Feature grammar systems. Incremental maintenance of indexes to digital media warehouses

Windhouwer, M.A.

Publication date
2003

Chapter 1

Digital Media Warehouses

> *Yama said, "Surely knowledge should be free to everyone, since all knowledge is the gift of the Preservers."*
>
> *"Ah, but if it was freed," Kun Nurbo said, "who would look after it? Knowledge is a delicate thing, easily destroyed or lost, and each part of the knowledge we look after is potentially dependent upon every other part. I could open the library to all tomorrow, if I was so minded, but I will not. You could wander the stacks for a dozen years, Yama, and never find what you are looking for. I can lay my hand on the place where the answer may lie in a few hours, but only because I have spent much of my life studying the way in which the books and files and records are catalogued. The organization of knowledge is just as important as knowledge itself, and we are responsible for the preservation of that organization."*
>
> Paul J. McAuley – *Ancients of Days*

Encouraged by the low price of digitizing methods (*e. g.* digital cameras, scanners) and storage capacity (*e. g.* DVDs) collections of media objects are quickly becoming popular. Public services like libraries and museums digitize their collections and make parts of it available to the public. Likewise, the public digitizes private information, *e. g.* holiday pictures, and shares it on the World Wide Web (WWW). Vast collections of digital media are thus constructed in a relatively easy manner.

The management of these media objects encompasses many more aspects than just populating the *digital media warehouse* (DMW). First, there is the retention issue of the digital content; will the digital image be accessible in 25 years from now? Dedicated database [Sub97] and file systems [Bos99] have been developed to handle the input, storage and output of media streams. Second, security issues play a role: who is allowed to retrieve the data and should the data be encrypted? Third, the mass of information in a DMW stresses our capability to find relevant information: how to retrieve all images related to, for example, jazz music? The next section will delve

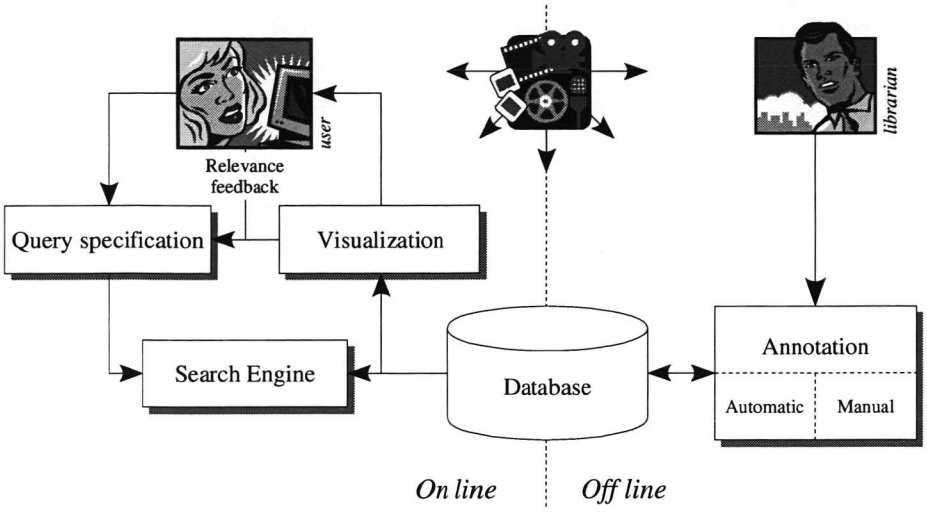

Figure 1.1: Multimedia information retrieval system

deeper into this last issue, because the contribution of this thesis lies within this grand challenge.

1.1 Multimedia Information Retrieval

A known and limited subset of a warehouse can be accessed by browsing: the common practice of every day life on the WWW. However, if this subset is unknown, identifying relevant media objects in the vast collection poses a major problem.

Identifying relevant media objects is studied in the area of *multimedia information retrieval*. This research community is multidisciplinary and thus attracts scientist from various disciplines, *e. g.* computer vision, artificial intelligence, natural language processing and database technology. These disciplines play their specific role in the subsystems of the generic multimedia information retrieval system sketched in Figure 1.1 (based on [Del99]). A small walk through this generic system will clarify the information flow between and the individual roles of the various subsystems.

The user, *i. e.* the person in the left part of the figure, starts a session with the system to resolve a query intention, for example: find a portrait of the jazz trumpeter Chet Baker. Query specification tools offer assistance in translating the query intentions into query clues understood by the system. For example: *query-by-sketch* (QbS) [DP97, JFS95] or *query-by-text* (QbT) [dJGHN00, KKK+91, OS95] are well-known paradigms being used. In QbS a global impression of the image has to be drawn. Keywords or phrases, like "Chet Baker", "jazz" or "trumpet", form the clues used by the

QbT paradigm.

These query clues are subsequently translated by the search engine into database queries. The type of the information stored in the database, and thus these translations, is as diverse as the query specification paradigms. For example: the QbS paradigm maps the clues on numerical feature vectors containing information about color, texture and shapes. The keywords and phrases from the QbT paradigm may map on entries in an ontology [SDWW01], controlled vocabulary or textual annotations. This mapping from query clues to the information stored forms the basis to find matching media objects. When the mapping is also used for the ranking of matching objects the search engine needs a notion of similarity: how similar are two objects in the space induced by the mapping? Using this distance metric the objects can be ranked from the best to the worst match [Fal96].

The database executes the query specification to match and rank the media objects. A visualization tool presents these query results for further inspection to the user. Also for this part of the generic system many paradigms are available: the results may be shown as clusters in a multidimensional space [vLdLW00] or the user can browse through them [CL96]. Other senses than the user's eyes may also be used to present the query results, *e. g.* when the media type is audio or a score the musical theme is played [MB01].

In most cases the user will have to refine the query to zoom in on the relevant set of multimedia objects [MM99, VWS01]. This relies on a better understanding by the user of the database content thus allowing a better formulation of the information need. Query refinement is supported by a relevance feedback mechanism [CMOY96, RHM98, RHOM98, RTG98, Roc71], which allows the user to indicate the positive and negative relevance of the objects retrieved. These indications are used by the system to adjust the query clues better to the user's query intention. Such a mechanism connects the visualization tool to the query specification tool and creates an interactive loop. The hypothesis is that when the user terminates the loop he or she will have found the media objects in the collection with a best match to the query intention.

In every system part the original media objects play a role. These media objects can be either stored directly in the database, or reside on a different storage medium, *e. g.* the file servers of the WWW. The information exchanged between the various subsystems will seldom contain the raw media objects. Instead database keys, filenames or *Uniform Resource Identifiers* (URIs) [BLFIM98] are passed along.

The information about the collection of media objects is produced by the annotation subsystem. Part of this system handles the interaction with the librarian, *i. e.* the person in the right part of Figure 1.1. This librarian uses his domain knowledge and standard conventions, *e. g.* in the vain of the traditional *Anglo-American Cataloguing Rules* (AACR2R) [GW98], to annotate the media objects. These annotations range from content-independent [DCM01], *e. g.* this image was added to the collection at July 1, 1998, to content-descriptive data [ISO01], *e. g.* this image is a portrait of Chet Baker [Gro94]. Apart from a manual part the annotation system also has an

automatic part. In the automatic part the system uses algorithms and additional information sources, like an ontology or a thesaurus, to automatically extract additional information. Interaction between the two parts may be used to complete and verify the annotation, *e. g.* automatic extracted concepts may be approved by the librarian.

The database functions as a persistent buffer between the off line produced annotation information and the on line use of this information to answer queries. This database is managed by a *Database Management System* (DBMS). A DBMS offers not only persistent storage of the data, but also other functionality needed by a DMW. For example, to assure a consistent representation and to control mixed access, but also, one of the major research themes in database technology, query optimization. The search engine will profit from the last one in its search for matching media objects.

As the main focus of this thesis lies within the idea of automatic annotation extraction extraction the coming section will further describe the role of this subsystem.

1.2 Annotations

As discussed in the walk through and shown in Figure 1.1 the annotation information is produced manually and/or automatically extracted. However, with the increasing size of media collections manual annotation of all media objects becomes unfeasible. Likewise, when the collection is unconstrained, *i. e.* contains media objects from various domains, manual annotation of the objects can never meet all possible query intentions. Even for domain and size restricted collections manual annotation remains hard, due to the fact that annotations tend to be subjective, *i. e.* they describe the personal perception of the librarian. These aspects increase the importance of the automatic part of the annotation subsystem.

1.2.1 The Semantic Gap

The holy grail for automatic annotation is to take over the content-descriptive part of the manual burden. To realize this, the *semantic gap* between raw sensor data and "real world" concepts has to be bridged. For visual data this gap is defined as follows [SWS+00]:

> *The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.*

This definition may be generalized to raw sensor data in general without loss of validity.

The semantic gap is visualized in Figure 1.2. The user with all his/her general knowledge will have many associations with this photo. These associations range from generic to specific ones, *e. g.* from "this is a portrait" to "this is a portrait of

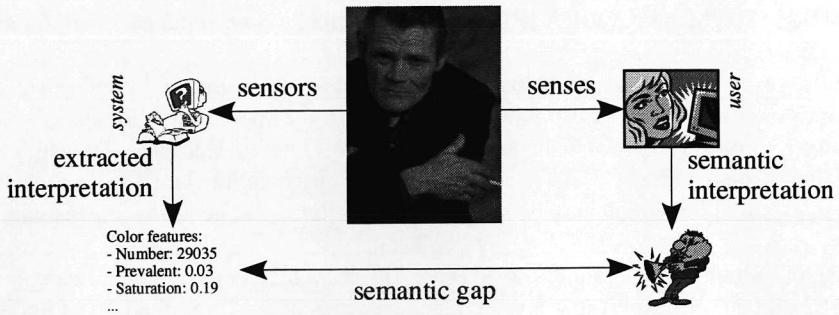

Figure 1.2: The semantic gap visualized

the jazz trumpeter Chet Baker". Ideally, in the case where there is no semantic gap, the computer system can extract the same information from a digital version of this photo. Algorithms to classify this image as a photo and to detect the frontal face are available. Combining this basic information the validity of the generic semantic concept portrait can be induced. The validity of more specific concepts often depends on the availability of more contextual knowledge about the media object.

However, the semantic gap is still not filled and may never be. One of the reasons is the role of ambiguity. The more abstract a concept becomes the more subjective, due to *e. g.* cultural context-sensitivity, interpretations are possible. In [Eak96] the authors distinguish three image content levels:

level 1 primitive features: color, texture, shape;

level 2 derived (or logical) features: contains objects of a given type or contains individual objects;

level 3 abstract attributes: named events or types of activities, or emotional or religious significance.

The higher the level the more subjective, and thus ambiguous, annotations become. State of the art annotation extraction algorithms reach level 2. Level 3 algorithms are only possible for clearly defined and distinguishable (narrow) domains. To provide enough support for an attack on the third level the annotation subsystem will need specialized constructs to handle this ambiguity, *e. g.* using probabilistic reasoning.

1.2.2 Annotation Extraction Algorithms

The predominant approach to try and bridge the semantic gap is the translation of the raw data into low-level features, which are subsequently mapped into high-level, *i. e.* semantic meaningful, concepts. This approach is reflected in frameworks like

ADMIRE [Vel98] and COBRA [PJ00] and the *compositional semantics* method used in [CDP99].

Low-level features (level 1) are directly extracted from the raw media data and relate to one or more feature domains embedded in the specific media type [Del99]. For images color, texture and shape features are well known examples. The choice of domains gets even bigger when several media types are combined into one multimedia object, *e. g.* a video which may be seen as a, time related, sequence of images with an audio track.

Rules, which may be implicit, map these low-level features into semantic concepts (level 2 and 3). An expert may hard-code these rules, *e. g.* a combination of boolean predicates, or they may be learned by a machine learning algorithm [Mit97]. Such an algorithm may result in human readable rules, as is the case with decision rules [Qui93], or the rules may be hidden inside a blackbox, *e. g.* in the case of a neural network [Fau94].

In fact there is a wealth of research on extraction algorithms for both features and concepts. When a subset of them are used to annotate a collection of media objects they depend on each other to create a coherent annotation.

1.2.3 Annotation Extraction Dependencies

Annotations of the example image of Chet Baker may be extracted by using these mappings (illustrated in Figure 1.3):

1. the image is classified as a photo: feature values, *e. g.* the number of colors and the saturation of these colors, are used in a boolean rule, which determines if the image is a photo or not [ASF97];

2. the photo contains a human face: the group of skin colors in the *c1c2c3* color space, are used to find skin areas and these areas form the input to a neural network which determines the presence of a human face in the photo [GAS00].

This example shows that concepts do not only depend on features, they may also depend on each other. In this example the face detection presupposes that the image is classified as a photo. This is a different kind of dependency. The dependency between feature and concept extraction is based on a direct output/input relation: the output of the feature detector is input for the photo decision rule. This type of dependencies is called an *output/input dependency*. However, the dependency between the two concepts is based on context: the photo concept functions as a contextual filter for the face concept.

This *context dependency* can be hardcoded as an output/input dependency. Unfortunately this will harm the generality of the face detector: it can not be reused in a different context, where there is no photo pre-filter. Context dependency is a design decision or domain restriction and is not enforced by the extraction algorithm. In this specific case the decision to use the photo classifier as a pre-filter is made because

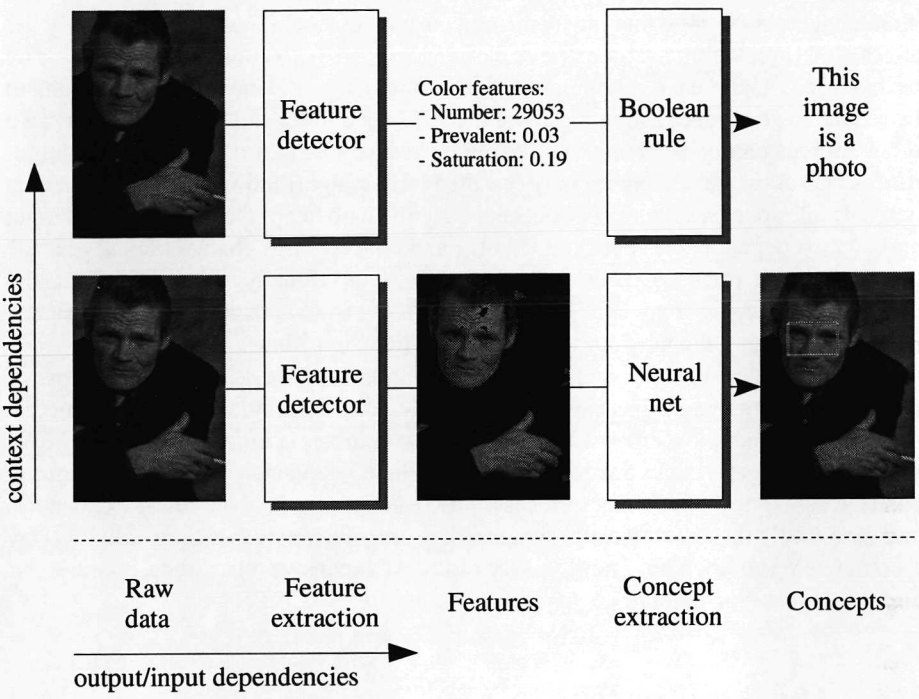

Figure 1.3: Automatic information extraction steps

the face detector is expensive, while the photo classifier is cheap. By using the photo classifier as a pre-filter only images with a high chance on the presence of a face will be passed on to the expensive face detector. Due to the explicit handling of this context dependency the face detector stays generic in nature and is able to be reused in a different context, *e. g.* black and white images.

The subsystem which controls the automatic information extraction has to take care of these dependencies and use them to call the algorithms, evaluate the rules and run the machine learning algorithms to produce the features and concepts to be stored in the database.

1.2.4 Annotation Maintenance

Complicating the task of the annotation subsystem further, supporting multimedia information retrieval in a non-static environment, like the WWW, involves the maintenance of the annotations, features and concepts, stored in the database so they reflect the current status in this evolving environment.

There are several possible sources of change leading to the need of annotation

maintenance. Assuming that the media objects are not stored in the database, only the annotations are, the first source is an *external* one. The media objects themselves may be modified. Upon each modification the automatic (and manual) annotation has to be redone to guarantee that the database contains the correct and up-to-date data. Two other sources can be seen as *internal* to the system: changes in the extraction algorithms and in the dependencies between them. If an algorithm is improved (or a bug is fixed), the specific features or concepts have to be updated. Due to the output/input and context dependencies between features and concepts this change may trigger the need for reruns of many other extraction algorithms. Finally, the output/input and context dependencies may change. The addition or removal of a context dependency may, again, trigger the need for reruns of extraction algorithms.

When the dependencies and algorithms are embedded in a, hand crafted, special purpose program there is basically one option: rework the program and do a complete rerun of the annotation process for the affected multimedia objects. However, when (at least) the dependencies are described in a declarative manner, a supervisor program can take care of the maintenance process. Such a supervisor analyzes the dependencies and reruns only the extraction algorithms which are affected by the change. In this way a complete rerun, including unnecessary reruns of expensive algorithms, is prevented and the database is maintained incrementally.

1.3 The Acoi System

Although incremental maintenance of multimedia annotations has been identified as a key research topic [SK98], there has been little actual research to solve this problem and no satisfactory solution exists yet. This thesis describes the Acoi system architecture and its reference implementation, which provides a sound framework for the automatic part of the annotation subsystem, including incremental maintenance.

1.3.1 A Grammar-based Approach

Formal language theory forms the foundation of this framework. Its choice was based on the observation that proper management of annotations all involve context:

the semantic gap the more specific a concept, the more structural contextual knowledge is needed for validation (see Section 1.2.1);

disambiguation the more abstract a concept, the more user specific contextual knowledge is needed to disambiguate it (see Section 1.2.1);

contextual dependency to promote reuse of detectors, context dependencies should be explicitly handled (see Section 1.2.3);

incremental maintenance exact knowledge of the origins, *i. e.* the context, of an annotation is needed to localize the impact of internal or external changes and thus enable *incremental* maintenance (see Section 1.2.4).

The Acoi system would thus benefit from a dependency description or processing model which covers context knowledge for both annotations and extraction algorithms. Traversing the dependency description a path from the start of the extraction process to the actual extraction of a specific annotation can be maintained. A set of annotation paths can easily be described by a tree. Sets of valid trees, *i. e.* valid annotation paths, are naturally modeled by grammars. Grammars form a context preserving basis for a dependency description. However, the context descriptions should be underspecified enough to keep algorithms generic and enable, and even promote, reuse. The theoretical and practical implications of this intuition is investigated in this thesis.

1.3.2 System Architecture

Detailed descriptions of the Acoi system components, shown in Figure 1.4, and their relationships form the core of the thesis.

Chapter 2 starts with a description of the Acoi system foundation: the *feature grammar systems*. This foundation is based on a careful embedding of extraction algorithms into formal language theory and to formally describe both types of information extraction dependencies.

The next chapter introduces a non-mathematical notation for feature grammar systems: the *feature grammar language*. This language supports the core of a feature grammar system. Based on earlier experience extensions are added to conveniently support the various forms of feature and concept extraction.

In Chapter 4, the *Feature Detector Engine* (FDE) uses the execution semantics of feature grammar systems to produce the annotations. This involves traversing the dependencies described and execution of its associated extraction algorithms. The core is supplied by a practical algorithm taken from natural language research and compiler technology and adapted to handle the specific needs of feature grammar systems.

The impact of the system on the database is discussed in Chapter 5. The engine delivers its data, *i. e.* annotations and their complete context, in a format related to the semantics of the feature grammar language. This format is generic and can be mapped to the requirements of any DBMS. In this chapter a DBMS specific mapping for the Monet back-end and related optimization issues are discussed.

The *Feature Detector Scheduler* (FDS), described in Chapter 6, analyzes the dependencies, *i. e.* the possible contexts, in a specific feature grammar to localize the effect of changes in source data, algorithms or dependencies. When the parts affected are identified, the scheduler triggers incremental maintenance runs of the engine, which result in the propagation of changes to the database.

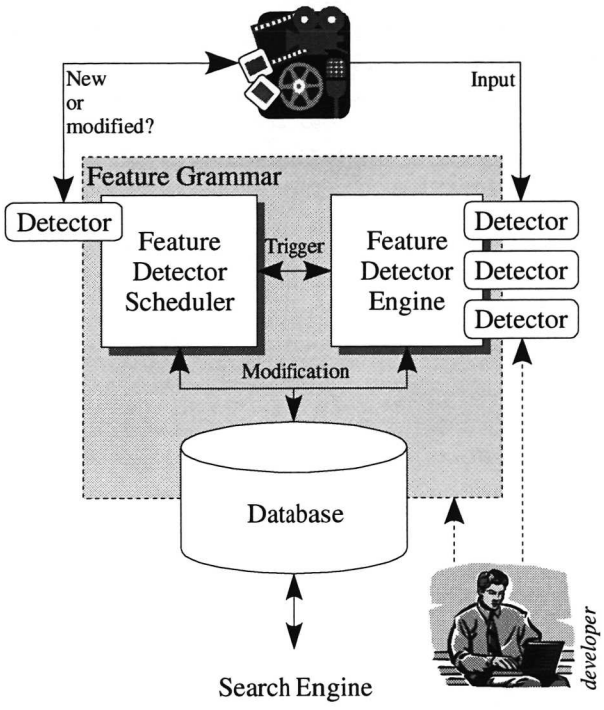

Figure 1.4: Acoi system architecture

1.3.3 Case Studies

Various real world applications have been used to identify and evaluate functional, performance and capacity requirements for the Acoi system architecture. The case studies will be entirely exposed in Chapter 7. But throughout the thesis they will also, just like the reference implementation, function as running examples to illustrate how specific requirements are met by the system architecture. Therefor the succeeding subsections will shortly introduce the case studies.

The WWW Multimedia Search Engine

The WWW is probably the largest unconstrained collection of multimedia objects available. Search engines have extracted text-based entry points to function as road-signs to browse this vast collection. With the growing popularity of the web the search and retrieval of other media types is getting more attention, *e. g.* both AltaVista [Alt01] and Google [Goo01] are offering some support for retrieval of multimedia objects. However, this support is still based on text, *e. g.* keywords extracted from either the

URL of the object or from the web page it appears on. Content- or concept-based retrieval play only a significant role in research prototypes, like WebSeer [FSA96], WebSEEk [SC96] and ImageScape [Lew00]. These prototypes allow the retrieval of images on the basis of a limited set of, hardwired, concepts, *e. g.* faces or landscape elements.

The Acoi system architecture is used to build and maintain a multimedia search engine's index. With advances in computer vision and sensor informatics the number of automatic extractable features and concepts will gradually increase. Due to the system's ability to maintain its index incrementally (prototypes of) new features or concept extraction algorithms are easily added. This ability also makes it well suited to adapt to the dynamic behavior of the Internet, *i. e.* the index is continually updated instead of completely recreated.

The basis is a simple model of the web: web objects and their links. This model is then evolutionary enhanced with content-based feature and concept extraction algorithms.

The Australian Open Search Engine

This Australian Open case study also involves the maintenance of a search engine's index. But in this case the domain is restricted to the Australian Open tennis tournament. In the WWW case study the model contains multimedia objects and generic relations. This limited model makes it possible to extract only very generic features and concepts, *e. g.* this video contains 25 shots. However, in this case study the system also contains conceptual information and, combined with domain knowledge, more specific feature and concept extraction can be realized, *e. g.* this video of a tennis match between Monica Seles and Jennifer Capriati contains 25 shots of which 20 show the tennis court.

The prime benefit of the Australian Open case study is to test the flexibility and open character of the system architecture. The Acoi system is embedded in a larger application and has to interact with separate systems, which handle the conceptual data or function as distributed extraction algorithms.

Rijksmuseum Presentation Generation

The Rijksmuseum in Amsterdam, like many other museums, makes part of its collection available in digital format[1]. This gives the public alternative and interactive ways to browse the collection. The database underlying this interactive system contains manual annotations of the museum pieces.

The underlying database is semistructured in nature, *i. e.* the annotation is not always complete. The Acoi system is used, in this case, as a style database. If the annotator did not specify the style period of a painting the system tries to infer the

[1] `www.rijksmuseum.nl`

correct style using the dependency description and associated extractors. The thus automatically augmented annotation may help in several ways. It may help the annotator in completing the annotation by providing useful hints. Furthermore, it may allow the museum visitor to retrieve possible matches.

The features and concepts extracted may also be used to influence and optimize the layout of the hypermedia presentation generated to browse a query result.

1.4 Discussion

This introductory chapter surveyed the domain of digital media warehouses. A number of research challenges exist within this domain and are the focus of attention for a multidisciplinary research community. The research described in this thesis is dedicated to the problem of automatic extraction and (incremental) maintenance of multimedia annotations. To retain enough contextual knowledge a grammar-based approach is taken, which grounds the approach in a well-studied field of computer science. The subsequent chapters start with laying the formal basis and work towards a practical solution to the problem. Chapter 7 will showcase the solution in the form of the evaluation of several case studies in the problem domain, and may thus be of main interest to practical oriented readers.