Towards Context-aware Technical Service

Alexander Legler[1] and Joachim Baumeister[1,2]

[1]denkbares GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg, Germany
[2]University of Würzburg, Institute of Computer Science,
Am Hubland, 97076 Würzburg, Germany
{alexander.legler,joachim.baumeister}@denkbares.com

Abstract. Context-aware systems have long found application in everyday use cases, assisting users with their daily lives. Technical service covers any tasks concerning the maintenance, diagnosis, and repair of industrial machinery. It is a more specific domain that would also benefit from the introduction of context-aware systems. This domain requires the filtering and consumption of a vast amount of information resources. Employing semantic technologies enables engineers to more precisely find information as compared to full-text search. However, it still requires a search query to be actively formulated to the system. This paper applies the principles of context-aware systems to information systems for the technical service, where the technician is guided though the service process influenced by various sensors defining their current context. An implementation based on an established ontology for context-aware systems is presented that integrates with semantically enriched documentation.

Keywords: Context-aware Computing, Ontology Engineering, Decision Support

1 Introduction

In the technical domain, the trouble-shooting and maintenance of advanced machinery is a complex task. Technical documentation describes the service-related tasks for these machines and typically comprises some thousand pages of information for a single machine. Consequently finding relevant information bits for a specific fault is difficult and time-consuming.

In the last years, many semantic information systems were introduced in the technical domain to support technicians during service tasks [3]. Semantic information systems add ontological knowledge to the information bits included in standard information systems. In advance to full-text search, such semantic systems introduce semantic search [4], where the retrieval of information is based on semantic queries. Due to the unambiguous query statement, the research time for finding the relevant information is reduced dramatically. Nevertheless, the amount of information is overwhelming in many cases.

 In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

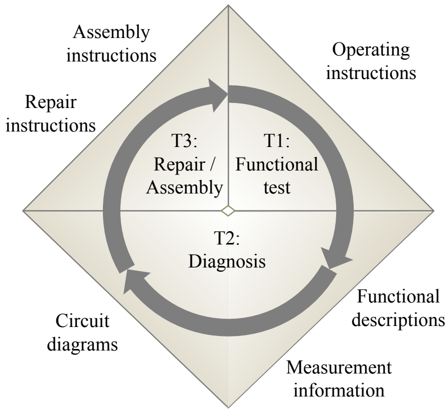

Fig. 1. The typical tasks of a service technician are shown in the center and the relevant information bits in the surrounding space.

In this paper, we propose the extension of semantic information systems by context-aware techniques, that provide the users with relevant information for their current tasks. For example, a service technician in a trouble-shooting task is unlikely to be interested in maintenance information when formulating a query. More generally, by knowing and tracing the context of a working engineer and their corresponding use of technical documentation, the system will be able to provide more relevant information. In Figure 1 the typical trouble-shooting workflow of a service technician is depicted. Essentially, the workflow is partitioned into three sub-tasks: 1. The functional test assures that the failure is actually present, 2. The diagnosis aims to find the cause of the failure, and 3. The repair and assembly fixes the failure. In every task, the service technician has different information needs, i.e. is interested in different types of documentation. The most common documentation types are depicted along the edge of the figure. For example, during the task *functional test* the service technician needs the operating instructions in order to know how the failing function is operated properly. Context-awareness tries to guess the actual task of the technician and to recommend the best-fitting information for the current situation.

The paper is organized as follows: Context-awareness is based on the interpretation of sensors. Thus, we first describe specific sensors with respect to technical service scenarios. For an implementation the context-awareness needs to be represented within the semantic system. Therefore, we introduce an ontological representation and show its application in a selected case study. The paper is concluded with a summary and planned future work.

2 Context-aware Systems in Technical Service

Modeling a specialized application domain such as technical service in a context-based system requires the use of various sources of information. In a context-based system, this information is provided by *sensors* that can be physical, but in this case mostly are *virtual* and *logical* (following the definition in [1]), i.e. providing data from software systems and combining data from other sensors. This is the case as data from a physical sensor, predominantly the location, only affects few environmental parameters and not the machine's overall condition on which the focus lies. Figure 2 shows the main entities (machine, engineer, service case, and location) and their properties that we propose to be described by sensor information.

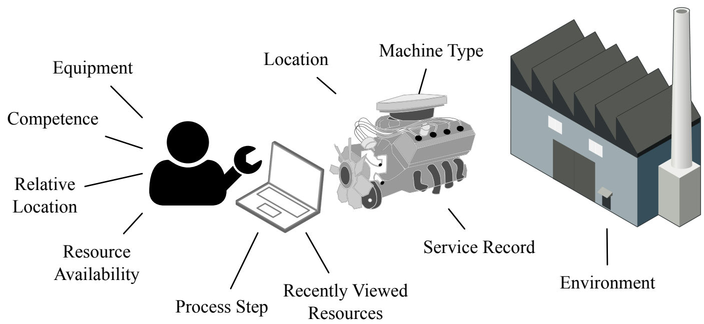

Fig. 2. Additional sensors providing information about an engineer's context while servicing a machine.

The following list describes potentially useful sensors for technical service:

Machine type The most essential sensor captures the exact model and equipment status of the machine that is being worked on. Functions are often provided by different components within a model range, each of which having their own maintenance procedures and documentation. Misidentifying the specific component setup can increase service turnover times or even damage equipment.
There are several feasible implementations: a physical sensor beacon on the machine can transmit its type or a virtual sensor can provide it as manually set information.

Machine service record As the equipment status as provided by the first sensor can change over time, it is useful to provide the service record and part changes as well. This information is also relevant when faults in machines reappear after a period of time. Previous repair attempts can be factored into the support process to directly suggest a remedy or exclude it if it can not fix a fault permanently.

Machine-relative location One of the most common sensor types in everyday use of context-based systems is the location sensor, using AGPS to determine a person's position with an accuracy of up to a few meters.

Location information is also valuable in the target domain, but required on a finer scale. Its use becomes evident when dealing with machines that exceed a certain installed size as it for instance is the case with offset printing machines. Knowing the module at which the engineer currently is located at enables a context-based information system to narrow down the relevant documentation to that specific part.

Engineer equipment Another factor to consider is the equipment available to the engineer. This includes both the available tools in the engineer's toolkit as well as the devices they have available to consume documentation: augmented or virtual reality displays and expert systems may not be available on all device types or require special gear.

Using this information enables a timely detection of faults that are not remediable with the currently available material and improves clarity in the software system by hiding information that can not be displayed.

Information and resource availability The applicability of documentation items is further determined by the resources at the engineer's disposal. Most importantly, it needs to be determined if the engineer has Internet access to reach further materials on a company network. For problems requiring in-depth analysis and triage, the ability to contact off-site support staff may be required as well.

Engineer competence level Given the ever increasing complexity of modern appliances, training engineers is expensive, both financially and in terms of time consumption. The context can factor in the engineer's competence level to offer additional guidance for lesser experienced engineers while not disturbing the workflow of seasoned mechanics with basic knowledge. Additionally, the system can sense when a procedure is potentially unsafe if performed by untrained staff.

Step in the service process This logical sensor captures the step the engineer is currently working on to influence the choice of documentation provided. As service processes are usually provided by the manufacturer and to be followed in a specific order, the position in the overall process can be determined.

Consideration should be given to the level of detail used for modeling the process. The inclusion of atomic steps like removing a screw would incur unnecessary modeling complexity.

History of viewed documents Together with the current process step, the previously used information within this task is an valuable sensor. The already consumed information spans the knowledge and status of the technician and can also be used to deduce the next steps in the service process.

Environment at the repair site Much like the *engineer equipment* sensor provides information about the resources made available by the engineer, this sensor describes the environment at the work site. This information is important as the environment can be vastly different when working in a specialized workshop or on-site at the customer's premises. The latter location will most likely not have specialized equipment for advanced repair scenarios.

3 Ontological Representation of Context Awareness

There are various instances of existing ontological context models, such as the ontologies COBRA-ONT [2] and CONON [9]. In the context of this work, we use CONON as the base ontology due to its simplicity that facilitates ontology reuse and the fact that no other specific domains are included that are of no use for technical service. CONON provides a minimal upper ontology that can be easily extended by domain-specific ontologies, as shown in excerpts in Figure 3. Its root class `ContextEntity` is extended by four general concepts: `CompEntity` (computational entity), `Location`, `Person`, and `Activity`. The list of pre-defined computational entities (not shown) includes `Service`, `Application`, `Device`, `Network`, and `Agent`. Locations are further distinguished between indoor and outdoor places, and activities can either be scheduled, or deduced from the other context sensors (not shown).

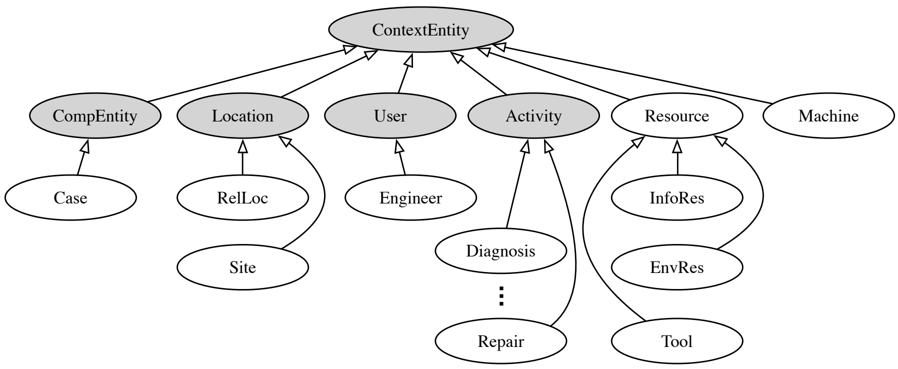

Fig. 3. The CONON upper-ontology (grey, in excerpts) and extensions for the technical service domain.

In the outlined technical service scenario, most classes are intuitively reusable: We extend `Person` with an `Engineer` class representing the technician working on a machine. An instance of this class will have several properties for identification (using SKOS' `skos:prefLabel` [8] or FOAF's `foaf:name`) as well as indications of their training status (`competenceLevel`) and information about provided resources, i.e. the tools they currently carry (`providesResource`).

To be able to further model the technical service domain, we also employ a few other entity classes, directly sub-classing `ContextEntity`. First, a `Resource` is defined as a resource that is available to or required by the engineer. Such a resource can be any kind of information (`InformationResource`), a `Tool`, or an `EnvironmentalResource`. Examples for information resources could be documentation (like sections of a manual, schematics, or wiring diagrams), expert systems, or the possibility to contact other support staff for further consultation. While tools are items contained in the engineer's toolkit, environmental resources are to be provided at the service location (service lift, expensive diagnostic utilities). The `Location` class provided by CONON is used to model the machine's location as well as the engineer's relative location. Its `Site` sub-class is instantiated for each work site to set `providesResource` properties to denote available resources. The other sub-class, `RelativeLocation`, is to be used to capture the current machine-relative location of the engineer. Finally, a `Case` class which is added as a computational entity represents a service case linking engineer, location, and machine. It also contains information on the current state in the process, modeled as instances of CONON's `Activity` class. We define a set of activities representing the service process: `FunctionalTest`, `Maintenance`, `Diagnosis`, `Repair`, etc.

Given the `tso` namespace (technical service ontology) and `ns` for the target application ontology, an exemplary minimal scenario could be as follows:

```
ns:SmallToolkit a tso:Tool .
ns:ServiceLift a tso:EnvironmentalResource .
ns:Machine_1 a tso:Machine .

ns:Engineer_1 a tso:Engineer ;
  tso:competenceLevel 4 ; tso:providesResource tso:SmallToolkit .

ns:Workshop_1 a tso:Site ; tso:providesResource ns:ServiceLift .

ns:Case_1 a tso:Case ;
  tso:locatedAt ns:Workshop_1 ; tso:servicedBy ns:Engineer_1 ;
  tso:machine ns:Machine_1 ; tso:currentStep tso:Diagnosis .
```

An Engineer (`Engineer_1`) has access to the `SmallToolkit` resource. Their competence level in this case is modeled as an integer and at level 4. The service case (`Case_1`) takes place at `Workshop_1` which provides a `ServiceLift` to work on `Machine_1`.

4 Implementation Example

To test the modeled ontology, we employ a small and understandable technical domain, in this case bicycles. We use the namespace prefix of `bts` (short for bike technical service) for framing the concepts of this domain. The example encompasses several bicycle models and contains information about repair steps and resources fulfilling the engineer's information needs while performing them.

We develop a demonstration application for the use by the service technician in the field. It has access to the ontology in order to read and write the context state and access the contained information resources. After setting initial parameters for the case, the application knows what process the engineer is about to begin, as an example diagnosis of faulty headlights on a bike. Initially, the `currentStep` property of the `Case` instance is `FunctionalTest` (c.f. Figure 4).

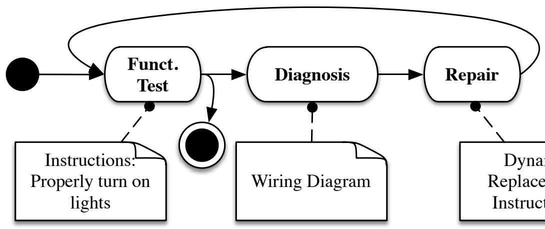

Fig. 4. A simple service process with associated information resources.

For every step of the process, the application queries the ontology of the semantic information system for relevant information, depending on the current context state. This task is performed by the semantic search engine as outlined in the introductory chapter. Context information additionally influences the results, for instance the engineer's competence level should be taken into consideration to ensure they can proficiently perform the actions suggested by the retrieved information resources. This functionality can be implemented using a SPARQL query that yields only resources matching `Engineer_1`'s level of competence for instance by using its `FILTER` functionality.

After reviewing the usage documentation, the current step changes, and so does the context. In the rest of the process, relevant resources for the next steps (diagnosis and repair) are retrieved, until a second functional test results in a working lighting system.

5 Conclusion and Further Work

In this paper, we motivated the introduction of context-based systems into the technical service domain. We proposed a basic ontology that provides a framework for modeling service steps and entities involved in the process of servicing industrial appliances. It can be easily integrated into semantic information systems that use ontologies to represent their semantically enriched documentation as well. In combination, such an application can be used to provide precise information to technicians without the need to manually invoke search operations.

Related work can be found both in different domains as well as system types: The application of context-aware information systems was proposed for instance in the medical domain [5]. The authors also present a specialized approach, like this paper does to allow for the integration of semantic search and domain-specific information sources. Reuss et al. [7] also discuss the application of case-based agents as a form of automated diagnosis support. While this is a different system type, we could envision the combined usage of such a tool and the system we propose in this paper as they both profit from context-awareness to reach the same goal.

The focus of our approach lies in ontology reuse and alignment. Based on an existing lean upper-ontology, we in turn implement a flexible domain-specific layer. Future work will keep this aspect in mind: Modeling the remaining sensor types such as machine history can be done by aligning the established PROV ontology [6]. Further research will be done on the user interface and applicable sensor types, including a survey of other specialized domain sensors that could be adopted. We expect to provide case studies with more complex appliances in the field of agricultural machines and explore reasoning strategies for the resulting, more complex models.

Acknowledgements

The work described in this paper is supported by the Bundesministerium für Wirtschaft und Energie (BMWi) under the grant ZIM KF2959903BZ4 "Mobile, Sprach- und MR-gestützte Dokumentation und Diagnose im Technischen Service (Service Troi)".

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. IJAHUC 2(4), 263–277 (2007)
2. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. Knowledge Eng. Review 18(3), 197–207 (2003)
3. Elst, L., Abecker, A.: Ontologies for knowledge management. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. pp. 435–454. Springer, Heidelberg (2004)
4. Guha, R., McCool, R., Miller, E.: Semantic search. In: Twelfth International World Wide Web Conference (WWW 2003) (2003)
5. Jahnke, J.H., Bychkov, Y., Dahlem, D., Kawasme, L.: Context-aware information services for health care. In: Proceedings Modeling and Retrieval of Context. Ulm, Germany (2004)
6. Moreau, L., Groth, P.: Provenance: An Introduction to PROV. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan and Claypool (2013)
7. Reuss, P., Hundt, A., Althoff, K.D., Henkel, W., Pfeiffer, M.: Case-based agents within the omaha project. In: Vattam, S., Aha, D.W. (eds.) Case-based Agents. ICCBR (2014)
8. W3C: SKOS Simple Knowledge Organization System reference: `http://www.w3.org/TR/skos-reference` (August 2009)
9. Wang, X., Zhang, D., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using OWL. In: 2nd IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2004 Workshops), 14-17 March 2004, Orlando, FL, USA. pp. 18–22. IEEE Computer Society (2004)