Reuse of STEP product data for the generation of technical documentation

Björn Höfling

Otto-von-Guericke-Universität Magdeburg
Institut für Wissens- und Sprachverarbeitung
P.O. Box 41 20, D–39016 Magdeburg, Germany
hoefling@iws.cs.uni-magdeburg.de

Abstract

In order to enable an effective communication between people inside and outside an organization (e.g. suppliers and customers), a large amount of natural language documents is necessary. The generation of multilingual natural language starting from a language independent representation of the content offers many advantages, for instance, consistency between a product and its documentation in different languages and adaptability to different kinds of users and purposes. The high effort of creating a domain model, which serves as a basis for and input to the generation process, can be reduced if formal knowledge already available in an organization can be reused. In this paper we propose to reuse product data which has been represented with the help of ISO 10303 STEP (**ST**andard for the **E**xchange of **P**roduct model data). In order to be able to exploit this data, the semantic of the respective entities has to be defined more strictly than in STEP. Therefore, we argue that an ontology should be defined at least for core parts of ISO 10303 and propose PDM-schema as a realistic subset with a high degree of usefulness. To make inferences from product knowledge and to allow for a seamless integration with many natural language generation systems, we suggest description logic as a means for defining this ontology.

1 Introduction

A corporate memory can be defined as "an explicit, disembodied, persistent representation of the knowledge and information in an organization" [vHvdSK96]. Heijst et. al. also note that this definition only restricts the form, not the content. "Any piece of knowledge or information that contributes to the performance of an organization could (and perhaps should) be stored in the corporate memory." If one looks at the different kinds of knowledge existing in an organization a great variety appears, ranging from undocumented knowledge such as capabilities and know-how of the employees to computer interpretable data at different levels of formality (from free text to data or knowledge bases). Only those parts of the knowledge which have been represented explicitly in a persistent way can be put together in a corporate memory.

Concerning the degree of formality one point of view is that the more formal an information is represented the more useful it can be. This is due to the fact that formal knowledge can be subject to further processing, e.g. in (management) information systems. On the other side, formal knowledge is not readily understood by an information seeker as it requires a certain familiarity with the syntax and semantics of the formalization. This is seldom the case, especially for the communication between departments or between a company and its suppliers or customers. Therefore, natural language documents are still the major means of communication within a company and for their external relationships. Text is a medium that can be understood without knowledge about a special formalism or representation.

For the manufacturing industry product related documents like assembly instructions, design rationals, end user documentation, etc. have to be created during the product life cycle. Much of the content of these textual documents, like the product structure, technical properties, descriptions of assembly processes, etc., has already been represented in a formal way by product engineers during the design process. Technical editors and other authors of documents normally have access to these sources which are mirrored in the resulting documents. Each time a minor product property changes during the development phase it has to be reflected in all corresponding documents. So each time the document authors are involved. An additional problem arises if the document has to be available in different languages. This is not only the case for end user documentation but also in international companies. Here, additional human translators may be envolved. These problems all together result in a prolonged and complicated document creation process and can considerably diminish the effectiveness of the information flow in an organization.

A solution to these problems can be offered by multilingual natural language generation [RGHH98]. It starts from a language independent representation of a documents content and generates different versions in the target languages automatically. Changes in document content have to be carried out only at one place, the language independent representation, and will be reflected automatically in all corresponding documents. So the consistency between a product and its multilingual documentation is guaranteed and the time between the product change and the availability of all documentation is considerably reduced. An additional advantage is that the resulting documents can be tailored to different kinds of users and purposes and that stilistic constraints can be assured.

One of the reasons why multilingual natural language generation has not yet made a breakthrough in industry is that the effort needed to create a domain model for the content representation from scratch is prohibitive. During the product life cycle much of the information concerning a product is already represented in a formal way by the engineers. Besides proprietary solutions more and more companies begin to use the ISO 10303 standard STEP (**ST**andard for the **E**xchange of **P**roduct model data) for representing their product data. This is made easier by the fact that nowadays most of the suppliers of CAD or PDM systems have integrated STEP into their software. In this paper, we focus on the reuse of already available product data represented with STEP and discuss the steps required to enable its use in automatic generation.

In the next section we present a brief survey of ISO 10303, discuss how it can be used to represent product data and point out some inherent deficiencies of STEP. Section 3 introduces those aspects of natural language generation which are relevant in our context. The following section describes how STEP product data could be employed for natural language generation with the help of additional ontologies. In section 5 we propose to choose a core part of STEP for the implementation of our approach. The paper concludes with related work and a discussion.

2 Product data represented with STEP

Product related information is a key asset in the manufacturing industry because the real value is not lying in a product itself but above all the corresponding information that is created and used during the product life cycle (ranging from development to maintenance and disposal). We will call the latter kind of information *product data* and, in analogy to information technology, we will call *product data technology* the technical methods for dealing with product data. It has been recognized very early that a standardization is necessary in order to allow the effective exchange of product related information independent of particular computer systems and software. Therefore, the development of STEP was initiated in 1984 and since 1994 (the initial release), various parts of the standard have been published by the ISO.

The aim of STEP is "an unambiguous representation of computer interpretable product information throughout the life of a product" (ISO 10303-1). With the help of this representation the exchange of product data covering all aspects of CAD/CAM should be supported. This strong claim could only be realized by splitting up the standard into a series of many structural parts beginning with the most fundamental ones and continuously developing new ones, e.g. for special industrial application areas. In addition to the description of product data models the architecture of ISO 10303 contains also methodological parts. Among the latter are description methods (part 11 and following, describing the fundamentals of how the models are represented), implementation methods (part 21 ff, implementation specific details like the physical file exchange or data access and sharing) and a conformance testing methodology and framework (part 31 ff).

The core part of STEP is formed by the definition of product data models. They are defined at two levels of abstraction. Generic models called the *Integrated Resources* (IR) describe the requirements that are common to all application areas. Specific product data models called *Application Protocols* (AP, part 201 ff) are based on the IR, but adapted to meet the requirements of a specific branch of industry. An application protocol is separated into the following three models: the *Application Activity Model* (AAM) which describes the functional view on the application with its activities and information flow, the *Application Reference Model* (ARM) for the definition of the detailed data model as it is determined by the application experts, and the *Application Interpreted Model* (AIM) which is the result of mapping the ARM onto the IR and therefore guarantees the consistency and integration between the different parts of STEP.

The formal data models of the IR and AP are specified using the *EXPRESS* language and its graphical representation EXRESS-G (cf. ISO 10303-11, Description methods: The EXPRESS language reference manual). It is a specification language (not a programming language) having object oriented properties but also resembling logical formalisms originating from the entity-relationship method. Note that EXPRESS only describes the reference model, i.e. classes of objects. If information about real objects has to be exchanged e.g. between two PDM systems, the classes have to be instantiated and exchanged via a physical file (cf. ISO

10303-21, Clear text encoding of the exchange structure).

Figure 1 gives an example for the definition of the information element `action` from the revised version of the IR ISO 10303-41, Fundamentals of product description and support (Document ISO TC184/SC4/WG12 N107). The declarative construct `ENTITY` is the most fundamental data structure in EXPRESS to define a class of objects. It can include generalisation-specialization properties (syntactically signalled with the keywords `SUPERTYPE` or `SUBTYPE`, not present in the example), different kinds of typed attributes (e.g. the attribute `description` of the type `text`), references to other entities (`chosen_method` refers to an entity `action_method`) and local constraint rules (syntactically signalled with the keyword `WHERE`, an example will be given later).

An action is the identification of the occurrence of an activity that causes an observable result.
An action identifies an activity that has taken place, is taking place, or is expected to take place in the future.
An action has a definition that is specified by an action_method.
EXPRESS specification:

```
ENTITY action;
  name          : label;
  description   : text;
  chosen_method : action_method;
INVERSE
  id : SET [0:1] OF action_identifier
                 FOR identified_action;
END_ENTITY;
```

Attribute definitions:
name: the word or group of words by which the action is known.
description: the word or group of words that characterize the action.
chosen_method: the action_method that specifies the procedure selected to carry out the action.
id: the action_identifier that specifies the identification of the action. The id need not be specified.

Figure 1: Example for an entity definition in STEP (excerpt)

EXPRESS is used in STEP to formally describe structural aspects of data objects. But additional information is necessary to clarify the meaning of the respective constructs. This is usually given by statements in natural language. This informal way of representing part of the semantics of a formal data model causes many important problems (see [Met96] and [GBM97]). Only for constructs like low-level geometric concepts, the common background knowledge is stable enough and well agreed-upon to be able to interpret the formal model correctly. In other cases the meaning is not clear enough to allow the effective sharing and integration of product data. In the example of figure 1 the natural language explanation leaves too much aspects open. E.g. it is not clear what an `observable result` can be, especially because this aspect can not be associated with something in the formal EXPRESS model. A consequence of this general problem is that constructs which are defined in the IR can be interpreted differently in distinct APs.

Despite this problem STEP is increasingly used in industry, especially in international projects where product data has to be exchanged between different companies with a great variety of infrastructure concerning product data technology. To give one example, the consortium which is constructing the Eurofighter (BAe, DASA, CASA and Alenia) has decided to base their product data management on STEP. The progress of STEP is also due to the fact that today most of the vendors of CAD or PDM software include interfaces which are compatible with ISO 10303. Therefore we believe that it is a good investment into the future to consider reusing product data represented in this way for other purposes like the generation of natural language documents.

3 Natural Language Generation

Natural language generation (NLG) can be defined as "the subfield of artificial intelligence and computational linguistics that is concerned with the construction of computer systems that can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information" [RD97]. Recent surveys over the state of the art in NLG are given e.g. in [Bat98], [RD97], and [UHvN+96]. We will summarize in this section only those aspects of NLG that we consider important for our context.

Although there is an ongoing debate about architectures for and appropriate subtasks of NLG, most existing implementations distinguish at least 3 tasks (which may not be sequential and are often not independent): *text planning* comprises the selection of the document content in the form of entities in a domain model together with planning the discourse structure which imposes an ordering and structure on the selected content. In the task of *sentence planning* this information is grouped together into sentences. Further specific words and phrases are chosen to express the domain entities (lexicalization) and referring expressions are generated. Finally, during *surface realization* text is generated using syntactical, morphological and orthographical knowledge.

These different tasks already reveal the types of knowledge that are necessary during the generation process. We will group these knowledge sources into the following three categories:

a) A language independent representation of the *domain knowledge*.The content which has to be communicated in a natural language document is chosen from this knowledge base.

b) *Discourse related knowledge* is necessary to be able to generate adequate text for the current context. Examples of this kind of resource are a user model with a representation of a user's knowledge and expectations, information about the intention of the text or constraints on the medium where the document is presented.

c) Language dependent *linguistic resources* like lexica and grammars for each natural language.

In the context of this paper we can classify the importance of resources necessary for NLG in the following way. The linguistic resources have the advantage of being relatively stable and can therefore be created once or even easily obtained and will then undergo only minor changes. Discourse related knowledge belongs to the responsibility of technical editors and is dependent on the kind of document which is to be created. As it is resource special to NLG it will hardly be available beforehand. The domain knowledge contains all the necessary information about what is to be said. For the manufacturing industry who is constantly developing new products and new configurations or versions this kind of knowledge will undergo perpetual changes. Therefore the domain knowledge necessary as input to NLG has to be at least tightly coupled with the formally represented knowledge in the design process about the current state of a product. In the ideal case, the latter should be directly reused for the generation of technical documentation which is investigated in this paper for the case of STEP.

The next important aspect to examine is, what kind of information does NLG need from the domain? Obviously all entities (objects, relations, activities, etc.) which will be mentioned in the text should be known and also their possible lexicalisations (there may be many at different degrees of abstraction and a flexible generation system should allow this variety). It is generally acknowldeged in the NLG community that there are many other pieces of information necessary which are often not represented explicitly but can be inferred from the domain model. We will give some examples to illustrate these kinds of inferences over the domain knowledge which have an effect on the linguistic realization in the text:

- Part-whole relations can be very helpful for the choice of definite or undefinite articles. A definite article can e.g. be chosen for an object like a motor if another object from which it is a part (like a specific car) has been mentioned in the preceding sentence.
- When similar objects are in the current area of interest it may be necessary to find differentiating properties to be able to clearly distinguish the objects for the reader. Examples of such distinguishing properties may be the relative position (like "the left-most"), different colours, size or other visually detectable features.
- Knowledge about time-relations between processes or events is necessary to choose the tense of a phrase and possibly to use temporal discourse markers for signalling temporal relations between clauses.

These examples show that it is not sufficient to choose just a number of information units in the stage of content selection which then can be verbalized. Additional queries to the domain knowledge are necessary during the subsequent stages of the NLG process. In the following section we will describe a proposal what should be done to allow such queries to product data which is represented with STEP.

4 Towards an ontology for product data

Two problems have appeared in the previous sections which have to be solved to allow the reuse of product data for NLG:

- The semantics of the information units formalized in EXPRESS is not clear enough (cf. section 2). As this has led to ambiguous interpretations in different AP it will also create problems when trying to verbalize product data represented on this basis.
- Inferences have to be drawn from the product data to be able to reveal implicit information which is necessary for queries from NLG tasks (cf. section 3).

We will investigate both aspects in turn. A solution for defining the meaning of the terms used in a domain and to allow the effective reuse of information represented on the basis of these terms can be offered by ontologies. According to [UG96], an *ontology* is "an explicit account or representation of (some part of) a conceptualisation" with a *conceptualisation* being a "world view with respect to a given domain, which is often conceived as a set of concepts (e.g. entities, attributes, processes), their definitions and their inter-relationships".

One might argue that an EXPRESS definition already fulfils this definition of ontology because it is formal and therefore explicit and it is based on a conceptualisation which is documented in the ISO 10303 documents with all its parts. But this explicit specification lacks the following aspects[1]. The terms (e.g. EXPRESS entities) are not really defined in the sense that necessary and if possible sufficient conditions for an entity are specified. In addition, the axiomatisation is very limited (in fact only to constraints between the attribute values of an entity but there are no axioms embracing more than this). One outcome of the last fact is the unclear semantics of STEP entities which could be overcome by defining an ontology for the terms used and their relationships.

Ontolingua [Gru92] has become a quasi standard for formalizing ontologies. Although it can be translated to several other languages it is itself not an implemented knowledge representation language and therefore it does not support automated reasoning like the test whether a model that has committed to the ontology is consistent with the axiomatization of the ontology. Among the numerous paradigms and knowledge representation formalisms that could offer the latter service we believe description logics to be especially suited. Many implemented NLG systems [Pai98] use this paradigm because the necessary inferences can mostly be based on classification (ref. [RM92]) as the core mechanism of description logics. In our group we are using LOOM [LOO91], which is LISP-based and one of the most expressive and fastest implementations of description logics. In the follwing we will give a simple example of how an entitity definition in EXPRESS can be redefined in LOOM using necessary and sufficient conditions and what advantages this can have.

```
ENTITY time_interval_with_bounds
  SUBTYPE OF (time_interval);
  primary_bound : date_time_or_event_occurrence;
  secondary_bound : OPTIONAL date_time_or_event_occurrence;
  derivation : OPTIONAL time_measure_with_unit;
  WHERE
    wr1 : NOT (EXISTS(secondary_bound) AND EXISTS(derivation));
END_ENTITY;
```

Figure 2: Definition of time_interval_with_bounds in EXPRESS

Figure 2 shows an example of an EXPRESS definition of the entity `time_interval_with_bounds`. It is a subclass of the entity `time_interval` (therefore inheriting its attributes). It has one obligatory attribute (`primary_bound`) and two optional attributes. In addition, following the keyword `WHERE`, a constraint is stated that only one of the optional attributes can be present,

[1] Although not being part of the definition of the term ontology, these aspects are usually prominent in implemented ontologies.

not both. Figure 3 shows a proposal for a definition in LOOM. At first glance there might be no essential difference in the structure. It is stated that every `time_interval_with_bounds` must be a `time_interval` (i.e. a subconcept). There are three relations (which correpond to the EXPRESS attributes) from which the first one is obligatory. The range of the relations (cf. value types) is described by a quantification over all the possible values (they must belong to the respective concept). As there is no restriction on the number of values for the second and third relation, there might be no value at all which is an implicit optionality. Finally, the constraint of the `WHERE` clause is described by adding it to the definition of the concept[2].

```
(defconcept time_interval_with_bounds
   :is
  (:AND time_interval
        (:EXACTLY 1 primary_bound)
        (:ALL primary_bound date_time_or_event_occurrence)
        (:ALL secondary_bound date_time_or_event_occurrence)
        (:ALL derivation time_measure_with_unit)
        (:OR
         (:AT-MOST 0 secondary_bound)
         (:AT-MOST 0 derivation))))
```

Figure 3: Definition of time_interval_with_bounds in LOOM

The essential difference between the definition in EXPRESS and the one in LOOM is that the latter actually *defines* the concept with the help of necessary and sufficient conditions. EXPRESS only describes the structure of the entity by stating its supertype, the attributes and additional constraints. The consequences of this representation are that an instance of this entity which is created in EXPRESS could only be tested on conformity to the integrity constraint. An instance in LOOM could for example be first created as an instance of type `time_interval` which would automatically be classified as being of type `time_interval_with_bound` when a `primary_bound` is added and would also automatically loose this property if both a `secondary_bound` and a `derivation` are added to the instance.

In contrast to EXPRESS, LOOM allows to specify the definition of terms and constraints on their interpretation (axioms in the general sense not only constraints over attribute values in an entity). An ontology for the terms used in the formal part of STEP (defined with the help of description logics, in our case LOOM) would considerably diminish possible ambiguities. Whenever possible, the ontology should also make explicit those assumptions in the conceptualization which are not visible in the EXPRESS structure (because they are not needed for the data exchange), but which are stated in the textual description of the STEP standard. In addition, such an ontology would not only solve the problem of the lacking semantics in STEP. At least parts of the inferences necessary for NLG are already possible through the axiomatization of the ontology (e.g. the transitivity of the part-whole relation). Although the axioms in an ontology are not a complete knowledge base, they can form the basis for describing additional knowledge like special rules which may also be company specific and though surpass the extent of the STEP reference models.

[2]Here `NOT(EXISTS(S))` in EXPRESS corresponds to `(:AT-MOST 0 S)`. The change from `AND` to `OR` is caused by moving the negation inside the parentheses

Concerning the inferential capabilities, the choice of description logics for representing product data has the following advantages [3] (see also [HJ97]):

- Instances can be classified automatically depending on their properties, as illustrated in the example above where additional knowledge of an instance-property leads to another class membership.
- Classes can be organized automatically, so that the introduction of a new class would position it at the right place in the class hierarchy.
- It is possible to define instances and classes only partially because of the open world semantics of description logics. Partial descriptions are something very common especially in product design but are not supported in EXPRESS.
- Instances can change their membership to classes during their existence and classes can change their definition during time.
- Description logics has much broader possibilities for representing and checking axioms or integrity constraints.

These inferences could not only be used for interaction between an NLG system and the domain model (see the examples described in section 3), but also for tasks in the context of knowledge based engineering systems [Kne96] like the capture of design rationales which go beyond the recording of informal (i.e. textual) descriptions or the intelligent retrieval of previous solutions.

A first step in deriving an initial LOOM representation from the EXPRESS definition can be done automatically. We have implemented with the help of the NIST EXPRESS Toolkit [Lib93] a partial translator from EXPRESS to LOOM, which in principle transforms entity and type definitions to concepts and attributes to relations with appropriate domain and range. `WHERE` rules have not been treated, because they very often include complete function calls which cannot be transformed automatically. This translation will evidently only provide necessary conditions for a concept definition. Sufficient conditions have to be added by hand if the conceptualization implies that the concept can be completely defined. The same is true for additional axioms. The inspection of already existing ontologies could also be of great help for STEP. For example, the `measure_schema` from ISO 10303-41 dealing with physical quantities has many similarities with parts of the ENGMATH ontology [GO94].

5 A proposal

In the previous section we have shown that EXPRESS models should be enhanced with ontologies defining the meaning of the used terms and their interrelationships in order to enlarge the range of possible applications using product data represented with STEP. Two important aspects hinder the implementation of this proposal:

[3] Of course some of these aspects are also present in other formalisms, but this combination is unique for description logics

- The sheer size of the standard would make it almost impossible to define an ontology for the entire STEP architecture. Even a subpart like AP 214 (Core data for automotive mechanical design processes), embraces in the printed form already several thousand pages.
- The experience has shown that it takes a very long time for the participating countries to agree on all aspects of an international standard.

These problems cannot be solved in general. Thus, we propose to choose a small subset of STEP with high usefulness to prove the advantages of this additional effort. In our opinion, PDM schema [MU98] is an ideal candidate. It is being developed cooperatively by ProSTEP and PDES Inc., describing a core set of entities in STEP which support product data management. PDM schema is interoperable with major APs used in the manufacturing industry and allows the exchange of information stored in PDM systems. Examples of units of functionality include item structure and classification, item properties like mass or costs, document management and effectivity. It has already gained acceptance in industry, e.g. the Eurofighter consortium has recently agreed to use PDM schema for data exchange capabilities.

PDM systems are important and frequently used tools for the persistent storage of product data which is shared among many different people for different purposes. The participation of many of the vendors of PDM systems in ProSTEP and/or PDES guarantees the conformity to PDM schema in the future. To use product data which has been represented in PDM systems based on PDM schema and additional ontologies has the advantage that the acquisition and maintenance of product related domain knowledge could be integrated into the product life cycle and would be done by the product engineers themselves during their normal tasks.

To illustrate the vision behind our approach: If a company's product data has been represented with the help of STEP models together with ontologies in description logic, this part of the corporate memory would be usable for more purposes than are currently possible. Consistency checks and inferences to reveal implicit knowledge would be possible with the help of the clear semantics and the axiomatization. But in our opinion the major advantage of this approach would be that it is a decisive step towards the practicability of NLG by reducing its implementation costs considerably. Companies could then profit from the mentioned benefits (cf. section 1 and 3) by generating many product related natural language documents automatically. Examples of documents which could be generated are: product manuals, quote documents, assembly instructions, or many different kinds of reports that circulate inside a company and that have characteristic properties for beneficial NLG (like multilinguality, stilistic conformance to standards, a broad range of users in number or in prior knowledge which requires tayloring of documents).

The representation of product data and the installation of linguistic resources together with a generator like KPML [Bat97] is of course not sufficient for NLG. The process of content selection has still to be supported and also discourse related knowledge has to be represented (or at least default values for important text types). But this could be supported with the help of authoring tools for technical editors allowing the content selection and construction of discourse representations at a more abstract level and through reuse of schemata which could be filled and specialized [RGHH98]. Nevertheless, a big obstacle for the introduction of NLG in industry can be at least reduced by the reuse of STEP product data for the generation of technical documentation.

6 Discussion

The only project we know of that is generating natural language out of STEP is VOLVEX [DHJ97]. But their focus is more oriented towards the validation of EXPRESS specifications by using NLG. They argue, that as STEP is one of the most complex standards ever developed, generated natural language text explaining the formal model is much easier to validate than the original STEP/EXPRESS specification. DOCSTEP [EK97]) is developing a platform for the consistent creation and management of product related documentation. Of special interest in their context is the integration of STEP and SGML. The architecture of DOCSTEP is including language technology such as tools for controlled language creation and terminology mangagement but does not focus on NLG. We agree in their opinion that the approach of bringing together the product related view of CAD/CAM and the more document oriented view has big potentials for the future. NLG is a very ambitious way of dealing with the problem of integrating both views but should be pursued further because of its enormous benefits.

Unfortunately, it has not yet been possible for us to propose a more or less complete ontology for PDM schema as a proof of concept, because its documentation is at the moment reduced to a description of the current AIM EXPRESS model together with rationales and comments on the conformity with the current reworking of the IR. But we hope that this will change in the near future towards the same thoroughness as in other parts of STEP. Even if it seems not possible for a standard that is as big as ISO 10303 to agree upon ontologies, there should be at least an agreement upon the semantics of the IR to avoid the problems of incompatible APs.

By having a look at the extent of phenomena appearing at the content level in technical documentation which must have a counterpart on the domain model one would notice the following. In many natural language documents, e.g. in maintenance manuals, actions and processes play a major role. The already standardized parts of STEP, however, are much more centered towards product than process models. There are research projects geared to the integration of product and process models in STEP (e.g. [Awi98]) and it is also envisaged for the future evolution of ISO 10303.

References

[Awi98] Birgit Awiszus. STEP als Basis für eine integrierte Produkt- und Prozeßmodellierung. *Produktdaten Journal*, 5(2):50 – 54, May 1998.

[Bat97] J. A. Bateman. Enabling technology for multilingual natural language generation: the KPML development environment. *Journal of Natural Language Engineering*, 3(1):15 – 55, 1997.

[Bat98] J. Bateman. Automated Discourse Generation. In A. Kent, editor, *Encyclopedia of Library and Information Science*, volume 62, pages 1 – 54. Marcel Dekker Inc., June 1998.

[DHJ97] Hercules Dalianis, Anders Hedman, and Paul Johannesson. Validation of STEP/EXPRESS Specifications by Automatic Natural Language Generation. In *Recent Advances in Natural Language Processing (RANLP '97)*, 1997.

[EK97] Kimmo Elovainio and Jürgen Kunz. DOCSTEP - Technical Documentation Creation and Management using STEP. In Graphic Communication Association, editor, *SGML Europe '97 Conference*, pages 43 – 53, Barcelona (Spain), May 1997.

[GBM97] Nicola Guarino, Stefano Borgo, and Claudia Masolo. Logical Modelling of Product Knowledge: Towards a Well-Founded Semantics for STEP. In PDTAG-AM and GB Quality Marketing Services QMS, Sandhurst, editors, *Proceedings of the European Conference Product Data Technology Days 1997*, pages 183 – 190, April 1997.

[GO94] T.R. Gruber and G.R. Olsen. An ontology for engineering mathematics. In J. Doyle, P. Torasso, and E. Sandewall, editors, *Fourth International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1994.

[Gru92] T. R. Gruber. Ontolingua: A Mechanism to Support Portable Ontologies. Technical Report KSL 91-6, Stanford University, Knowledge Systems Laboratory, 1992.

[HJ97] M. Maher Hakim and James H. Garrett Jr. An object-centered approach for modelling engineering design products: Combining description logic and object-oriented modelling. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 11:187 – 198, 1997.

[Kne96] S. Kneebone. Knowledge Technologies and STEP. Technical Report EuroK/T/960701-1/KBEC, ESPRIT Project 9806 (EUROKNOWLEDGE), July 1996.

[Lib93] Don Libes. The NIST EXPRESS Toolkit - Introduction and Overview. Technical report, National Institute for Standards and Technology, 1993. available at http://www.nist.gov/sc4/tools/nist/exptk.tar.Z.

[LOO91] The LOOM Knowledge Representation System. Documentation Package, USC/Information Sciences Institute, Marina Del Rey, CA., 1991.

[Met96] Felix J. Metzger. The Challenge of Capturing the Semantics of STEP Data Models Precisely. In Michael Wolf and Ulrich Reimer, editors, *Proceedings of the First International Conference on Practical Aspects of Knowledge Management (PAKM '96)*, 1996.

[MU98] Bodo Machner and Max Ungerer. Abbildung der Anwenderanforderungen aus der VDAORG mit dem STEP PDM-Schema. *Produktdaten Journal*, 5(1):8 – 12, May 1998.

[Pai98] Daniel S. Paiva. A Survey of Applied Natural Language Generation Systems. Technical Report ITRI-98-03, Information Technology Research Institute (ITRI), University of Brighton, UK, 1998.

[RD97] E. Reiter and R. Dale. Building Applied Natural-Language Generation Systems. *Journal of Natural-Language Engineering*, 3:57 – 87, 1997.

[RGHH98] D. Rösner, B. Grote, K. Hartmann, and B. Höfling. From Natural Language Documents to Sharable Product Knowledge: A Knowledge Engineering Approach. In U. M. Borghoff and R. Pareschi, editors, *Information Technology for Knowledge Management*, pages 151–182. Springer Verlag, Berlin, 1998.

[RM92] E. Reiter and C. Mellish. Using Classification to Generate Text. In *Proc. of the 30th Annual Meeting of the Association for Computational Linguistics (ACL'92)*, 1992.

[UG96] M. Uschold and M. Gruninger. Ontologies: Principles, Methods and Applications. *The Knowledge Engineering Review*, 11(2):93–136, 1996.

[UHvN+96] H. Uszkoreit, E. Hovy, G. van Noord, G. Neumann, and J. Bateman. Language Generation. In R.A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue, editors, *Survey of the State of the Art in Human Language Technology*, chapter 4, pages 161 – 189. Center for Spoken Language Understanding (CSLU), Oregon Graduate Institute of Science and Technology, 1996. available at http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

[vHvdSK96] G. van Heijst, R. van der Spek, and E. Kruizinga. Organizing Corporate Memories. In *Proc. of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW 96)*, 1996.