Applications and Good Practices of Semantic Technologies

Jürgen Angele, Eddie Mönch, Andreas Nierlich, Henrik Oppermann, Heiko Rudat, Hans-Peter Schnurr

ontoprise GmbH, Amalienbadstr. 36, 76227 Karlsruhe, Germany
{angele, moench, nierlich, oppermann, rudat, schnurr}@ontoprise.de

In the following article we point out the potentials of semantic technologies by the employment within concrete applications. The areas of application cover semantic search systems, which support the discovery of documents and data base information, solutions for the semantic integration of distributed data sources up to semantic expert systems. On the basis two practical examples the applied technologies are illustrated. The first example demonstrates the employment of a ontology-based system for the search and integration of documents and data in a large German telecommunications enterprise. Different sources - e.g. a DMS system and an Intranet - are integrated by ontologies, what enables the cross-over search over the entire system. The second industrial project shows the employment of a semantic expert system to support customer service at a manufacturer of industrial robots. Apart from the applied problem solving methods the project also points out the integration of structured sources. Semantic technologies have left the research laboratories and deliver great benefit in industrial projects. This central statement of our contribution is extended by deep explanations of the applied methods and tools.

Introduction

Whether you work in the development, in the service, in marketing or sales department, whether you constantly master new tasks as a consultant or successfully face the competition as the managing director: You permanently need current and relevant information. More decisions have to be made in ever shorter cycles. There is plenty of information, in enterprises, in the exchange between enterprises and in the Internet. The problem thereby is that there is too much information. The challenge here is making relevant information for everyone accessible and usable in a simple way. To cope with that, IT systems must become more flexible, have to better understand the requirements of the users and them have to support them faster and better. In order to receive a simple and broad entrance to the complex and constantly rising quantity of knowledge, semantic technologies are the best possible fit.

We see that the deployment of this technology can support enterprises of all industries to arrange their development-, service- and selling-processes in such a way that the efficiency and quality can be increased. Our customer projects in the automobile industry, in mechanical engineering, in the public sector and in the telecommunications industry clearly point out the potential of semantic technologies for automation in different line of businesses. In order to be competitive, enterprises must constantly develop faster and more reasonably, increase the quality in development and in service and enhance the efficiency of the sales process. Ontologies as an instance of semantic representation languages are on the way to become the lingua franca of the future. They bridge between humans and computers, between processes and knowledge as well as between IT experts and departments and thereby realize the desired synergetic effects.

In the following we present two Best Practices for the application of semantic technologies. The first example describes the deployment of semantic search for sales support at a telecommunications enterprise. The second example describes the impact of a semantic expert system that is deployed in the customer service department of a industrial-robot manufacturer. Finally we briefly summarize the up-to-date and future range of applications of semantic technologies.

Expert System for Customer Service

The example represented in the following describes the deployment of a semantic expert system to support the customer service department of an industrial robot manufacturer. For many manufacturers of capital goods the optimization of the customer service is an essential process. Here, semantic technologies can help to improve the malfunction diagnosis or to entirely eliminate errors via call center by further inquiries from the system, to train service engineers, to prepare their assignment at the customer-site and support them with the identification of the right solution for the customer problems. For Industrial robots, which are deployed e.g. in the automotive industry, an extremely high availability must be ensured. At the concerned robot manufacturer this was solved so far insufficiently by very simple means such as communication over email, by the use of a simple error data base etc..

By the strong growth of the enterprise, the shorter innovation-cycles, by the increasing customer basis and by the huge variety of different applications, the old approach was not sufficient anymore. Applications range from spot welding in the automotive industry, to gas-schielded arc welding up to maintenance of roller coasters in entertainment parks. This broad product range led to a high number of possible error variants and potential solutions. In parallel with the customer growth the number of service technicians grew very strongly as well. In consequence of that the knowledge about problem solving became wide spread while the demand for certain expertise grew especially among the new service engineers.

Technological basis of the implemented system are ontologies that represent the complex dependencies between the robot components and their applications, enable the access and the integration of existing information sources, enhance document search realize failure assessment. Besides the method for the search for problem solutions themselves is described by an ontology. The implemented system on top of the semantic layer is web-based. Thus, a three-layered architecture results: the existing sources of information in the backend, the semantic layer with the domain ontology and finally the user interface level. In the following we describe in detail the different parts and their dependencies.

Domain knowledge

At the beginning of the project an ontology was developed in close co-operation with the robot manufacturer, which represents the different concepts of robot technology and the interdependencies of error messages and symptoms. It was an important aspect to use a terminology that was understandable for the prospective users, in order to ensure the maintainability after project end. The basic modelling followed the OntoKnowledge Methodology and competence questions in different workshops with experienced service engineers, who are later in charge of the operational system as knowledge base editors. In the initial process step of the methodology the employees formulated specific questions, which the system later would have to able to answer. Out of the questions the most important concepts were identified and represented with their dependencies in the basic modelling. In addition some instances of these concepts were modelled exemplarily.

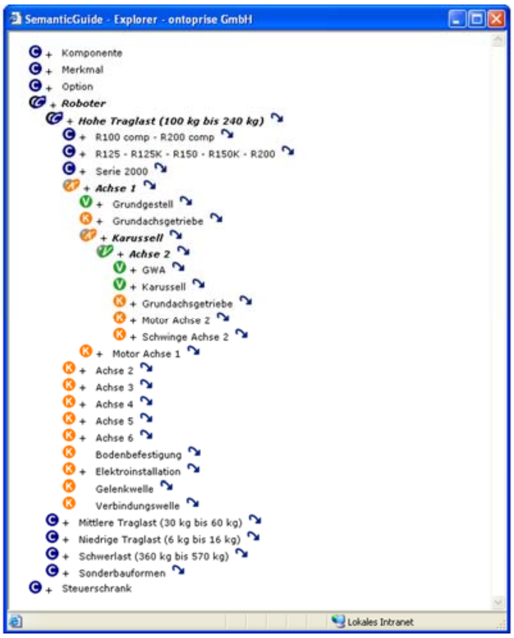

Fig. 1. Domain ontology

Fig. 6 shows a web-based visualization of the taxonomic hierarchy of the ontology. C-label denominate the concepts, indentions are their corresponding sub-concepts.

Thus robots have the specializations "high load", "middle load", "low load", "heavy load" and "special designs". K-labels denominate the different construction parts that belong to a certain assembly. E.g. the "axle 1" is a sub-part "high load" robot. V-labels are relations and mean "connected to". In our example the "axle 1" is connected to the "basic frame". Beyond that the editors modelled errors, subsequent errors and their possible solutions. Rules in the ontology describe e.g. when certain errors can be in and/or excluded:

```
If an error X arises at a certain construction unit and
the current robot does not contain this construction
unit, the error and thus all sequence errors are
excluded, as long as the sequent errors do not have any
others causes.

If an error X concerns certain software release and the
current robot runs with a newer software version, then
the error X and all sequent errors can be excluded.
```

Problem Solving Methods

In order to design the system as generic and thus adaptable to other domains as possible, the method for the search of suitable solutions was modelled independently of the domain knowledge. In particular the requirement was important that after the system suggested a list of possible solutions to the specified error symptoms, the system can ask differentiating questions to step by step reduce the number of presented solutions. Archetype for the modelling were the generic problem solving methods, as they have been discussed in research back in the 90's [Marcus 88]. In our case a simplified variant of the problem solving method Cover and Differentiate [Eshelman 88] was modelled, because it was moist suitable for the existing problem and requirements. Cover and Differentiate is the problem solving method, which was used in the expert system “Mole”[Eshelman and McDermott 1986]. Cover and Differentiate is suitable for covering classification problems, whose solution is a subset from a certain quantity of pre-defined solutions.

This method represents a search algorithm in a directed acyclic graph. The knots of the graph are the states. An edge of the state s1 to the state s2 describes that s2 is a cause for the state s1. Additional knowledge, so called differentiating knowledge allows qualifying or dis-qualifying further states and thus differentiating between different solutions. Two substantial principles are realized by this problem solving method:

- the exhaustive principle means that each arising symptom must be covered at least by a cause. Therefore a condition may not be removed from the further view anymore, if it represents the one and only cause for an observed symptom.
- The exclusivity principle means that for an observed symptom only one cause can be responsible (single fault hypothesis).

The qualifying knowledge was used to reduce the number of possible solutions. Such qualifying knowledge can refer to the conditions or to the edges as well. For a condition and/or an edge in a graph such additional knowledge can be defined. In our system this knowledge was used to place purposeful questions. Thus, the auxiliary knowledge „green oil “e.g. causes in our condition graph that certain conditions were disqualified and by that one of the three solutions could be eliminated.

The problem solving method was modelled itself as an ontology (see Fig. 7). All conditions are represented by the concept *state*. A condition is specified in more detail by a name. Possible causes for a condition are described by the relation *hasCause*. The cause arising in a certain problem solution is connected with the condition by the relation *isEdge*. The concept *EventQualifier* names qualifying knowledge for conditions. An element of this class is assigned to a concrete condition over the relation *forState*. A *ConnectionQualifier* qualifies in the same way the cause relationship between two conditions. The class *ObservedState* contains the actually observed symptoms like e.g. „oil in the arm “and *FinalState* marks the final causes. These are also connected with the proposals for solution by a relation *hasSolution*.

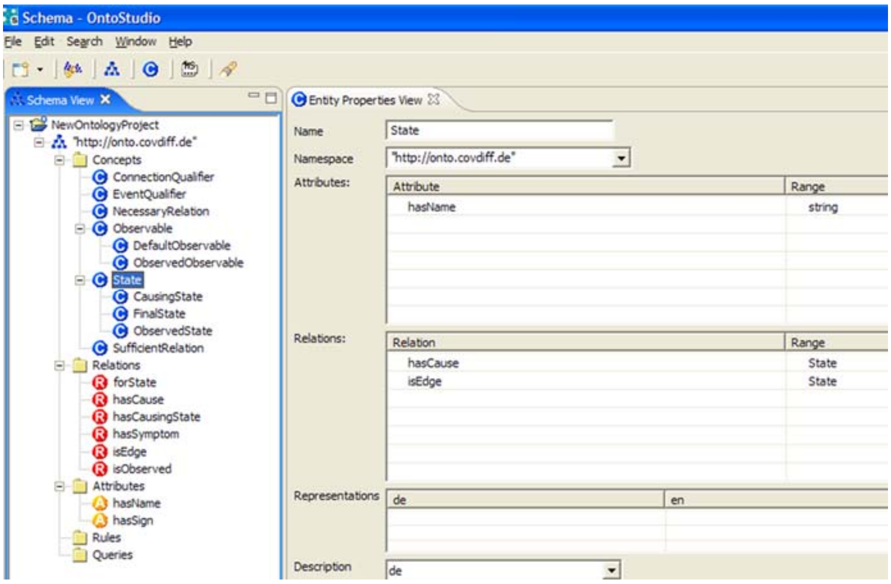

Fig. 2. Ontology of the problem solving method Cover-and-Differentiate in OntoStudio™

Rules describe now the above mentioned principles. Other rules define the paths of observed symptoms to the final solutions. With rules the influence of the qualifying knowledge on the conditions and the edges in the graph is described. Exemplary the rule is described, which defines the different paths and also the influence of the EventQualifier:

```
If for a ObvservedState or a State X the cause is Y and
neither the edge (X,Y) is eliminated by a
ConnectionQualifier, nor the condition Y or the
```

```
condition X over a EventQualifier is eliminated, then
the edge (X, Y) is an edge to a final cause.

If a path leads from an ObservedState X to a State Y
along the relation hasCause, then Y is a possible
cause.
```

One of the strengths of problem solving methods apart from the general validity is also the well-defined requirements, which it has to the necessary knowledge of the domain. It is clear e.g. after selection of Cover and Differentiate that there must be error descriptions, error causes relations and qualifying knowledge. Thus the choice of such a problem solving method naturally affects also the structuring of the knowledge of the application domain. While the terminology and structuring of the problem solving method are independently of the application domain and thus specific for the method, the terminology and structuring of the application domain are specific for these. In order to bring together both models and with it to receive an executable model, a suitable mapping must take place between both models. In Fig. 8 such a mapping was graphically done with the tool OntoMap™.

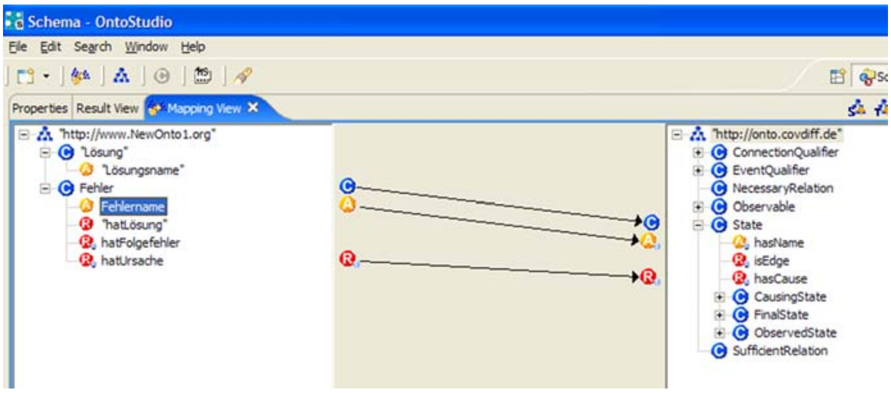

Fig. 3. Mapping between domain ontology and problem solving ontology with OntoMap™

Integration of different data sources

The Ontologie of the application domain, i.e. in our case of robots, their applications, their error and possible problem solutions serves on the one hand for the representation of this extremely complex and interwoven knowledge. On the other hand this ontology serves in addition, for the access and for the integration from existing sources of information. In our case it was about two existing data bases of the development department as well as about several sources of text of work instructions, over quality management references to technical service messages in email format. The integration of the structured information could be accomplished with the data base of the control error messages (on-line assistance of the PC control) and the development error data base. Both data bases describe a similar structure as the modelled errors in the ontology. In addition, an error has one or more solutions and can have a subsequent error. The Ontology serves thereby for the re-interpretation of these sources of information in a uniform and generally understandable terminology and structuring. In addition again by OntoMap™ the sources of information are attached to the ontology „"(see Fig. 9). With a query to the ontology at run-time corresponding queries are then generated to the information sources and thus the current information it contains is accessed.

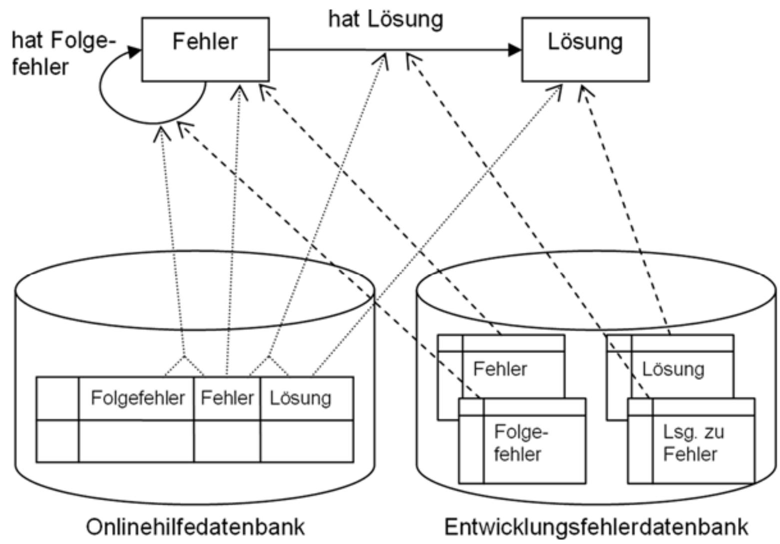

Fig. 4. Integration of existing information sources in the ontology

This enables the usual maintenance of the existing information sources by the system owner while the expert system can access the updated informations.

The integration of the unstructured, textual informations, e.g. the work instructions, quality management references and hand books, was realized by the semantic search technology SemanticMiner™.

SemanticGuide™

The system for the solution of faults, SemanticGuide™ was integrated completely into the existing IT-landscape at the customer side.. The support department uses the module Customer Service (CS) for the collection, completion and invoicing of the support deployment. The module CS is supplemented by SAP Mobile Asset Management (MAM), which is used by the service engineers for the treatment of the service orders. A service order starts with a customer call in the call center. A employee of the telephone support then generates anew order in CS, which contains as accurate and detailed data as possible about the arisen problem and the concerned robot. Subsequently, in CS the generated service order is transferred by a scheduler to a service engineer. Over GPRS the service engineer receives this order with all necessary information in the local MAM on his notebook. After handling the service order at customer side, the needed spare parts and the working hours are entered and the order from the MAM is synchronized back again in CS.

SemanticGuide® was integrated both in SAP CS and in SAP MAM environment (see Fig. 10). The employee in the call center is already supported by SemanticGuide® when he files the order. After the entry of the fault, the system asks questions to narrow down the cause of it. Solutions are already suggested, which the customer can apply. Thus, in best case the deployment of a service engineer possibly can be avoided, but in any case it is possible to contain the problem by the given answers of the customer more exactly. The collected information is transferred from SemanticGuide™ to CS. After transmission of the order to a service engineer these information is present in his MAM as well. With the call from within the MAM, SemanticGuide™ is started exactly the same state, at which the search for the solution was closed in the call center (see Fig. 10).

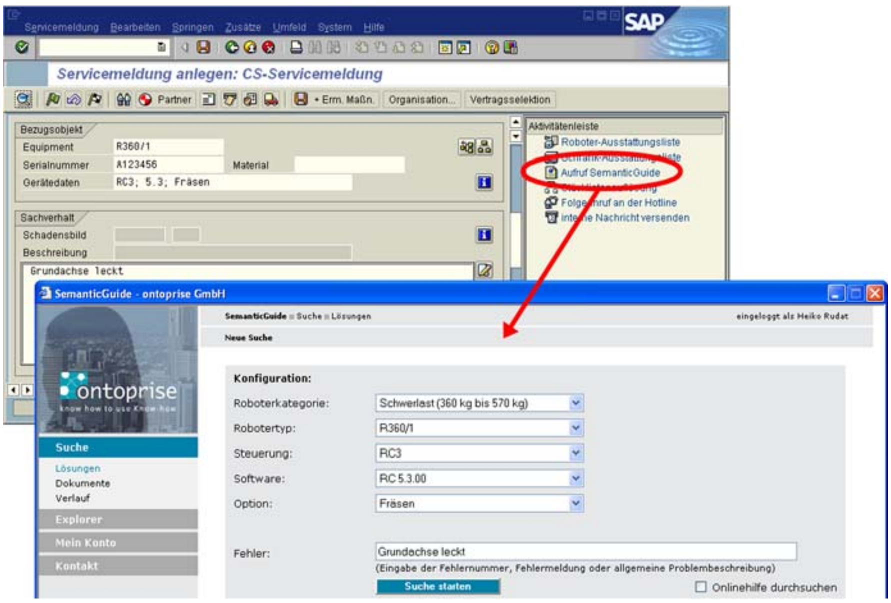

Fig. 5. Integration of Semantic Guide™ in SAP CS and MAM

After the selection of the fault, system internally an alignment takes place that considers the temporary context of the user profile (e.g. SessionID12345 [ausgwählterRob->Robot123; ausgewählteSteuerung->ST1234; …]). At this process a great number of rules are involved, because the solutions are mainly bound to certain parts and the robots or the controller etc. are determined by them (over relations as „isPartOf “or “isConnectedTo“). Subsequently, the results are ranked by a special rule:

```
max (utility) under the constraint of min (costs/time)
```

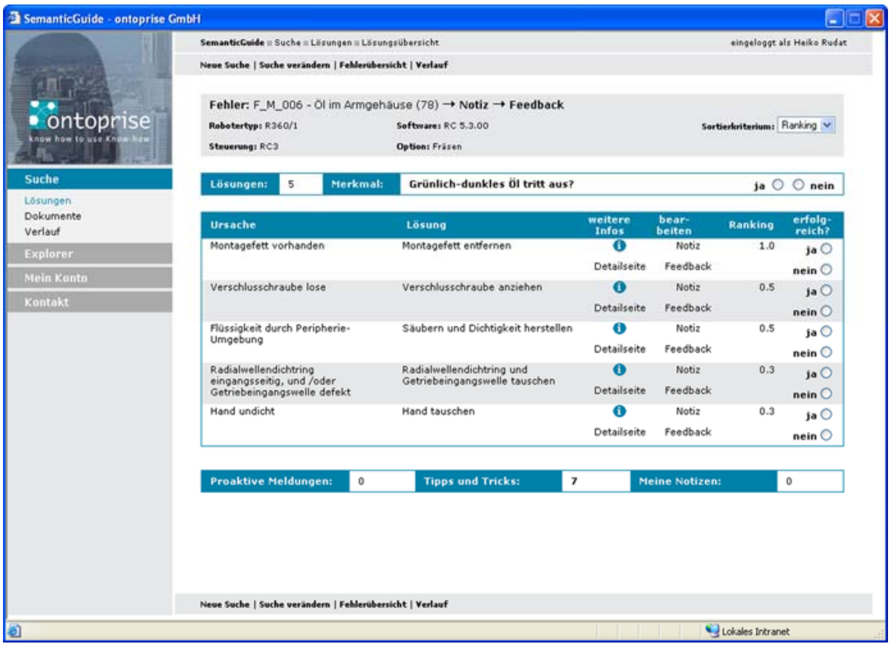

Fig. 6. Containment of a fault by call-backs from the system

For the determination of the containing call-backs, the set of all attributes of the remaining solutions, which belong to the fault are computed. (in Fig. 11. [„greenish-dark oil steps out “in two solutions; „engine X running “ in five solutions; …]). On the basis of the set of the attributes now that attribute is selected, which most frequently occurs in the set („greenish-dark oil steps out “) and in addition the question is generated: „Characteristic: greenish-dark oil leaks? “ Depending on how the question is answered, many of the other solutions can be filtered, whether they possess this symptom or not.

After the robot was repaired successfully all the applied steps, which SemanticGuide™ has suggested, are stored in form of a search log file. After the entry of the working hours by the service technician these information is synchronized with CS and serve as fundamental data for rendering the invoice to the customer.

Conclusion

Semantic technologies support the knowledge worker in the enterprise at his central work tasks. The search for information, are they supporting documents for the sales process or concrete decision supports for the maintenance of a robot, can be improved by the use of ontologies, which contain above all rules that describe the logic of the domain. It is remarkable that semantic technologies have entered in industrial projects considerably far. The described applications are used already by hundreds of users. „The research corner "was left and the customer's utility is clearly the main focus! The current trends in semantics lay in particular in their combination with IT architecture. Here service-oriented architectures (SOA), Web services and Grid technologies are combined with SemanticWeb technology. In numerous research projects (SIMDAT, XMEDIA, NEON), but in several industrial projects as well, interesting starting activities are already in progress. Likewise the W3C works on standards, that enable the comprehensive use of these technologies (RDF, OWL, Rules).

Die described best practices illustrate the opportunities that can be applied already – the current trends clearly indicate the further distribution of semantic technologies in the industrial area.

References

[Belkin 1980] N.J. Belkin. *Anomalous states of knowledge as a basis for information retrieval*. The Canadian Journal of Information Science, 5:133--143, 1980.

[Chapman 2005] Sam Chapman. SimMetrics: An open source extensible library of Similarity and Distance Metrics. http://www.dcs.shef.ac.uk/~sam/simmetrics.html. 2005.

[Cohen et al. 2003] W.W. Cohen, P. Ravikumar, and S. E. Fienberg. *A comparison of string distance metrics for name-matching tasks*. In Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web, 2003.

[Eshelman and McDermott 1986] Eshelman, L; McDermott, J.: MOLE: A Knowledge Acquisition Tool That Uses Its Head. In Boose, J.H.: Gaines, B. (Eds.): AAAI-Workshop: Knowledge-Acquisition for Knowledge Based Systems, Banff, Kanada, 1986.

[Eshelman 1988] Eshelman, L: Mole: A Knowledge-Acquisition Tool for Cover-and-Differentiate Systems. In [Marcus 1988], 1988, p. 37-80.

[Marcus 88] S.Marcus (Ed.): *Automating Knowledge Acquisition for Experts Systems*. Kluwer Academic Publisher, Boston, 1988.

[Moench et al. 2003] Eddie Moench, Mike Ullrich, Hans-Peter Schnurr, Juergen Angele: *SemanticMiner - Ontology-Based Knowledge Retrieval*. In: Special Issue of selected papers of the WM2003 in the Journal of Universal Computer Science (J.UCS).

[Jaro 1989] M. A. Jaro. *Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida*. Journal of the American Statistical Association 84:414–420, 1989.

[Jaro 1995] M.A. Jaro. *Probabilistic linkage of large public health data files (disc: P687-689)*. Statistics in Medicine 14:491–498, 1995.

[Taylor 1968] R.S. Taylor. *Question-negotiation and information seeking in libraries*. College and Research Libraries, 29, 178-194, 1968.