Knowledge-rich indexing of learning objects
Kabel, S.C.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
3 The role of ontologies in the material handling processes

This chapter studies the material handling processes in a case. This chapter is based on several publications. Four papers were published, one at the 9th International Conference on Artificial Intelligence in Education (Kabel, de Hoog & Wielinga, 1999); one at the 3rd Belgium-Netherlands Conference on Artificial Intelligence (Anjewierden & Kabel, 2001); and two at the 6th International Intelligent Tutoring Systems conference (de Hoog et al. 2002a; de Hoog et al, 2002b); and a report was published by the Netherlands Organization for Applied Scientific Research (TNO, Physics and Electronics Laboratory (Kabel, de Hoog and Wielinga, 2000). This chapter is also based on IMAT project deliverables.

3.1 Introduction

In this chapter the role of ontologies in the material handling processes is illustrated in a case study in a technical domain. In technical domains such as described in this chapter, the necessity to re-use existing sources has been evident for some time, but is still pertinent. The idea to re-use content was recognized in for instance CALS (Continuous Acquisition and Lifecycle Support) initiated in 1988 by the US DoD. Their slogan “store once, use many times” reflects that idea. CALS proposed that the exchange of technical information between government, weapon systems suppliers and their subcontractors should be undertaken and controlled electronically for the lifecycle of the system. This resulted in structured content in the form of IETMs [Interactive Electronic Technical Manuals] of different classes, varying from hypertexts to knowledge based systems [IETM Guide, 1999; Mil-std-3001, 2001]. The initiative quickly spread to the civil sector because many business processes and underlying concepts are not specific to defense. In technical domains, for example where equipment has to be maintained and repaired, training material is often created in-house by the buyer of the equipment (usually an instructor or teacher), because there are no publishers for text books for those limited domains. The number of students is often low, and/or the system or machine people have to learn about frequently changes, making the training material outdated fast. It is not feasible or attractive for training companies to give the training, nor is it profitable for educational publishers to publish the training material. As a consequence the major source of content is the technical manual delivered by the vendor. Technical manuals often come in large volumes and usually a large amount of (electronic) material is available. Unfortunately these manuals are most of the time not immediately suitable for training and instruction, leading to tedious and time consuming re-working of the source material. Rather than creating instructional material from scratch, development of the material can be made cheaper by re-use existing sources, certainly in these domains. Technical manuals are relatively structured documents suitable to be analyzed automatically. Segmentation of technical manuals, which are of an informational nature, leads to fragments that must be stored, retrieved and enhanced through instructional editing before the material is ready to be used in a lesson.
The role of ontologies in the processes of handling material is illustrated with the IMAT project since this project covers the steps necessary to go from source material to instructional material. IMAT was a 2.5 year lasting Esprit project that aimed to facilitate the re-use of technical manuals for maintenance training in industry (Barnard, Grandbastien, de Hoog & Desmoulins, 2002). To this end tools and methodologies were developed that break up the manuals into re-usable chunks or fragments; index these fragments from different perspectives and store them in a database, from which they can be retrieved and re-assembled into instructional material using any authoring tool. Three domains were addressed in the IMAT project: aircraft maintenance, car repair, and the programming and configuration of traffic control equipment.

Section 3.2 describes ontology types included in a knowledge-rich annotation structure. Section 3.3 describes the role of ontologies in the analysis, indexing, retrieval and re-use processes and illustrates how problems can be solved with the deployment of ontologies. Section 3.4 describes the IMAT case, and Section 3.5 provides conclusions.

3.2 Ontology types in a knowledge-rich annotation structure

A knowledge-rich annotation structure for indexing material reflects an abstract view of the world that one wishes to represent. As was briefly explained in Chapter 2, three perspectives on material must be covered in an annotation structure. Three perspectives, shown in Figure 3-1, are taken on the material that is handled. First, the physical appearance of the material itself (what is it?), for example "a video shot in mpeg format". Second, the domain (what is it about?), for example "the assembly of a steering wheel". The third is the perspective of a task carried out with retrieved material (what can one do with it?), for example including a "demonstration" in instructional material.

Figure 3-1 Perspectives on material.

6 The Integrating Manuals and Training (IMAT) project (ESPRIT 29175) was partly funded by the European Union.
7 The IMAT annotation structure was developed in 1999, in parallel to the working drafts that later became the LOM standard. A detailed comparison of the LOM standard and the IMAT annotation structure is not the purpose of this thesis.
These three perspectives on material are expressed in three types of ontology: ontologies that represent physical aspects of the material, ontologies that represent domain aspects, and ontologies that represent aspects of a task. These types of ontology are used in the processes underlying the handling of material, shown in Figure 3-2.

![Figure 3-2 Ontology types used in material handling processes.](image)

Ontologies are used at several points in the material handling processes. To analyze and carve up a source document, knowledge about the document structure (the physical appearance of material) is necessary. In addition, knowledge about the domain is needed to obtain fragments with coherent topics. These types of knowledge are captured in ontologies about the physical appearance of material and the domain. To support the re-use process, knowledge of the instructional design task is required, captured in a task ontology. For storage and retrieval, all types of ontology are used. These three types of ontology together cover relevant aspects of material in an annotation structure necessary to retrieve fragments and to create instructional material. Material can be seen as content annotated with a set of attribute-value pairs. The annotations correspond to concepts in one of the ontologies that regard the physical appearance of material, the domain, and the task, and the user can specify any combination of keys to retrieve the material based on the annotations. To illustrate what kind of concepts are used for retrieval and re-use, Table 3-2 shows the perspectives taken on handling material (left column), reflected in ontologies (middle column), and the type of concepts used for annotating material and specifying an instructional design (right column).
### Table 3-1 Perspectives, ontologies and top-level concepts.

<table>
<thead>
<tr>
<th>Perspective</th>
<th>Ontology</th>
<th>Top-level concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>Fragment</td>
<td>Medium Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Representational Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structural Type</td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td>Description Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description Scope</td>
</tr>
<tr>
<td>Instructional Description</td>
<td></td>
<td>Knowledge Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Instructional Role</td>
</tr>
<tr>
<td>Domain</td>
<td>Domain</td>
<td>Topic</td>
</tr>
<tr>
<td>Design task</td>
<td>Instructional Design</td>
<td>Learning Goal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Instructional Strategy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Instructional Activity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Instructor Action</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learner Action</td>
</tr>
</tbody>
</table>

Except for domain ontologies, the ontologies are generic, meaning that they are applicable to material in general, or to instructional design and development in general. The ontologies in Table 3-1 (Kabel, Riemersma & Wielinga, 2000; Kabel & Wielinga, 1999) are described in more detail in the next sections.

#### 3.2.1 Physical aspects of material

The fragment ontology covers morphological and syntactical aspects of material. Figure 3-3 shows examples of concepts in the Medium Type, Representational Type and Structural type branches in the fragment ontology.

![Figure 3-3 Examples of concepts in the fragment ontology.](image-url)
The Medium Type attribute covers formats of text, image, audio and video fragments. Representational Type stands for ways of representation. Representational Type concepts concern the visual appearance of a fragment, for example a “body of text” or a “table”. Structural Type covers structural properties of the material. Structural Type concepts concern the structural appearance of a fragment, for example a “paragraph” or a “section”. The Representational Type and Structural Type vocabularies are used to classify layout and logical components in a document. The attribute Learning Resource Type in the LOM standard described in Chapter 2 is a mix of Medium, Structural and Representational Types. Taking an example of a picture fragment used in the domain of motorcycle maintenance, shown in Figure 3-4, the Medium Type and Format would be “image, .jpg”; the Representational Type would be “pictorial representation, structured representation, wiring diagram”, and the Structural Type “document segment, figure”.

Figure 3-4 Example fragment.

The description and instructional description ontologies cover semantic aspects of material. Figure 3-5 shows the description ontology with Description Type and Description Scope concepts.
Description Type allows classifying material in terms of the kind of description or depiction. Description Type concepts concern the perspective from which material was written, for example a “physical description” of components. Description Scope stands for the scope of a description or depiction. In a fragment, one or more concepts (or ideas) are described or depicted. A concept is furthermore described or depicted in general terms or elaborately. For example, the fragment in Figure 3-4 would be annotated with the Description Type “structural description”, and Description Scope “single elaborate” meaning that in the image a single concept is depicted in an elaborate (detailed) fashion.

Figure 3-6 shows the instructional description ontology containing the vocabularies to annotate material on Knowledge Type and Instructional Role.
Knowledge Type allows classifying material in terms of the kind of knowledge embedded in the material. In the example fragment in Figure 3-4, the Knowledge Type is “knowledge of structure” and Instructional Role “explanation” (the example fragment could be used to explain the composition of parts and functions between them). Top-level Knowledge Type concepts are inspired by the work of Gagné, Briggs & Wager (1992) who identify five categories of learned capabilities: intellectual skills, cognitive strategies, verbal information, motor skills, and attitudes. These categories of learned capabilities are re-interpreted in the light of modern knowledge engineering (Schreiber et al. 2000). Instructional Role covers the possible roles a fragment can play in an instructional context, for instance a text fragment can play the role of “definition” in instructional material. Instructional Roles are categorized into concepts that are typically used before (pre-instructional), during (instruction/learning block) and after instruction (consolidation).

3.2.2 Domain aspects
A domain ontology is a conceptual representation of some specific domain, allowing classifying material in terms of the Topic treated in the content. For example, the fragment in Figure 3-4 would be annotated with the Topic “alarm switch”. Building an ontology for a particular domain from scratch requires a profound analysis, revealing the relevant concepts, attributes, relations, constraints, instances and axioms of that domain. Design principles for building an ontology from scratch are described in Gruber (1995); van Heijst, Schreiber, and Wielinga (1997); and Uschold and Gruninger (1996). In general, to build a domain ontology, one should think of naming conventions in advance and define unique concepts. One should represent synonyms and abbreviations in attributes, in order to be able to recognize concepts in the material and in queries. Furthermore, different generality levels have to be distinguished, corresponding to different levels of re-usability. Overviews like an index, table of contents, a list of parts or a tree diagram can often serve as good starting points for creating a domain ontology from scratch, because they contain (often structured) lists of relevant terms. Constructing a domain ontology can also be a matter of assembling existing ones. The simplest way to combine ontologies is through inclusion. Inclusion of one ontology into another has the effect that the composed ontology consists of the union of the two ontologies (their classes, relations, axioms). In other words, the starting ontology is extended with the included ontology. Another way to combine ontologies is by restriction. This means that the added ontology only is applied to a restricted subset of what it was originally designed for. Conflicts between names have to be resolved, and a good tool for engineering ontologies should take care of that. See Corcho, Fernandez-Lopez & Gomez-Perez (2002) for an overview of methodologies, tools and languages for building ontologies. Recently there is significant progress in semi-automated development of domain (and task) ontologies, (Anjewierden, Wielinga, de Hoog & Kabel, 2003).
3.2.3 Task aspects

The task ontology for instructional design comprises 5 top-level concepts (shown in Table 3-1).

- Learning Goal
- Instructional Strategy
- Instructional Activity
- Instructor Action
- Learner Action

These concepts represent the steps a developer of instructional material has to take in instructional design methods (Gagné et al., 1992). Instructional design tools such as Designer’s Edge also support similar instructional design steps.

A Learning Goal describes the desired outcome of an instructional curriculum in terms of knowledge, skills and attitudes that are necessary to carry out certain tasks. In general a learning goal refers to the knowledge, insight and competencies a learner has to master, which is reached through motivation, communication of information, information processing, or storing and recalling information. Learning goals are often formulated specifically for the task that has to be learned and will therefore be explained in the context of the IMAT project (Section 3.4.1). Knowledge is communicated to the learner by means of Instructional Strategies. Figure 3-7 shows Instructional Strategy concepts (based on Chen, 1995).

Instructional Strategies refer to an effective way of teaching or to characteristics of an effective learning environment.

Instructional Strategies can consist of Instructional Activities. Figure 3-8 shows examples of Instructional Activity concepts.

---

Instructional Activity concepts are based on Gagné's nine events of instruction: gaining attention, informing the learner of objective, stimulating recall of prior knowledge, presenting the stimulus material, providing learner guidance, eliciting performance, providing feedback, assessing performance, and enhancing retention and transfer (Gagné et al., 1992). Gagné is considered to be an important contributor to the systematic approach to instructional design and training. Gagné is also known as a behaviorist, meaning that the focus is on the outcomes (or behaviors) that result from training. In the maintenance training design context in the IMAT project Gagné's nine events of instruction were considered to be appropriate.

Instructional Activities can comprise certain Instructor Actions and Learner Actions. Figure 3-9 shows examples of concepts.

Instructor Actions stand for everything the instructor does within a certain Instructional Activity; Learner Actions stand for everything the learner does within a certain Instructional Activity.

The choice of concepts for instructional descriptions of material and instructional design are subject to debate in the instructional design community. The instructional description and instructional design ontologies described in this chapter are derived from the literature.
(Chen, 1995; Gagné et al., 1992). Other taxonomies for sequencing structures exist, as described in Chapter 2 (IMS Learning Design Information Model, 2003; Merrill, 1997; Wiley, 2000). Section 3.3 describes how the ontologies are used for segmentation of source material, indexing, the specification of an instructional design and retrieval of fragments relevant for a design.

3.3 Using ontologies for analysis, indexing, design and retrieval

From an indexing point of view, there are different ways to index fragments. Some index values can be derived automatically from the source, others need to be deduced using mappings between ontologies and the content in material, combined with heuristics, while still other value types can not be determined up-front and need to be added to fragments while using them. The way to index material depends on the time at which the information is available: attributes about the physical aspects of the material itself and the domain can be determined before storage, whereas task-related aspects are determined while specifying the instructional design task. Concepts from different ontologies are connected in such a way that when the steps in an instructional design are specified, relevant fragments can be retrieved based on physical and domain-related attributes. In this way, the designer’s focus shifts from the retrieval task to the work task of creating instructional material. Together the vocabularies in these ontologies serve as a rich annotation structure allowing for flexible retrieval from different perspectives.

The role of ontologies in handling material is elaborated in the next sections. While automation is possible to some extent, all ontologies can of course be used by a human actor to index and retrieve material.

3.3.1 Automatic segmentation and indexing on syntactic aspects of material

Segmentation can be done manually or automatically, depending on the extent to which a source document is structured. A novel, for example, is relatively unstructured as opposed to a TV-guide. A TV-guide would be easier to analyze automatically because of its structure and because the individual segments still have meaning when lifted out of their original context. An unstructured document generally requires human interpretation and will have to be segmented manually.

Dependent on what is known about a document in advance, a top-down or bottom up approach to discover the logical structure is possible. If a document contains systematic markup explicitly expressed in a markup language, this markup can be used to discover the logical structure in a document. However, in practice this is problematic. Even though markup languages such as HTML allow expressing a document structure and style, this is not always expressed in markup in a systematic way. For example, a new chapter can be indicated by the HTML tag “heading 1”, but often people use font face properties rather than tagging to indicate that text is a heading. If “heading 1” is not explicitly expressed in HTML markup, algorithms must be used to discover that text with a large font face can be a heading. A bottom up approach is necessary if a document contains only layout information, as for example in PDF. The layout information is analyzed to reconstruct the hierarchical document segments.
To support segmentation of source material, ontologies can be used as a knowledge base in the document analysis process. Automatic annotation can be achieved for syntactic aspects of material:
- a fragment’s Medium Type and Format
- a fragment’s Representational Type
- a fragment’s Structural Type

Concepts that refer to the syntactic aspects of material that occur in the content of a fragment can be identified and be used for identifying the logical document structure. To find the logical document structure, a document analysis system reasons about geometry and formatting properties in a document. For example, parts of a document are classified as a header or footer, or paragraphs are aggregated into sections. Once the logical document structure is identified the constituting fragments are indexed accordingly with the concepts in the ontologies.

3.3.2 Automatic indexing on domain and semantic aspects of material

Indexing with ontologies can be done manually or automatically, depending on the attributes that have to be described. Automatic indexing can be achieved when the attribute values can be directly derived from the source document or deduced using additional knowledge. Automatic indexing with ontologies is essentially defining a mapping between a concept in one of the ontologies and fragments in the document. The ontologies that can be used for automatic segmentation can also be used for automatic indexing. Attribute values of Medium Type, Structural Type, and Representational Type can be derived automatically from the source and can be added as index to the fragment (see Section 3.3.1). Other general syntactic attributes of which values do not come from ontologies, for example Size for text fragments and Width and Height for images can also be derived from the source and added as indexes. In addition, the domain topic as well as semantic attributes that characterize the material can be added automatically to document fragments. Besides syntactic aspects of material, automatic annotation can be achieved for:
- a fragment’s domain Topic
- a fragment’s Description Type
- a fragment’s Description Scope

Domain ontologies can serve to identify and index fragments on Topic by matching terms in the title of a fragment to terms in the ontology. If the title of a fragment maps onto a concept, then the Topic of the fragment becomes the name of the concept and recursively all sub-fragments will also be indexed under this Topic. The idea is that if a section is about some concept, all sub-sections will also be about that concept. Automatic indexing becomes more complicated when the concepts in an ontology do not occur in fragments but have to be identified using heuristic rules, as is the case for the type and scope of a description or depiction. Additional knowledge is necessary to discover attribute values for Description Type and Description Scope. A way to derive Description Type is to map the words in the body of a fragment to a set of terms that indicate a high probability of a certain Description Type. For example, if combinations of terms such as
“weighs ... kilo” occur in the content of a fragment, the fragment is likely to be a “physical description”. The Description Scope of a fragment can often be found by considering the relative position of the fragment, assuming that the style of the source document is such that general descriptions are followed by elaborations. Another general rule is that the deeper the nesting the more detailed the content will be. For example, if a section heading is “threats to gorillas”, and a subsection heading is “illegal poaching”, the first is expected to have general description scope while the latter will probably have an elaborate scope. The heuristic rules that support the derivation of semantic concepts like these do not always guarantee correct indexes.

### 3.3.3 Manual indexing on semantic aspects of material

The attributes Knowledge Type and Instructional Role are closely related to a design task, but reflect aspects of material that can be determined independent of a design task. To support the development of instructional material, these should also be added as indexes. Annotation must be done manually for:
- a fragment’s Knowledge Type
- a fragment’s Instructional Role

The Instructional Role attribute allows for multiple values because a single fragment can be used in different instructional contexts.

### 3.3.4 Retrieval and re-use, manual task-based indexing “by doing”

As noted in Chapter 2, a task ontology covering aspects of instructional design can serve as a blueprint for the development of instructional material. This blueprint takes the form of a description of the task of the instructional designer. A task description of instructional design starts with specifying learning goals. Learning goals, combined with the type of knowledge that the learner should acquire, point to a certain instructional strategy. Subsequently, the instructional activities and the various instructor and learner actions that have to be carried out are specified. For example, in the context of maintenance training a Learning Goal is “recognition of normal/deviant appearance of components” for which “perceptual knowledge of structure” (Knowledge Type) is required and which is taught through “articulation” (Instructional Strategy). An Instructional Activity would be “presenting” the states of components. A teacher “presents” (Instructor Action) an “illustration” (Instructional Role); the learner “studies” (Learner Action) it.

In Chapter 2 it was also noted that available standard annotation structures to some degree lack knowledge-rich descriptions of semantic aspects of (information) fragments, which are necessary to support development of instructional material. The ontologies described here are an example of a knowledge-rich annotation structure, allowing detailed and structured descriptions of physical, domain-related, and task-related aspects of material. The task ontology for instructional design described in Section 3.2.3 supports constructing the process of creating instructional material. Development of instructional material is supported by specifying each step in the instructional design, such that a blueprint is created that can be filled with fragments suitable for each step.

Section 3.4 describes a case study that investigates how different types of ontologies can sup-
port the transformation of material between the processes, and support flexible, task-based retrieval of material.

3.4 IMAT: re-use of technical manuals in maintenance training

The problem of analyzing content and organizing it in such a way that retrieval and re-use is supported was investigated in the IMAT (Integrating Manuals and Training) project. In this project technical manuals of complex industrial systems are re-used in maintenance training design. The aim of section 3.4 is to demonstrate how ontologies, implemented in tools, support the processes of handling material in practice. IMAT serves as a good example of all processes from analysis of source documents to re-use of fragments in instructional material. Figure 3-10 again shows the processes of handling material, but with the tools that have been developed in IMAT to support the separate transformation stages.

![Figure 3-10 Tools supporting material handling processes.](image)

The tools make use of the ontologies that have been described earlier (see Table 3-2), that is, with some extensions. AIDAS, a document analysis and indexing tool supports the analysis process. This tool automatically carves up technical manuals into text and image fragments, indexes these fragments using the fragment, description and domain ontologies and exports them to an object-oriented database, which facilitates the storage process. An authoring environment interface supports the retrieval process with extensive search possibilities and allows adding indexes such as Knowledge Type and Instructional Role manually. Retrieved fragments can be transferred to an authoring environment of choice. A scenario tool is developed to support the re-use process. This tool helps the instructional designer to define an instructional scenario, and produces the characteristics of fragments that are relevant for the specified scenario, which can be used for retrieval.

3.4.1 Ontologies in IMAT

Some extensions of the physical and task ontologies were necessary to specifically describe the type of source material of concern in IMAT: technical manuals, and to specifically suit the type of instruction in IMAT: maintenance training. To this end, the Description Types are refined in order to describe technical manuals. This also goes for detailed Knowledge Types that are specific for maintenance training. A concept hierarchy of Learning Goals, which are specific for the maintenance training task, had to be developed from scratch. In
the kind of technical training dealt with in IMAT, the following learning goals are defined to suit the maintenance training task (see Figure 3-11).

- learning goal
  - being able to do subsidiary tasks
  - being able to perform all maintenance tasks
  - being able to reason about a system
  - functional knowledge of a system
  - structural knowledge of a system
  - naming of subcomponents
  - location of subcomponents
  - perceptual knowledge about subcomponents/recognition
  - recognition of normal/deviant appearance of components
  - structural description of subcomponents
  - structural layout of subcomponents

Figure 3-11 Examples of Learning Goal concepts.

In addition, for each application domain in IMAT (weapon systems, cars and traffic control systems) specific domain ontologies were developed. Such domain ontologies contain hierarchies of components and functions, part—of relations, operations etc. These domain ontologies were built from scratch using several aids. The generic description type ontology (see Figure 3-5) served to identify perspectives from which a system can be described, for example physical and functional system descriptions. Furthermore a generic system ontology that identifies system components served as a skeleton for building specific ontologies. For example, from a physical perspective a system can be decomposed at various levels of aggregation. In abstract terms, a system can be decomposed into “compound system”, “equipment”, “system”, “subsystem”, “unit”, “subunit”, “component”, and so on. In specific terms, for example the car domain ontology, these physical descriptions would be something like “car”, “bodywork”, “door”, “door handle”, and so on. The technical manuals often offered very structured overviews, divided into physical, functional and operational parts, used as starting points to create the specific domain ontologies. For example, a list of system parts served as a good representation of the physical elements of a system; the functions of a system listed in the table of contents served as a good representation of a system’s functions. Table 3-2 shows an overview of generic and specific levels of applicability of the various ontologies, including example concepts.
<table>
<thead>
<tr>
<th>Domain</th>
<th>Specific</th>
<th>Specific for cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applies to systems in general</td>
<td>Topic: - System, - Subsystem, - Unit, - Component</td>
<td>Topic: - Car, - Body work, - Headlight carrier panel, - Lamp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical</th>
<th>Specific for technical manuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applies to digital fragments</td>
<td>Description Type: - Depiction of a component, - Cross section of a component</td>
</tr>
<tr>
<td>Medium Type: - Image, - Jpg</td>
<td>Knowledge Type: - Procedural knowledge, - Knowledge in taking precautions</td>
</tr>
<tr>
<td>Structural Type: - Figure</td>
<td></td>
</tr>
<tr>
<td>Representational Type: - Pictorial representation, - Structured representation, - Chart</td>
<td></td>
</tr>
<tr>
<td>Description Type: - Structural overview</td>
<td></td>
</tr>
<tr>
<td>Description Scope: - Single elaborate</td>
<td></td>
</tr>
<tr>
<td>Knowledge Type: - Procedural knowledge</td>
<td></td>
</tr>
<tr>
<td>Instructional Role: - Learning block, - Illustration, - Example</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design task</th>
<th>Specific for maintenance training design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applies to instructional design</td>
<td>Learning Goal: - Being able to maintain system X, - Skill in performing inspection, - Skill in taking precautions</td>
</tr>
<tr>
<td>Instructional Strategy: - Modeling</td>
<td></td>
</tr>
<tr>
<td>Instructional Activity: - Giving examples</td>
<td></td>
</tr>
<tr>
<td>Instructor Action: - Provide example,</td>
<td></td>
</tr>
<tr>
<td>Learner Action: - Comprehend example</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2 Overview of abstraction levels of ontologies.

The tools used in IMAT make use of the ontologies and are described in the next sections.

3.4.2 The analysis process in IMAT
The highly structured technical manuals allow for automatic segmentation. The document analysis and indexing tool AIDAS supports segmentation and indexing of technical manuals (Anjewierden, 2001; de Hoog et al., 2002a). The indexed fragments can be exported to an object-oriented database. Additionally, AIDAS supports the creation of domain ontologies from overviews or from scratch. Document analysis techniques are combined with the use of ontologies, to subsequently discover the layout structure in a document, the logical structure (the fragments), and to index the logical structure. Segmentation is based on the original document structure: the arrangement of chapters, sections, subsections and paragraphs, resulting in both small fragments (for example an item in a list) and larger fragments (for example a list combined with a picture). One fragment can contain one or more other fragments. AIDAS analyses text and images: sound and video source material is not analyzed and segmented. Image analysis is performed on large schemas of electrical
systems. Segmentation of these images is based on identification of lines, regions, boxes and the like using image processing techniques and OCR for text. The logical analysis of an image attempts to identify semantically meaningful subparts to be re-used together with text fragments about that particular subpart. The stages that are distinguished in AIDAS to analyze and index the technical manuals are described below.

3.4.2.1 Discovering the layout structure: interpreting PDF
The technical manuals in IMAT are available in PDF (Portable Document Format). The approach in IMAT is to use layout information in PDF documents. PDF is a page description language that is powerful in terms of the rendering capabilities it provides and is widely used for document exchange. PDF represents a document as a set of instructions for a rendering engine. When these instructions are processed a bitmap is generated which can be displayed on a graphics device. The instructions in PDF fall into four categories: (1) control instructions which work on the image model and produce no output; (2) text instructions render glyphs; (3) graphics instructions render lines, curves and rectangles etc.; and (4) image instructions render images. The stream of PDF-instructions is converted to a set of text, graphics and image objects. There are about ten different classes of layout objects (line, rectangle, text, image, curve etc.). Each of these classes has about ten features (position, font face, font size, color, line width etc.). This information is used in the next step in the analysis process.

3.4.2.2 Discovering the logical structure
The fragments that need to be stored correspond to elements in the logical document structure (sections, tables, images, items, etc.). To discover the logical structure, the layout structure is interpreted: a set of layout objects is converted to a single logical structure object. To this end, AIDAS reasons about geometry (the positioning of objects on a page); markup (fonts and styles), and sometimes lexical information, for example to recognize that (a) is the start of an item in an itemized list. The approach in AIDAS is based on the idea that layout objects represent not only their layout features, but that these features also contain cues about the role in the logical structure. For example, a text object in a large bold font could play the role of a section title, and a text object containing an asterix could play the role of a bulleted item. AIDAS uses this idea by assigning a set of possible roles to each layout object. This process is performed incrementally until the logical structure is produced.
Figure 3-12 shows a page from a technical manual after AIDAS has analyzed it. At the left, subsequent pages can be browsed; at the right the colored lines correspond to the discovered logical structure. Additionally the textual PDF fragments are reproduced in HTML format.
The buttons at the bottom of the AIDAS tool, from left to right, allow to set a context (for example car repair); set a document (for example a manual of a certain type of car); perform text and image analysis (segmentation and indexing); and to export the indexed fragments to a database. The drop down menu “Storage” allows either clearing the database before storage, or overwriting changed fragments.

Clicking on the “Image analysis” button performs the analysis of the images in the source material (Worring et al., 2001). This process converts the images to SVG, which are stored in this format in the database. The transformation to SVG makes it possible to search the content of an image, something that is not possible with bitmap images. Figure 3-13 shows the results of analyzing a wiring schema.
The lines in Figure 3-13 are colored and indicate what has been recognized, shown in the list of elements in the upper right hand corner. Components in the drawings are identified as boxes with text and (directed) lines associated with text labels. Text is recognized using OCR techniques, and mapped onto concepts in fragment and domain ontologies. By means of colors it is indicated how reliable the OCR is, as this is not 100% certain for documents having a low quality of the source. As no mistakes are allowed in technical diagrams the user must have a clue about the quality of the OCR.

Additional functionalities in the AIDA S tool include a search facility with which the user can have a look at the created fragments, and a functionality to build and maintain ontologies.

### 3.4.2.3 Indexing the fragments
During this stage ontologies are used to index the text and image fragments. The logical structure is analyzed and various techniques are used to define a mapping between the logical and the semantic structure in a document. Each mapping found between a concept in one of the ontologies and the content of a fragment becomes an index on the fragment.

The fragment ontology is used for indexing the fragments on general syntactic properties such as Size for text fragments and Width and Height for images, and on Medium Type, Structural Type, and Representational Type. The attribute values are derived automatically from the source.

Specific domain ontologies serve to index fragments on Topic by matching terms in the title of a fragment to terms in the ontology. Titles in technical manuals are often similar to concepts in the domain ontology. For example when a section describes how a particular
component works or how it can be replaced, the title will at least include the name of the component. AIDA S uses a procedure that takes a title, normalizes it to a set of relevant words and then compares this list of words with the concepts in the domain ontology. Although this process may seem trivial, knowledge about language is required, for example about inflections and conventions used in particular languages (deleting d’ in d’écoulement in French). If the title of a fragment maps onto a concept, then the Topic of the fragment becomes the name of the concept and recursively all sub-fragments will also get this Topic. The idea is that if a section is about some concept, all sub-sections will also be about that concept, at least indirectly. Technical manuals basically contain text and drawings that are often physically separated in the document. A convention that is often used in titles is “General theory of operation (Fig. 14-15A)”. Before this can be properly indexed a number of inferences need to be made. First, a picture fragment must be found which has “14-15A” in the caption and then extract a Topic from the caption. The domain ontology should contain at least abbreviations and the most important synonyms of components for matching titles onto concepts to work well.

The description ontology is used for indexing fragments on semantic properties that regard the Description Type and Scope of the material, by mapping terms in the text to terms in the ontology, while making use of heuristics. A way to derive Description Type is to map the words in the body of a fragment to a set of terms that indicate a high probability of a certain Description Type. For example, a fragment is indexed as a “structural description” if the terms or sets of terms like “component”, “consists of” occur in the body. The Description Scope of a fragment can often be found by considering the relative position of the fragment. A simple rule is that if a chapter has a certain Topic and the first section of the chapter is called “Introduction” or “General” then the section can be indexed on the Topic with a Description Scope of “general”. Another way to derive the Description Scope is by looking at the occurrences of a certain Topic in a larger context. For example, by checking if more fragments about the same Topic occur, the value for the Description Scope attribute can be derived. If for instance two fragments occur, both with the concept “CWARR-radar” (or any of its synonyms or abbreviations) in the title of the fragment, the first is likely to have a general Description Scope, while the second probably has an elaborate Description Scope. This, of course, depends on the document structure.

AIDA S demonstrates that a mapping between layout and logical document structure with support of ontologies allows identifying document fragments and indexing them on syntactic properties, and that a mapping between logical and semantic structure is feasible with the support of ontologies, allowing indexing the fragments on semantic properties.

3.4.3 The storage process in IMAT
In the storage process, fragments are stored in an object-oriented multi-media database (Jasmine ®). Fragments can be stored in the database three ways: by hand; they can be exported from the AIDA S tool, or they can be re-fed into the database from the Authoring Environment Interface (see Section 3.4.4). Fragments exported from the AIDA S tool result from segmentation and contain automatically derived indexes. Fragments re-fed into the database may be enhanced with instructional editing and with additional, manually added indexes. In addition, “bags” of possible relevant fragments for a certain instructional
scenario can be stored. An example of a stored indexed fragment is shown in Figure 3-14. Automatically derived attribute values are printed in italics.

Managing the database consists of two parts: creating contexts for applications and managing fragments. Creating contexts is done by loading ontologies that are relevant for the context (or domain) from which the fragments in the technical manual are extracted. In addition, versions of fragments can be created, so that for instance a fragment containing a mistake can be edited and the modified fragment stored as a draft. In technical domains, manuals can contain mistakes that are discovered during use, but the cycle time between the detection of a mistake and an official update from the manufacturer is often very long. In the mean time it is necessary to store the mistake without changing the "official" content of the technical documentation.

The structure of concepts, attributes and relations in the ontologies is represented in a data-model (Kabel & Wielinga, 1999). A data-model defines a placeholder for the content that is to be stored, which means objects, with attributes, their possible values and relations between objects. Figure 3-15 shows the structure of concepts, attributes and relations is represented in a datamodel.

Figure 3-14 Example of an annotated image fragment.

Figure 3-15 Datamodel.
The rectangles represent objects. The lines represent relations and the italic text indicates the name of the relation. One fragment, textual or pictorial, can contain one or more other fragments, be it textual or pictorial. This design implicates that besides atomic fragments (the most elementary fragments as defined in Chapter 2), larger fragments can be stored, in such a way that larger fragments contain atomic fragments. Each fragment has a description and an instructional description. One (instructional) description may elaborate another, accounting for different levels of granularity.

Ontologies are represented in the Conceptual Modeling Language (CML), (Schreiber et al., 2000). In the fragment ontology, the concept "fragment" is defined as the fundamental placeholder for the indexing of an information element (see Figure 3-16).

![Diagram of concepts in the fragment ontology](image)

All fragments in the database are represented as instances of the concept "fragment" (see Figure 3-17). A fragment can be a text fragment or a pictorial fragment, which inherit attributes from fragment and have additional attributes specific for pictures (for example, start X, start Y, width, height).

```concept
fragment;
DESCRIPTION:
"A fragment is a unit of information";
SUB-TYPE-OF: fragment entity;
ATTRIBUTES:
    source: { string };
    'date created': { string };
    size: { number };
    url: { string };
    medium: { 'fragment medium' };
    'structural type': { 'fragment structural type' };
    'representational type': { 'fragment representational type' };
    descriptions: { description };
    'instructional descriptions': { 'instructional description' };
    parts: { fragment };
END CONCEPT fragment;
```

Figure 3-17 Concept "fragment" with attributes.
The terms in curly brackets refer to the possible values that the attribute can have. The values of attributes Source, Date Created, Size and URL are standard data types and are not derived from ontologies.

For the attributes Medium Type, Structural Type and Representational Type, possible values come from branches of the fragment ontology. For example, the possible values for the Structural Type attribute come from the structural type branch in the fragment ontology. The attributes Descriptions, Instructional Descriptions and Parts represent relations to other objects. The Descriptions attribute represents a relation to an object “description”. In the description ontology a concept description is defined as the placeholder for a description (see Figure 3-5). A “description” object has two attributes, Description Type and Description Scope, the possible values come from the description type and scope branches in the description ontology. Likewise, the Instructional Descriptions attribute represents a relation to an object “instructional description”. In the instruction ontology a concept instructional description is defined as the fundamental placeholder for an instructional description (see Figure 3-6). An “instructional description” object has two attributes, Knowledge Type and Instructional Role, the possible values come from the knowledge type and instructional role branches in the instructional description ontology. The Parts attribute also represents a relation to another object, fragment, allowing a fragment to contain one or more other fragments.

3.4.4 The retrieval process in IMAT

The Authoring Environment Interface (de Hoog et al., 2002b) supports the retrieval process. The tool operates on the database, serving as a mechanism for transferring selected fragments from the database to the “authoring environment” that is used to create instructional material (for instance Microsoft Word). The main functionalities of this tool are to compose queries, browse and view results, add additional indexes, and copy fragments to an authoring environment of choice, which can be any application that supports the copy/paste facility in Microsoft Windows.

In the window in Figure 3-18 the user can specify what is searched for.
At the top of this window, the origin of the fragments can be selected. Fragments either come from technical manuals (the source document fragments), or from user customized content (fragments enhanced with instructional editing), or both. One can also search for fragments coming from one particular manual by specifying its title in “Source title”. The slots in the window in Figure 3-18 represent the search terms that can be used: Topics, Keywords, Description Type, Knowledge Type and Use (Instructional Role). The “+” and “−” push buttons open lists of predefined terms, which are derived from the ontologies, that can be selected by the user. After filling all or only a few of the slots, the user clicks on the “Retrieve” button, after which the search is started. In the “Search log” section, a count is given of the number of fragments matching the query. After the search is over a “Display” button appears, which after being pushed, shows a list of retrieved fragments, each of which can be displayed. Figure 3-19 displays a picture that has been retrieved.
The “tabs” in Figure 3-19 make it possible to display the four main types of fragment (text, pictures, video, sound), but simultaneous retrieval can be done activating the set boxes to the left under “Search”. By clicking on “Set criteria” the user can specify queries, as shown in Figure 3-18. The retrieved fragments are listed at the top, clicking the set boxes displays a fragment on the canvas. For pictures, the tool offers possibilities to scale it, to show thumbnails when many pictures are retrieved, as well as cutting out a part of a picture. When the picture is an analyzed diagram present in SVG in the database, search can also be performed on the content of the picture. Thus searching for a particular component with “text” and “picture” selected would return all fragments (text and pictures) containing this component. Transfer to an authoring environment can be done by using the Clipboard (copy-paste) facility or by storing the fragment as a file.

In the Authoring Environment Interface a retrieved fragment can be given additional indexes using the instructional description ontology. This is shown in Figure 3-20.
In Figure 3-20 a retrieved picture (not shown) is going to be indexed with a particular Knowledge Type: perceptual knowledge of structure with visual identification of parts. In the retrieval process, of course all ontologies can be used. The distinction between various perspectives on content has obvious advantages with respect to retrieval: search and retrieval can be performed based on combinations of physical, domain, and task related properties. For example, an instructional designer may look for a picture of some component, using it for “illustration” in the lesson material. The vocabulary in the instructional descriptions ontology will serve to find all illustrations, while the domain ontology will provide the topic. Additionally, so called instructional material bags with fragments relevant for an instructional scenario can be saved, and annotations concerning organizational memory (user experiences) can be added to the material.

3.4.5 The re-use process in IMAT
The re-use process is supported with the Scenario Tool (Barnard, Kabel, Riemersma, Desmoulins & Grandbastien, 2000; de Hoog et al., 2002b). The Scenario Tool (see Figure 3-21) supports constructing a skeletal lesson in order to retrieve those fragments suitable for a certain training scenario.
In Figure 3-21 the top-level bars from left (Learning Goal) to right (Material Use), are the sequential steps in constructing a skeletal lesson, which in turn follow the instructional design ontology. First a Learning Goal is selected and the tool automatically selects the fitting fragment description (for example “definition; location of component”). Next the user selects a Knowledge Type (for instance “declarative knowledge, knowledge of structure, composition”) and an Instructional Strategy. The options are constrained by previous choices. After choosing a particular Instructional Strategy the remaining columns are derived automatically. The content of the column Instructional Activities contains the skeletal lesson consisting of a sequence of things that have to be done. At the top is the activity that has to be carried out first (for example “setting goals”), and at the bottom is the one that has to be carried out last (for example “explanation of next lessons”). The column I/L Actions specifies which kinds of activities are expected from instructors and learners during a phase of the skeletal lesson. The final column Material Use indicates what role the material used plays in the related phase. For example, during the Presentation activity the material used plays mainly a definitional role. After the skeletal lesson has been identified it can be saved to file as an instructional material bag in the Authoring Environments Interface. This “bag” can be opened and then it shows the structure of the skeletal lesson developed using the Scenario Tool. The user can now employ the characterizations from the skeletal lesson, for example Knowledge Type and Material Use, together with a Topic or keyword to guide the retrieval, that is setting search criteria just as in Figure 3-18. The retrieved and selected fragments can now be stored as files in the relevant sections of the bag, ready to be copied to an authoring environment.

In the re-use process the benefit of discriminating between physical, domain, and task-related aspects of fragments is found in the shift of the instructional designer’s focus from the retrieval task to the goal task of instructional design. By specifying the task-related aspects and a topic, an instructional designer is provided with a set of potentially useful fragments for some learning goal and is supported in sequencing the fragments.

3.4.6 Evaluation of the ontologies in IMAT
A formative evaluation (Kabel, Riemersma & Wielinga, 2001) was performed at the three IMAT user sites: RNLAF (Royal Netherlands Airforce), AFPA (Association Nationale pour la
Formation Professionnelle des Adultes) and ETRA. At the RNLAF the focus is to learn the learners to reason about a (weapon-)system; the most important learning goal at AFPA is geared towards corrective maintenance (of a car); at ETRA all learning goals would be suitable, but the instructional strategies mainly are exploration and coaching (to learn solving problems with traffic control systems). The goal of the evaluation was to provide directions for improvement of three ontologies, description, instructional description and instructional design, in terms of completeness and correctness, specificity level and comprehensibility. To evaluate the ontologies, the three user partners composed lesson material from their database with indexed fragments using the ontologies in the Scenario Tool and the Authoring Environment Interface, and filled out a questionnaire.

3.4.6.1 Completeness and correctness

Users were asked if they would add or remove top-level concepts. The resulting comments are listed below.

- Despite the inclusion of an extensive corrective maintenance branch in all relevant parts of the ontologies, users missed “problem solving” to deal with problems that are not covered by the standard fault-tree or dealing with unexpected situations (for example during exercises). They would like to have this represented in a Learning Goal, Knowledge Type, Instructional Strategy (“problem-based and/or case-based learning”) and Instructional Role (“problem” and “case”).

- The difference between installation and configuration, mainly in those systems with embedded software, should be explicit. The first one would relate to physical installation of the equipment and its components, the second one (configuration) to the preparation of the system to carry out a given functionality. Users would add a Learning Goal “being able to configure a system”, and add Description Types “overview of installation task”, “overview of configuration task” and “overview of programming task”, and lastly they would add a Knowledge Type “knowing how to configure a system”.

- The Learning Goal “being able to perform inspection tasks” is seen as a sub-goal of both preventive and corrective maintenance.

- To Knowledge Types, users would add “knowledge about responsibilities and competencies of different staff members/organizational units” and they would remove “attitude with respect to organizational goals” and “attitude with respect to teamwork”.

- Users would like to be able to define their own Instructional Strategies, for example “collaborative learning”. They would also like to choose a combination of two instructional strategies instead of just one.

In general, improvements with respect to correctness and completeness would lie in extending the ontology with a problem solving branch; differentiating between installation and configuration of a system; and in extending the number of Instructional Strategies.

3.4.6.2 Specificity level

To evaluate the specificity level of concepts, users were asked if they would split or merge detailed concepts. The idea behind this is that when users want to split a single concept into two or more concepts it is regarded as being too large, and if they want to merge two or more
concepts into one, it is regarded as being too small. The comments on this question are listed below.

- With respect to Learning Goal, the difference between “structural description of (sub) components” and “structural layout of (sub)components” is not clear, nor is the difference between “goals of the function” and “function”. According to the users, the concepts are too detailed and should be merged.
- The specificity level of Knowledge Type concepts is generally regarded to be too detailed. The detailed Knowledge Type concepts are suitable for an expert (a learning psychologist) but not for instructors as in their own education they do not learn much more detail than the distinction between knowledge, skills and attitudes.
- The list of actions is considered a form of guidance or a reminder for the instructors (since it is automatically generated). It may be very detailed for many instructors but it is valuable as a checklist.
- Some distinctions between concepts in Instructional Role require differentiating definitions, for example “feedback” and “answer”; “hint”, “guideline” and “reminder”.

In general, the specificity level of concepts is sometimes considered to be too detailed. Either the differences between concepts have to be made clear or concepts have to be merged. Sometimes not all detailed concepts are used, but they are useful serving as a checklist.

3.4.6.3 Comprehensibility
To evaluate if users comprehend the ontologies, they were asked if they understand the meaning of the detailed concepts. Comments are listed below.

- Learning Goal concepts are regarded to be complex and formulated in theoretical terms, for example learning the learner to recognize (sub)components in larger (sub)components or units.
- Knowledge Type concepts too are found quite theoretical and abstract.
- With respect to Instructional Strategy concepts users say they are unfamiliar with the term “exploration”, and that “articulation” is a non-standard term. In general they need more explanation to understand these strategies.

General improvements lie in reducing the complexity of mainly Learning Goal and Knowledge Type concepts, and in a better explanation of Instructional Strategy concepts. The evaluation aimed to highlight directions for improvement. The ontologies that are at the core of the Scenario Tool could be improved in terms of completeness and correctness, specificity level and comprehensibility. However, the evaluation also showed that users were able to perform the task of creating instructional material with the Scenario Tool. Beside this evaluation, the tools were used in the daily practice of the people who created training material and give maintenance courses. When asked, users confirm that the tools have added value in their work.
3.5 Conclusions

The IMAT approach to re-use technical documentation for training purposes is based on the idea of dispensing with the traditional sequential structure of documents and to replace it with a much richer structure that allows retrieval and re-use of material in a variety of instructional contexts. In essence, the technical documents are converted into a large collection of information fragments stored and indexed in a database. Each information fragment is described by a number of attributes to which values are assigned from a collection of ontologies. These descriptions concern the subject matter in the application domain; syntactic properties of the material such as the medium type, structural and representational properties; semantic properties of the material such as the description type and scope; and semantic, task-related properties of the material such as the type of knowledge and the role of the material in instruction. These properties provide flexible access to fragments. The task of developing instructional material is not always well understood and the diversity of learning goals and learning tasks requires flexibility in instructional material design. A rich set of ontologies appears to be a crucial means of achieving this flexibility of (re-)use of source materials.

A “proof of concept” of using ontologies in the material handling processes was demonstrated in the IMAT project. Ontologies have a number of advantages that are an improvement with respect to other methods: they solve a number of terminological problems, they support the (semi-) automated segmentation of source documents, they enable a rich, multi-faceted description of document fragments, and their modular structure allows retrieval from multiple viewpoints. Retrieval is even more extensively supported when the context of re-use is taken into account: instructional designers are able to create instructional material from a collection of information fragments, fully based on task specifications.

Having shown the feasibility of using ontologies for the processes, the next question to be addressed regards the effect of using ontologies for indexing with respect to other methods. Semantic annotations are important to provide meaningful information to retrievers, but manually added annotations may contain inconsistencies, which would undermine the power of retrieval based on these annotations. The next chapter investigates whether using ontologies leads to higher consistency between annotations made by different indexers when compared to a bag of keywords.