

# **AIDA – Experiences in Compensating the Mutual Weaknesses of Knowledge-based and Object-oriented Development in a Complex Dental Planning Domain**

E. Finkeissen<sup>1</sup>, R. Weber<sup>1</sup>, S. Haßfeld<sup>2</sup>, U. Koke<sup>3</sup>, T. Wetter<sup>1</sup>

<sup>1</sup>Institute for Medical Biometry and Informatics, Department of Medical Informatics,  
University of Heidelberg, Germany

<sup>2</sup>Hospital for Oral and Maxillo-Facial Surgery, University of Heidelberg, Germany

<sup>3</sup>Hospital for Dentistry, Department of Prosthodontics, University of Heidelberg, Germany

## **Correspondence Address:**

Ekkehard Finkeissen

Im Neuenheimer Feld 400

69120 Heidelberg, Germany

Ekkehard\_Finkeissen@med.uni-heidelberg.de

## **Abstract of the project**

**Objective:** Dentistry is a discipline with two properties that pose a serious challenge to knowledge based decision support: (a) It has to integrate six sub-disciplines ranging from conservative measures to invasive disciplines, such as implantology; (b) A plan may have to cover a complex treatment often lasting one year or more. It is the aim of the AIDA-project<sup>1</sup> to set up a planning strategy that is suited to incorporate all dental peculiarities in one methodology.

**Methods:** Generic tasks that can be assigned to individual persons involved in dental treatment have been designed with the help of KADS. They have been integrated into a planning super-structure for the planning of all dental solution alternatives that can principally be applied on the basis of the given patient status.

**Results:** Besides an evaluation of the implemented planning system itself, it has been evaluated how well the development is supported by (1) knowledge-engineering methods and (2) object-oriented methods.

**Conclusion:** Common knowledge-based tools are not powerful enough for the planning of complex dental constructions. Therefore, a solution combining object-oriented and knowledge-based methods is proposed.

**MESH-keywords:** Dentistry, Decision Making, Theoretical Models, Knowledge, Decision Support Techniques

---

<sup>1</sup> Artificial Intelligent Dental Agents; <http://aida.uni-hd.de>.

## 1. PROBLEM DESCRIPTION AND METHODS

The final aim of dentistry is to provide an 'impeccable stomatognathic system'. From the functional point of view the patient has to be enabled to chew his food adequately. Agreeable esthetics of the set of teeth is necessary for the patient's balanced mental situation. If the current situation of a patient is not satisfactory according to these criteria, a treatment may try to recover dental health.

Planning, in anticipation of decisions [1], is supposed to improve the quality of the treatment by sketching the possible treatment alternatives and their expected properties. Herein, financial aspects can be seen as an additional constraint for the choice among medically appropriate treatment alternatives [2].

Dentistry is a field of medical decision-making where it is obvious that the scope of planning and justification is constrained by the fact that the patient can formulate and influence the decision much more than in many other medical fields. He or she may, for instance, prefer certain materials to be used for reasons of esthetics. Or the patient may prefer a certain type of denture for reasons of practicability, finance, hygiene, etc.

Therefore, the scope of objective decision support ends with identifying alternative solutions, each of which has to be in accordance with up-to-date medical treatment quality. For individual therapy decisions, transparency of treatment alternatives, their outcomes and personal risks for dentist and patient is necessary [1,3-6]. Among these alternatives patient and dentist have to decide according to preferences that do not necessarily have to be justified.

Due to a lack of a unified planning methodology in dentistry there is not yet a certified procedure for the derivation of solutions on the basis of a given patient status. This makes an objective comparison of dental treatments difficult, if not impossible.

This is why the AIDA-project was set up to build a unified decision methodology for dentistry. To guarantee a stringent formulation of the planning rules, the approach had to be designed in a top-down manner. To include as much knowledge as possible into the methodology, it is necessary to expose the coming methodology to the criticism of peer scientists and practitioners. So, later on an authoring tool should be built to support the discussion between the persons involved in dental treatment.

The complexity in dental planning results from the interdependencies between the different dental sub-domains (conservative dentistry, prosthetics, periodontics, pediatric dentistry, surgery,

orthopedics, etc.). For that reason it was decided within the AIDA-project to plan and build corresponding modules for grouping problem areas into sub-domains. Here, the task of generating all solution alternatives is to be automated by the whole system.

In principle, generic dental tasks have been designed, that can be assigned to each person involved in dental treatment. For the ease of access and maintenance, the respective planning tasks are bundled into agents. Since a machine cannot make the final decision about the acceptance of a partial plan, the responsibility has to be assigned to a specific person for verification.

The overall architecture of agents for the sub-disciplines and their embedded knowledge has been designed using the KADS methodology. The prosthetic agent has been implemented using a combination of KADS and UML.<sup>2</sup> In this article, our attention will not focus on sophisticated agent architecture but on the current degree of methodological support for the top-down modeling of dental planning. Therefore, the features of the major paradigms of knowledge-based systems and object-oriented modeling will be compared for the domain of dentistry.

On the basis of both theoretical and practical prerequisites, a general recommendation is given for the design of dental knowledge representation. Later, persistent methodological limitations will be described. The advantages and disadvantages of knowledge-based and object-oriented approaches are outlined with concrete reference to dentistry.

## **1.1 Knowledge-based Modeling Approaches**

KADS is the major methodology for authoring knowledge-based systems from reusable components [7-10]: KADS covers the process for the development of knowledge-based systems to a design model. In addition to the development of a knowledge-based system, (Common)KADS offers exhaustive possibilities for the analysis of knowledge-intensive tasks and processes. Furthermore, KADS supports the investigation of chances and bottlenecks of development, distribution and application of knowledge resources in organizations and thus it is supposed to be a tool for integrated knowledge management. As part of it, the reuse of generic

---

<sup>2</sup> The validation could successfully be carried out on the basis of constructed and real cases. Herein, the appropriateness of the AIDA-approach could be shown. A major problem appeared to be the existing language, which has been found to not to be consistent.

model components is possible on the basis of a formal scheme for the support of the top-down knowledge acquisition and the process of differentiation of simple to complex models [8].

On this basis, KADS seems to be the favorite approach for modeling dental decision structures. But unfortunately it was recognized in the AIDA-project that the primary problem for the development of a completely new dental domain model is not the reusability because replications (e.g., for 32 teeth) can appropriately be addressed by an object-oriented analysis and the resulting class model.

The main task during fundamental structuring is to derive a comprehensive decision strategy, which can be discussed and communicated among all groups involved in the process from elaboration to implementation. But the whole approach can only be assessed on the basis of concrete treatment plans provided by the methodology. This requires operational code emerging from the methodology. But generating code is not explicitly supported by KADS (Figure 1). In addition, transformation of the design model into a running system also is not a major subject of the KADS methodology.

One might, therefore, consider other methods for knowledge modeling. Knowledge-based development tools such as D3 [11] provide support for rapid prototyping, but they are very limited in their variety of methodology. These solutions are suitable mainly for the development of diagnostic and tutor systems, as the solutions have to be chosen out of a set of predefined alternatives. For complex construction tasks, as in dentistry, they are hardly appropriate. Moreover, the user is bound to the given solution paradigm, which constrains his abilities and might contradict the demand to comply with the complexity of the problem. Additional useful features such as the automatic generation of a user interface cannot compensate this fundamental lack.

The development environment Protégé-II [12] is supposed to facilitate the reuse of knowledge. The libraries of reusable role-limiting problem-solving methods address this. They define – in domain-independent terms – the manner in which propositional domain knowledge may be used to solve application tasks [13,14]. This approach is useful for the representation of a domain model, but it did not seem to be appropriate for the prior conceptual analysis of a complex dental domain and the development of an underlying structure for decision-making.

The latest version of Protégé-2000 [15] supports the incremental development of models. It provides interfaces for the export of the knowledge base and the derivation of a UML (Unified Modeling Language) implementation model. The main aspects of Protégé-2000 are the

formulation of an ontology and the derivation of a respective knowledge-acquisition tool. However, it is still not possible to develop the decision-support system within one homogeneous environment with the option of reverse engineering and roundtrip engineering, respectively.

The planning systems Plakon and Konwerk [16,17] have been built to support the formal representation of knowledge-based configuration tasks in technical domains. They are based on the projects TEX-K and PROKON dealing with the development of suitable representation language, tailored to the needs of the configuration, domain and a knowledge-based method for actually solving configuration tasks in technical domains. This approach is only useful if predefined pieces have to be combined within a temporal and spatial arrangement. But if the domain consists of correlated medical structures a main emphasis lies on the permanent adaptation of the domain model according to sample plannings. Plakon and Konwerk do not support this task.

For an interaction between medical experts and for the development of a dental domain model the tools described above are not appropriate, because most of the medical experts do not have the time to learn the formal language required to understand the model. A variable toolset for building an appropriate graphical interface for problem-solving agents is missing; this will allow a discourse between both experts from different medical domains and still provide adapted views to each of them.

## **1.2 Properties of object-oriented analysis and UML**

The object-oriented paradigm has brought forth UML as a focal point [18]. UML is a language and notation for specification, construction, visualization and documentation of models. Wieringa describes the most important object-oriented methods and compares them referring to their techniques and properties [18]: Besides OMT (1995) only UML (1997) provides sufficient expressive power for the complete coverage of the whole range of software. Concretely, UML offers development diagrams that can be divided into the following types: use case diagrams show actors, use cases, and their relations. Class diagrams describe classes and their mutual relations. To describe the behavior there are activity diagrams (they show activities, object status, transitions and events), collaboration diagrams (they show objects and their relations with their spatially ordered information exchange), sequence diagrams (they show objects and their temporal ordered relations), status diagrams (they show states, transitions and events). For the

support of the implementation there are component diagrams (they show components and their relations), distribution diagrams (shows components, nodes and their relations).

UML is not a development method but can be used as a basis for development processes, such as RUP (Rational Unified Process<sup>TM3</sup>). Well-known and evaluated techniques used in the RUP methodology are the use-case definition, support for finding classes, object-interaction models, sequence diagrams and the specification of relationships (aggregation, association, association for aggregation, naming relationships, role names, multiplicity indicators, reflexive relationships, package relationships). Behavior can be added and checking of the model is supported as well as designing the system architecture on the basis of development iterations and cycles [18-21].

Each type of diagram offers a different view to the model and the software system. Thus, advantages of UML are the large amount of modeling elements, which results in a high expressiveness and thus the ability to model a broad spectrum of domains. The communication between each developer and the whole team is supported as is the mapping of business processes to software architecture. Critical decisions with regards to design can be visualized. The controlled iterative development leads to reductions of constructive iterations. Round-trip engineering is supported for the generation of a model out of the source code.

The main points of this paradigm are procedural analysis and information hiding. Even large projects proved that it is possible to build large systems with a professional quality. On this basis, object-orientation has become a world standard in software modeling [19,22].

On the other hand, a deeper understanding for UML constructions still requires large efforts. The powerful expressiveness demands appropriate CASE-tools. Existing software does not utilize the potential of UML thus far.

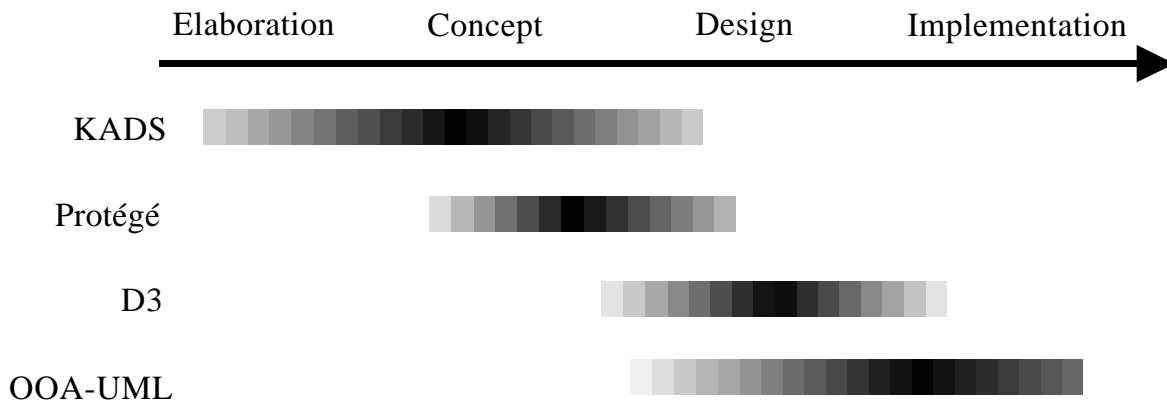
### **1.3 Comparison of the approaches with respect to dental decision modeling**

Plakon and Konwerk are generally inappropriate for dental modeling as described above. A comparison of the other knowledge-based (KADS, Protégé, D3) and object-oriented methodologies respectively shows the according range of support (Figure 1). KADS' advantages can be seen in the development of the project elaboration, concept model and design model. Object-oriented analysis, on the other hand, can strongly support the development of a design model and its implementation. Therefore, as many advantages as possible of these approaches

---

<sup>3</sup> Rational Unified Process<sup>TM</sup> is a trademark of Rational Incorporate.

should ideally be combined, to handle the overall problem of modeling dental decision making more comprehensively.



**Figure 1. Comparison of the functionality of KADS, Protégé, D3 and object-orientated-analysis in combination with UML. Roundtrip-engineering for the whole process from elaboration to implementation cannot yet be covered within one paradigm.**

The object-orientated approach is appropriate for the implementation of an operational system due to maintainability of comprehensive structures; multiple tools for design<sup>4</sup> and implementation<sup>5</sup> are available, some with features such as roundtrip engineering and visual modeling [18].

For the combination of knowledge-based analysis and object-oriented implementation, interfaces have been built within the KADS project to automatically pass the CommonKADS model to a UML tool. Unfortunately, UML cannot be mapped back into the KADS knowledge model. This implies a mismatch between these media, caused by a lack of appropriate interfaces<sup>6</sup>. Because of both the concept model and the related code cannot be developed within one single methodology. Further problems arise because the mismatch between UML and KADS makes round-trip engineering impossible. Practically, problems arising from tests of an implementation should be traced back via the source code to the concept model. Changes have to affect each level of modeling to make the outcome transparent to all members of the community. Only on this

<sup>4</sup> E.g. Rational<sup>®</sup> Rose<sup>™</sup> and Together<sup>™</sup> from TogetherSoft<sup>®</sup>.

<sup>5</sup> E.g. Microsoft<sup>®</sup> Visual C++<sup>™</sup> and Borland<sup>®</sup> C++ Builder<sup>™</sup>.

<sup>6</sup> The appropriateness of a dental model can only be checked on the basis of detailed individual planning. KADS can be used for structuring a solution, but cannot support the planning itself in detail.

basis, the effect of feedback is guaranteed. In the absence of tools that map the code back into concepts, the consistency of the various levels of development can only be achieved manually with the obvious risk that either errors occur in the process of mapping or that it is actually never accomplished. So, the operational code might perform differently than intended by the conceptual model. In other words: debugging is decoupled from the improvement of the model of concept.

Given the mismatch between object-oriented design and the KADS concept model, one might consider to leave out KADS and use the object-oriented approach only. However, object-orientation does not directly support the explicit representation of how conclusions are drawn, i.e., of an inference structure. The inferences can be coded within the methods, but they do not present themselves as an inference structure. This lack could complicate the discussion with domain experts.

In addition, the strength of Protégé and D3 could also be included into the development with KADS if there were appropriate interfaces to UML.

Additional functions and views ought to be coded to explain the outcome of each planning step to keep the overview in large decision-support systems. Therefore, it is suggested to build some kind of meta-structure above the object-oriented decision methodology, which adds appropriate justifications to the medical decision path. Similar approaches have already proven to be appropriate for technical problems [23].

## **2. REQUIREMENTS AND TASKS FOR DENTAL DECISION MAKING**

The integration of KADS-tools into the process of modeling appeared to be difficult due to missing interfaces to the chosen object-oriented implementation tools. Because of this, round trip engineering cannot not be achieved. Nevertheless, the general KADS methodology has been used for fundamental domain analysis.

A backward planning approach was chosen to comply with planning the treatment alternatives in which the general aim is predefined and all planning steps try to comply with it. Since information about the patient cannot be complete due to reasons related with invasiveness, time, finances, etc., in the beginning it has to be acquired step-by-step with a minimal effort and specifically for those aims and solutions that seem appropriate. Therefore, a *propose and revise* problem-solving method has been established on top of the backward-planning strategy.

As will be outlined in more detail below, dental planning starts with three major steps:

1. Documentation. The present state of the patient's denture is described on a rather abstract level. Presence and absence of teeth, implants and prostheses is recorded; an activity that can be done by the dentist with minimal effort [2].
2. Targeting. For the above data, whose nature is rather topological, appropriate types of future states are identified. They consist of different suggestions on how the existing topology can be modified by adding implants and prostheses to achieve functional and esthetic final states [24].
3. Planning. The major part of this article deals with planning the process from the present state to the goal state or, better expressed, from a goal state via pre-, pre-pre-, etc. conditions back to the present state.

We will briefly mention that constraints are also used. They serve the purpose to check whether the patient can be treated immediately. The exhaustive treatment for the improvement of the topology can in some cases be planned and initiated right away, but often, acute non-dental or dental problems (e.g. pain, inflammation) have to be solved first.

As already outlined above, the long-range planning strategy follows a *propose and revise* problem solving method [25]. For a suggested final state that satisfies the constraints, its immediate predecessor state is derived and patient data are checked whether this predecessor state can be realized.

For instance, it may have to be checked whether a tooth, which is intended as a support for a bridge, is healthy enough. This is a question the dentist should answer. It means that, when it comes to terms, the dentist may have to add qualitative data to the purely topological data required for the targeting. Qualitative information is, however, only asked when it plays a role in detailed planning of an apparently good solution. Predecessor states may not yet be present in the patient, but it may be possible that they can be achieved by some treatment, e.g., of conservative dentistry. In this case planning goes one step back into pre-predecessor states, etc.

Generally, a proposed solution may either find support via intermediate states, which lead back to the actual patient state, or the proposed solution may have to be revised if its sequence of predecessor states cannot be achieved by a sequence of treatments starting from the present patient state.

In the following it will be argued that representation of the knowledge should depend on the user's role.

## **2.1 Handling Different Domains in Dental Planning**

Various domains have to be taken into account for the implementation of dental decision support. General medical problems surrounding dentistry have to be associated as well as dental sub-domains. Economical and informational aspects cannot be omitted for the proper organization of the treatment.

To handle the complexity, a well-structured concept with related permissions and obligations for the medical specialties involved is necessary. Whenever a problem occurs by applying the decision methodology, it has to be assigned to a specific level of planning and thus to one specific expert group, representing a sub-field of dentistry for correction, because responsibility cannot be shared. As the concept is not directly related to the coordination of the responsibilities for the decision structure, it will not be addressed in detail in this article.

## **2.2 Roles for Building and Maintaining Knowledge Structures**

Different levels of detail are necessary for different *users*, such as experts, practitioners and patients. Before explaining technical requirements, content-related aspects will be discussed. The persons involved in building, improving and maintaining a dental decision system can be assigned to three major *developer* categories.

- a. Knowledge collectors (e.g., medical specialists<sup>7</sup>) gather documents including free text, images, etc. (informal but explicit knowledge).
- b. The knowledge structurers (e.g., medical experts, medical informationers) use this knowledge to generate a formal decision structure, which can directly be implemented and later be processed by a machine (formal decision structure and related justifications). Different views onto the decision structure are necessary for each user.
- c. The knowledge critiques deliver information about the knowledge structure. Their comments are founded on experience in applying the system to individual cases and are aimed at improving the practicability of the advice given by the machine.

Any local change of the structure might affect other sub-domains and their responsible groups. Therefore, additional responsibilities of domain-group interaction have to be defined.

---

<sup>7</sup> A specialist is defined as a practitioner having the implicit skills to perform the treatment. The experts in addition know about the current literature and approaches for both planning and treatment.

However, knowledge does not only have to be collected and structured. The consequences of the planning also have to be presented and illustrated to the users. During the meta-planning of such presentations and illustrations one faces several tradeoffs: each user group is capable of handling a different level of complexity. For that reason the representation should be adapted to their individual skills and therefore has to be designed by the knowledge structurer for covering these levels. But the different views should not diverge too much, as this might hinder the communication between the groups that are collectively building the decision structure. To comply with these demands, different levels of detail could be integrated in a hierarchical structure for the cooperative communication between the entire user group Figure 2.

### **2.3 Roles Involved in Knowledge Application**

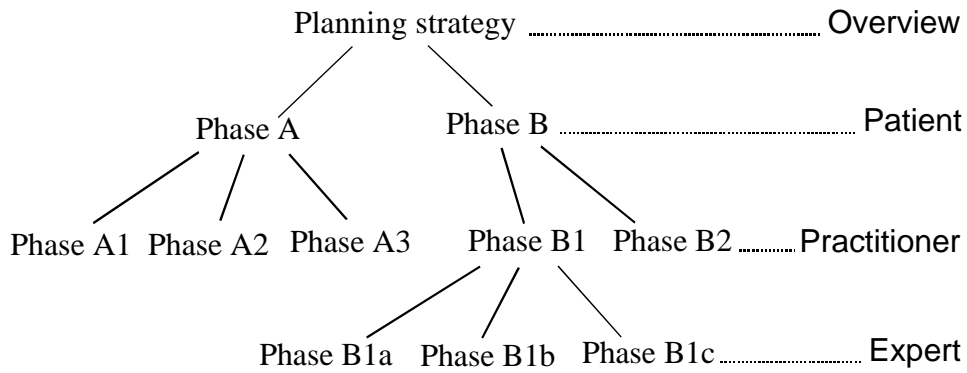
Beyond people involved in structuring, other parties are using and applying the knowledge. Practitioners, for instance, are planning the individual treatment and partial tasks may be assigned to nurses or technicians. All these parties have to perform their part of the treatment according to the overall aim, and not to forget about the patient. He or she is an important factor, not only for the choice of the treatment alternative but also for the success of the treatment itself.

Within a unified approach, all of these groups should found their decision making on the same domain model. Only the justifications should vary and adapt to the user's needs as they aim at completing the user's knowledge.

The structure of the dental decision model cannot be sketched in detail here due to its size, complexity and amount of necessary sample cases and will be discussed in a separate article. So, only the methodology of deriving user-specific justifications from the object model will be discussed in the following.

### **2.4 Levels of Presentation for Different Users**

For all of the users involved, adapted illustration and justification of the decision structure are necessary. A default level should be shown to a user according to his role. If necessary, more details can be uncovered, up to the expert level. Therefore, a decision and justification structure should be built in a hierarchical manner to be able to fade in or out various degrees of complexity (information hiding or uncovering).



**Figure 2: The Planning strategy should be presentable in various levels of detail depending on users’ needs. If necessary more details have to be uncovered.**

For patients and practitioners it might not be possible to understand more advanced levels due to the language used. For that reason a concept of rights is necessary to limit the access to comprehensible parts of the system for each user. For the conception and permanent extension of the methodology a concept of permissions and obligations is necessary both for user role maintenance and user role application of the methodological knowledge.

### **3. Applications and Benefits of the Multi-agent Approach**

After the general properties for dental decision-making have been described, experiences with the development methodology and the system will be reported. During three years, eight experts have been involved in developing the dental concept model. Five concentrated on dental aspects and three on the representation of the knowledge model.

It is not claimed that this approach is the only appropriate solution. But it seems to be suitable for dentistry with its complex treatments and technical constructions. There is no evidence yet whether it is adequate for other medical disciplines.

#### **3.1 Building and Maintaining the AIDA Inference Machine**

AIDA Version 1.1 showed that corrections both for ‘building the right system’ and ‘building the system right’ could easily be integrated based on the UML model. Within the AIDA project it was found that the classes, the objects and their methods can vary during all stages of the development and even during system application as the dental domain is not closed and thus is subjected to strong interactions with other disciplines. This leads to steady covariations with other fields of medicine and continuously new insights within the domain itself. Formerly

unknown influences can become important during all phases of the development. Therefore, a powerful approach and appropriate tools are needed for permanent adaptation of the class model. Here, the tasks to find the appropriate location for necessary changes should be supported. For dental demands, the object-oriented paradigm proved to be well suited.

The object-oriented analysis in combination with UML has turned out to be an adequate method for transforming the knowledge model into an implementable design model. Because of the high expressiveness of UML it has been possible to address the problem adequately both from a functional and an informational point of view and to develop the design model systematically. The integration of conceptual levels of KADS into the UML tools could in a way "repair" the mismatch mentioned above. The planning rules formulated within the knowledge model could be converted to classes, their methods and the object interactions on top of it with few iterations.

Concretely, AIDA was made publicly accessible in the following way: the AIDA interface for documenting case data can be downloaded (<http://aida-uni-hd.de>) and executed. This has been used by several experts and specialists who have submitted selected but typical cases as well as cases they have concretely been dealing with. On the basis of these cases the system has been assessed. Due to the expressiveness of object orientation all kinds of dental problems arising could be addressed adequately. In the first evaluation phase it took about three weeks to identify medical problems of the first prototype and only three days to implement necessary changes to the system.

The results of the evaluation have to be discussed in a different article on a dental level in detail. However, the comparison of the AIDA-planning and real-life cases has shown that 68% of the cases can already be identified by the decision rules of Version 1.1 for prosthetic planning. It seems to be possible to strongly enlarge the planning quality of AIDA if aspects of conservative dentistry are included in the prototype, because in many cases there is a more complex interaction between the denture and the underlying natural structures.

### **3.2 Communication among Multiple Developers**

The system has reached a level where support of the planning in daily routine seems to be feasible soon, as the neutral platform for interdisciplinary discussion between specialists has already proven to be valuable as a mediator of the decision rules.

The visualized modeling method allowed to continue the communication from the design phase up to the implementation of the planning tool. The presentation in UML offered all

necessary means to express the medical contents in a way intelligible for dentists and medical experts. The defined cases and the developed diagrams supported an appropriate debate of the subject based, on domain-specific notions. Also, the experts' concepts – e.g., in the field of class methods – could easily be illustrated to interested practitioners. The option of information hiding was helpful for the introduction of our approach to novices.

The contribution of the system to rationalizing the discussion about the dental domain is underlined by the following observation: in-depth analysis of those cases where the system recommendation deviated from the expectations of experts or practitioners revealed a problem in documenting the finding statements. During the evaluation phase the system required a certain protocol for the communication of the patient's status. But without being aware of it the cooperating dentists often misused finding abbreviations.

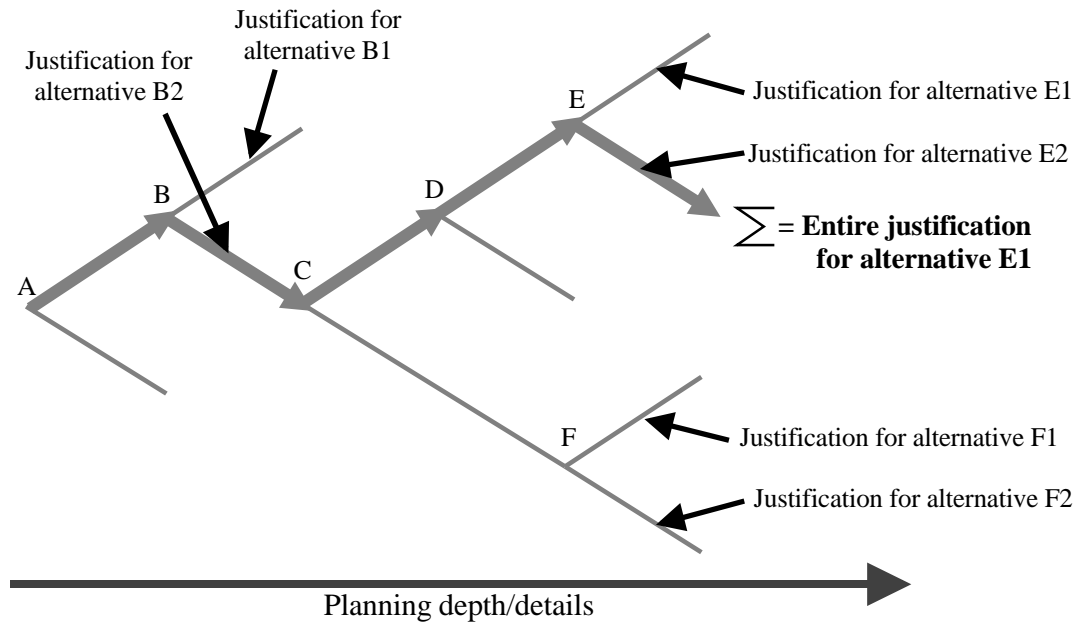
#### **4. Technical Tasks for Future Work**

The services described below have already been developed. Not all of the respective implementations have started yet, but they seem to be feasible in reasonable time. Therefore, we will focus on the discussion of fundamentals planning problems.

##### **4.1 Explanation of Individual Decisions**

Several disadvantages of the modeling methodologies described above have been discovered (not for the development of the dental decision methodology itself, but for the appropriate representation of referring justifications. An implementation of justification structures is neither supported by UML nor by the object-oriented tools. Therefore, in the following a solution for this problem will be explained theoretically and practically.

For each decision the appropriate justification has to be formulated for the referring user group. As the dental planning is aim-oriented, the planning strategy zooms into the problem space. Thus, the sum over all partial justifications along the decision path has to result in a complete justification (Figure 3). Several such paths may exist, one for each plan that has not been discarded in the *propose and revise* iteration [8,10]. For those paths that survived, their justification is the result of *establish and refine* elaboration of the knowledge applied. In this way, the planning and justifications can be arbitrarily extended into depth.



**Figure 3: During the planning procedure the sum over all decisions is traced. Thus, the sum over all partial justifications results in the overall justification of the current solution. Alternative justification can be linked for different user roles (practitioner, patient, nurse, etc.).**

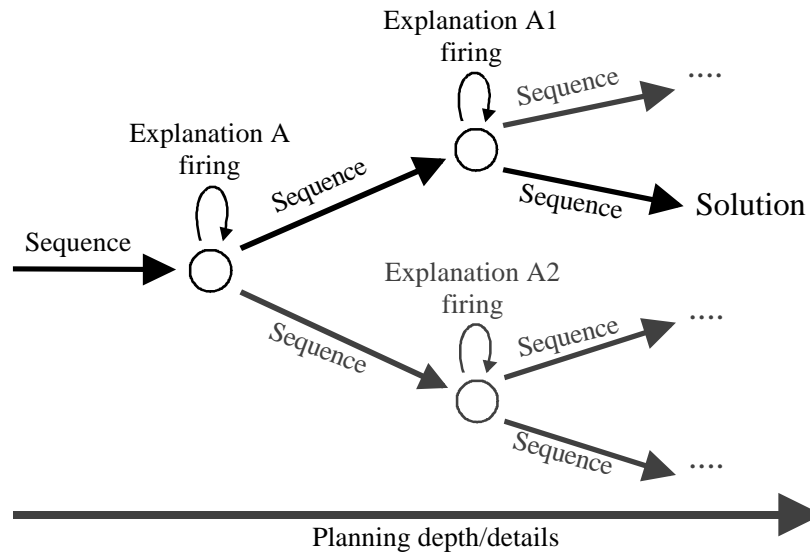
#### **4.2 Integration of Justifications into the Inference Agent**

To better understand a single decision in the context of the planning strategy within a given case, justifications of the decisions are necessary. The justifications can be superimposed on the existing decision algorithm by adding ‘justification’ methods to the decision sequence at appropriate locations. ‘Junctions of decisions’ in the sequence diagram can characterize the proceeding. Concretely, the chosen decision path can be defined by the referring partial choices, which form the whole treatment planning.

In doing so, this both procedural and declarative aspects of the problem can be combined in the same decision model (Figure 4).

Special authoring features should be implemented into standard software development tools for the integration and management of the various levels of justification. If the development tools are extended by an additional view for illustration and correction of merely these justifications, problems can more easily be focused, discussed and changes be applied to the decision structure without inconsistencies.

A separate view on formal knowledge could be a strong support in terms of information hiding, because planning sequences can be summarized and explained.

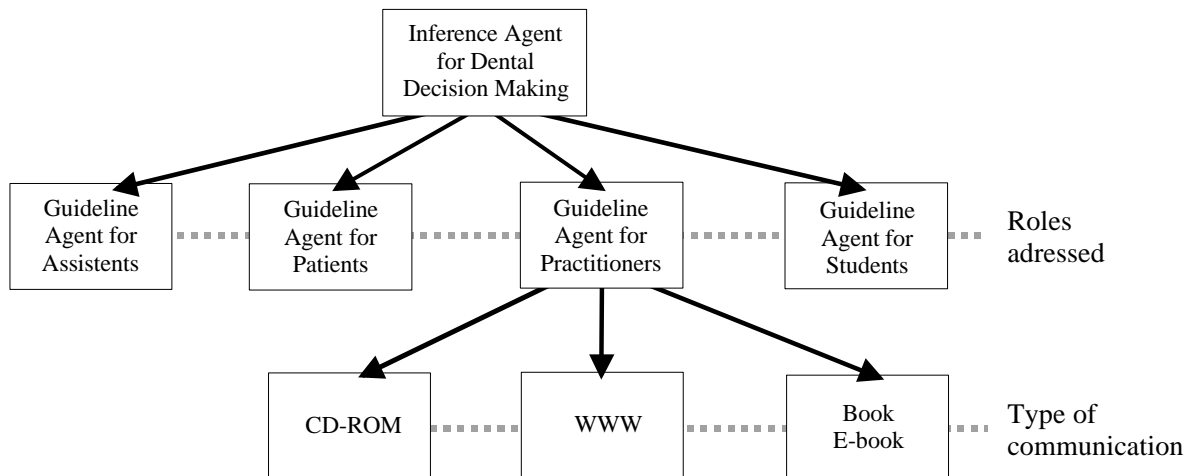


**Figure 4: Knowledge-based components can be added to the object-oriented model by justifications ‘firing’.**

From a different point of view the superstructure can be used as an extended referencing methodology for any type of explicit knowledge. Queries on the justification structure, for instance, can subsume the assigned documents for the explanations ‘firing’. If they justify each partial decision appropriately, they can be joint together for the overall explanation of the planning. This can hardly be achieved by a mere retrieval of keywords on the object model.

### **4.3 Adding Illustration Agents to the Decision Structure**

Not only modeling of the expert’s decision structure is necessary. The structure also has to be made transparent to all users involved to make an integrated treatment possible. For that purpose, adapted justifications can be derived from the decision model as described above. An additional layer for illustration can support the organization of related documents. Various types of concepts are feasible. For instance, the derivation of views can be concentrated on the person addressed (Figure 5) or the type of communication.



**Figure 5: As a derivation of the inference model, an illustration agent generates problem- and user-specific views onto the domain. The information provided within one view – which can be called ‘guideline’ – can automatically be compiled to the required media, resulting in a CD-ROM, WWW-presentation, or a book.**

Principally all kinds of media can be included in this approach: text, multimedia documents, audio and video material can be referenced within the justification structure. Technically, fragments of complete explicit documents are attached or linked to the formal decision structure.

## 5. Summary of the Multi-agent Knowledge Model

The comparison of real-life dental treatment with a certified derivation of dental solutions from the patient status requires an acknowledged planning strategy. It was not clarified before the AIDA-project, to what extent and with which quality complex medical planning tasks such as dental decision modeling can be supported by existing modeling techniques.

None of the existing modeling paradigms alone can support this task appropriately. A development environment has to be powerful enough to provide a large variety of problem-solving methods and knowledge-based authoring functions for justification. Only a combination of the KADS methodology and object-oriented UML seems to provide the necessary spectrum for dental decision modeling. However, in this way a mismatch cannot be avoided during development.

The fundamental elaboration of the decision approach has been achieved on the basis of the KADS levels of description: once comprehensively modeled in KADS, the objects and methods for decision making could be appropriately mapped into UML. To cope with the complexity of

both modeling and system development, a multi-agent structure has been developed. The inference structure is subdivided into agents for prosthetics, conservation, surgery, etc. and concentrates on deriving both the treatment alternatives and the stereotypical manipulations to carry out each alternative.

Additional agents for the illustration of each planning phase will be added on top of the inference machine. To include knowledge-based functions within the domain model a multi-layered justification structure can be superimposed on the dental inference machine to supply various user groups with appropriate justifications. This justification technology could help to reduce the major disadvantage of object-oriented approaches, lacking comprehension of the domain model. UML tools could be extended by additional views for justification so that a deeper understanding of decision constructions can be mediated.

The AIDA approach can be used for the definition of a dental class library for both the representation of biological structures and the planning algorithm. The communication of such an object model can support the exchange between researchers in dentistry.

In future, a combination of the KADS approach and UML development environments could ease the task of medical decision modeling on the basis of a unified round-trip-engineering tool. The construction, use and adaptation of large decision-support systems in the long run can only be achieved on the basis of a comprehensive representation from concept via design to implementation.

## **Literature**

1. Chernoff H, Moses LE: Elementary decision theory. vol. XV. New York: Wiley; 1959.
2. Finkeissen E, Schwarzer, H, Wetter, T, Hassfeld, S: Voraussetzungen einer Medizinischen Wissens- und Entscheidungsreferenz (Pre-conditions of a Medical Knowledge and Decision Standard). In: Jaeckel A, Telemedizin Führer 1999: Deutsches Medizin Forum; 1999.
3. Pauker S: Clinical problem-solving. Some familiar trade-offs. N Engl J Med. 1994; 331:1511-4.
4. Pauker S: Clinical problem-solving. Invasive interventions. N Engl J Med. 1994; 331:601-5.
5. Pauker S: Clinical problem-solving. A masked marauder. N Engl J Med. 1994; 330: 1596-8.
6. Brim OG, Glass, D.C., Lavin, D.E. et al.: Personality and Decision Processes. Stanford; 1962.

7. Wetter T: First-order logic foundation of the KADS conceptual model. In: Wielinga B, Boose J, Gaines, B, Schreiber, G & van Someren, M, Current trends in knowledge acquisition. Amsterdam: IOS Press; 1990. p. 356-375.
8. Wielinga B, Schreiber, AT & Breuker, JA: KADS: A modeling approach to knowledge engineering. Knowledge Acquisition - Special issue 'The KADS approach to knowledge engineering' 1992;4.
9. Breuker J, van de Velde, W, et al.: CommonKads library for expertise modeling, Reusable problem solving components. vol. 21: Amsterdam Oxford IOS Press; 1994.
10. Schreiber AT, Akkermans, JM, Anjewierden, AA, de Hoog, R, Shadbolt, NR, Van de Velde W, Wielinga, B: Engineering and Managing Knowledge - The CommonKADS Methodology. 1999.
11. Puppe F, Gappa, U, Poeck, K, Bamberger, S: Wissensbasierte Diagnose- und Informationssysteme - Anwendungen des Expertensystem-Shell-Baukastens D3 (Knowledge-based Diagnose and Information System – an Application of the Expert System Shell Tool D3). Berlin: Springer; 1996.
12. Tu SW, Erikson H, Gennari J, Shahar Y, Musen MA: Ontology-based configuration of problem-solving methods and generation of knowledge acquisition tools; The application of PROTÉGÉ-II to protocol-based decision support. Artificial Intelligence in Medicine. Artif-Intell-Med. 1995;7:257-89.
13. Musen M (editor): David J, Krivine JP, Simmons R: An Overview of Knowledge Acquisition. Berlin: Springer; 1993.
14. Musen M, et al.: Protégé-II, Computer support for development of intelligent systems from libraries of components. Medinfo. 1995;1995:766-770.
15. Grosso WE, Eriksson, H, Ferguson, RW, Gennari, JH, Tu, SW, Musen, MA: Knowledge Modeling at the Millenium (The Design and Evolution of Protege-2000). In: . Stanford: SMI; 1999.
16. Cunis, R, Günter, A, Strecker, H: Das Plakon-Buch - Ein Expertensystemkern für Planungs- und Konfigurationsaufgaben in technischen Domänen (An Expert System Kernel for Planning and Configuration Tasks in Technical Domains). Berlin - Heidelberg - New York: Springer; 1991.
17. Schröder, C, Möller, R, Lutz, C: A partial logical reconstruction of PLAKON/KONWERK. Dresden; 1996.

18. Wieringa, R: A Survey of Structured and Object-Oriented Software Specification Methods and Techniques. *ACM Computing Surveys* 1998; 30: 460-527.
19. Booch G, Rumbaugh, James, Jacobson Ivar: *UML: The Unified Modeling Language user guide*. Reading, Mass.: Addison Wesley; 1999.
20. Rumbaugh J, Jacobson, I, Booch, G: *The Unified Modeling Language Reference Manual*. Berkeley: Addison-Wesley; 1999.
21. Oestereich B: *Objekt-orientierte Softwareentwicklung mit der Unified Modeling Language (Object-oriented Software Development with the Unified Modeling Language)*. 3., updated edition, Munich: Oldenbourg; 1997.
22. Quatrani T: *Visual Modeling with Rational Rose and UML*. Harlow, England: Addison Wesley; 1998.
23. Gehbauer F, Lennerts, K: *Modellierung und Entwicklung eines wissensbasierten, objektorientierten Systems zur Planung optimierter Baustellen-Layouts; ESBE (Modeling and Development of a Knowledge-based, object-oriented System for the planning of an optimized Layout of Building Sites)*. In: *Forschungsgemeinschaft D, Objektorientierte Modellierung in Planung und Konstruktion - Forschungsbericht*. Weinheim: Wiley-VCH GmbH; 2000.
24. Finkeissen E, Weber, R, Wetter, T, Hassfeld, S, Koke, U: *AIDA wird in wenigen Sekunden detaillierte Lösungen bieten können (AIDA can Provide Detailed Plannings within Seconds)*. *DZW-Special* 1999.
25. Marcus S, McDermott, J.P.: *SALT: A Knowledge Acquisition Language for Propose-and-Revise Systems*. *Artificial Intelligence* 1989; 39: 1-37.