelpful information. In the past, a number of ontologies for representing context-awareness were introduced [4, 12]. When introducing context-awareness for Technical Service applications, these ontologies need to be reconsidered and extended accordingly, as it was already discussed in previous work [8].

3.3 User-Awareness

The user-awareness can be attributed to the following information:

- **Consumed knowledge** identifying which information was already seen by the technician.
- **Emotional state** of the user showing the satisfaction with the current situation. The situation can be possibly derived from the software functions or problem context. The user's satisfaction may be derived by affect detection algorithms and/or using body sensors.

Knowing the state of the user the tool interface should adapt accordingly. Adaptation of the software can be implemented by multi-modal user interfaces. For example, after realizing that a speech recognition did not succeeded for a couple of tries, the system should switch to alternative input interfaces.

3.4 Competence-Awareness

- **User persona** means distinguishing expert vs. apprentice. A user persona is identified in order to present an appropriate interface and information granularity to the user.
- Already **researched and consumed knowledge** can be used to derive the competence level/needed information granularity of the current user.

Competence-awareness should be used to select the appropriate information granularity for a specific query. For instance, the system can decide to immediately show the disassembly of a specific component (expert level) or to first explain the location and operation of the component, before presenting its disassembly (beginners level). The disassembly itself could also distinguish the user's competence: Experts may only need a summary of steps with service-relevant facts (bolt strengths, pressures, etc.), whereas for beginners the systems shows a step-by-step disassembly instruction possible supported by 3D animations for better identification and orientation of components.

4 Discussion

In this position paper we introduced a number of requirements of information systems posed in Technical Service. The requirements are based on a 10-years experience of establishing and improving information systems in this field. We claim, that current state-of-the-art information systems need affective extensions in order to cope with this requirements. In the previous sections we sketched possible ways of extensions. Although the presented approach is by far not exhaustive, we see a field of exciting research questions with a practical motivation in industry.

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. IJAHUC 2(4), 263–277 (2007)
2. Baumeister, J., Reutelshoefer, J., Puppe, F.: KnowWE: A semantic wiki for knowledge engineering. Applied Intelligence 35(3), 323–344 (2011), http://dx.doi.org/10.1007/s10489-010-0224-5
3. Bobek, S., Nalepa, G.J.: Uncertain context data management in dynamic mobile environments. Future Generation Comp. Syst. 66, 110–124 (2017), http://dx.doi.org/10.1016/j.future.2016.06.007
4. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. Knowledge Engineering Review 18(3), 197–207 (2003)
5. Elst, L., Abecker, A.: Ontologies for knowledge management. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. pp. 435–454. Springer, Heidelberg (2004)
6. Furth, S., Baumeister, J.: On the semantification of 5-star technical documentation. In: Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. pp. 264–271 (2015)
7. Guha, R., McCool, R., Miller, E.: Semantic search. In: Twelfth International World Wide Web Conference (WWW 2003) (2003)
8. Legler, A., Baumeister, J.: Towards context-aware technical service. In: Bergmann, R., Görg, S., Müller, G. (eds.) Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. CEUR Workshop Proceedings, vol. 1458, pp. 288–295. CEUR-WS.org (2015), http://dblp.uni-trier.de/db/conf/lwa/lwa2015.html#LeglerB15
9. Moreau, L., Groth, P.: Provenance: An Introduction to PROV. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan and Claypool (2013)
10. W3C: SKOS Simple Knowledge Organization System reference: http://www.w3.org/TR/skos-reference (August 2009)
11. W3C: PROV-O: The PROV Ontology: http://www.w3.org/TR/prov-o (April 2013)
12. Wang, X., Zhang, D., Gu, T., Pung, H.: Ontology based context modeling and reasoning using owl. In: 2nd IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2004 Workshops). pp. 18–22 (2004)