3D Building Model-Based
Life-Cycle Management of Reinforced Concrete Bridges

M. Kluth, A. Borrmann & E. Rank

Computation in Engineering, Technische Universität München, Germany

T. Mayer & P. Schiessl

Centre for Building Materials, Technische Universität München, Germany

ABSTRACT: Since large parts of western countries' infrastructure have been erected in the early 1960's, many infrastructure buildings such as concrete bridges are now entering a critical age, where the need for continuous inspection and reconditioning is becoming apparent. Facing these challenges, many national highway authorities have developed computer systems that allow for the management of data collected during bridge inspections. Based on this data the current condition of the inspected bridge is manually assessed and, if necessary, revivification measurements are planned. The life-cycle management tool presented in this paper goes one step further: It allows a 3D model-based, component-wise collection of detailed material and inspection data and a mostly automated computation of the bridge's future state by using probabilistic deterioration models for all individual components and hierarchical aggregation of the components' conditions. The basis of the component-wise input and processing of all relevant data forms a hierarchical 3D building model, which is described in detail in the paper. The particularity of our approach is that the building model can easily be adapted to specific building types or the special demands of regulatory authorities by means of an explicitly available meta-model.

1 INTRODUCTION

1.1 *Motivation*

Most of the public and private bridge operators in the western countries are faced more and more with the problem of maintaining a stock of aging bridges with just limited funds at their disposal (Schiessl and Mayer 2006). This has given rise to the development of *Building Management Systems* (BMS) and *Life-cycle Management Systems* (LMS). These systems offer computer-aided support for planning and realization of bridge inspections and repair works.

The most import feature of an LMS is that information from inspections is stored and can be reviewed at any time. Furthermore this data is used to compute the building's current and future condition, described within a system of condition grades. Two major drawbacks of many existing LMS are the lack of adequate deterioration models and the use of merely visual inspections in order to detect possible structural damages (e.g. cracking, staining, spalling). In most cases the optimal point in time for (preventive) repair measures of concrete building elements is already exceeded once deteriorations are visible at the concrete surface.

Predictive life-cycle management systems (PLMS) follow a novel approach to overcome this major drawback of conventional LMS. To detect deteriorations earlier additional non-destructive and visual inspection techniques are used in combination with fully-probabilistic deterioration models to predict the condition development of building elements. The prognosis is constantly updated, thus yielding a more precise prognosis of the future condition development. Such a PLMS can thus be used to optimize the operation of bridges over their entire service life (Schiessl and Mayer 2006). Furthermore, the system supports the institutions responsible for bridge maintenance in the long-term planning of inspections and repair measures at both bridge-level and network-level. Another advantage is that budgeting for reconditioning measures can be planned more precisely.

In a current research project we are developing a software-tool for the predictive life-cycle management of reinforced concrete bridges. A 3D building information model (BIM) forms the center of this system. All relevant data like measurement results, photos, or condition prognoses are stored within the BIM. This way, the owner of a stock of bridges can easily obtain an overview of the condition of individual structures or the complete bridge stock. The hierarchic subdivision of structures into building elements, sub-elements and hotspots chosen for the BIM allows for a detailed allocation of information.

1.2 *Related Work*

Several life-cycle management systems for bridges are already in operation. In Germany "SIB-Bauwerke" has been developed by the Bundesanstalt für Straßenwesen and is now in use on national and federal state level (Haardt 2002). The following list shows some systems from other countries (the list does not lay claim on completeness):

- Bridgelife in Finland (Vesikari 2006),
- DANBRO in Denmark (Henriksen 1999),
- Eirspan in Ireland (Duffy 2004),
- Kuba-MS in Switzerland (Haller and Basurco 2006),
- Pontis (Robert et al. 2003) and Bridgit (Hawk 1999) in the USA
- Ontario Bridge Management System in Canada (Thompson 1999)

At present “mobile model-based bridge lifecycle management systems” are being developed in Canada (Hammad et al. 2006).

The above listed systems can be characterized by the following properties:

- Except for the “mobile model-based bridge life-cycle management systems” (Hammad et al. 2006) the geometry of the bridge is not stored. Thus, photos or measurement results can only be allocated to building elements as text information.
- Adding a bridge to such a system, the bridge is structured horizontally into “parts” and vertically into levels. The number of levels differs from system to system. The smallest “part” in all these systems is a building element. A further subdivision is not used in any of these systems.
- Computing the condition prognosis of building elements or the whole bridge deterministic models (e.g. Haardt 2002, Henriksen 1999) or Markovian Chain-systems (e.g. Vesikari 2006) are used, respectively. Fully-probabilistic deterioration models are not used in any of these systems. The condition of a building is assessed manually, solely on the basis of visual inspections. Other non-destructive inspection methods are rarely used.

1.3 *Deterioration mechanisms, measurement methods, repair measures*

Reinforced and pre-stressed concrete bridges can be subject to different deterioration mechanisms such as corrosion of the steel reinforcement, freeze- and freeze-thaw-attack or alkali-aggregate-reaction.

Among them, the corrosion of the steel reinforcement due to chlorides from de-icing salt or carbonation of the cover concrete are the most important mechanisms. Chloride-induced corrosion is initiated once the concentration of chlorides ingressing from the concrete surface to the steel rebar exceeds a critical chloride concentration and shifts the rebar surface from its originally passive state into a state of active corrosion.

As both chloride ingress and carbonation of the concrete cover as well as the early stages of rebar corrosion cannot be detected visually, different non-destructive inspection methods have been developed in order to assess the present state of the structure (e.g. potential mapping, concrete cover mapping, determination of carbonation depth or determination of chloride profiles).

Aside from these possibilities to assess the condition state on site, recent achievements in the field of deterioration modeling nowadays allow for a reliable prognosis of both chloride ingress and carbonation and the subsequent corrosion of the reinforcement by means of fully-probabilistic deterioration models (Gehlen 2000). Inspection results can be used to continuously update the original prognosis, thus yielding an improved knowledge of the future condition development as the base for possible rehabilitation measures. Depending on the owner's maintenance strategy, these measures can range from preventive coating of concrete surfaces to a replacement of chloride-contaminated concrete or a cathodic corrosion protection of actively corroding structures.

2 THE PREDICTIVE LIFE-CYCLE MANAGEMENT SYSTEM

2.1 *Architecture*

The architecture of the life-cycle management system we are developing is shown in Figure 1. The systems consists of five modules, each of them representing a particular stage in the life-cycle management workflow, and a central database where all required data is stored. The individual modules are:

Acquisition module. In a first step, the bridge manager enters all construction-related information, including the 3D geometry of the bridge, the mate-

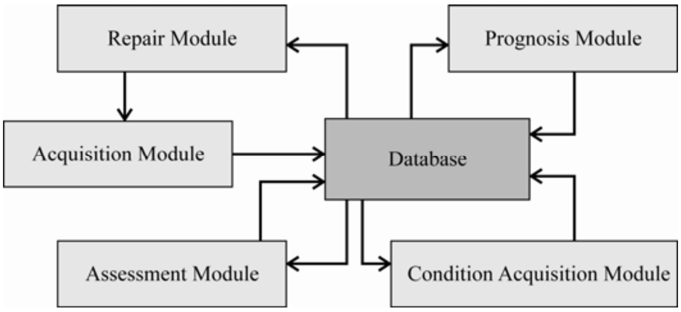

Figure 1. The modules of the predictive life-cycle management system (Schiessl and Mayer 2006).

rial parameters and the known environmental loadings. Additionally, the bridge model is semantically structured by assigning classes of building elements to individual 3D solids, grouping similar elements, and arranging them in a level-of-detail like hierarchy. To this end, the responsible engineer uses the acquisition module which offers not only means to import 3D geometry from standard CAD formats, but also provides an intuitive way for structuring the building and assigning life-cycle relevant parameters to individual components or even individual faces of the component's surface.

Condition Acquisition module. The condition acquisition module helps in capturing the current condition of the respective building. In the first step this module is used to determine which kind of inspection methods are required. Depending on the building's or building element's condition, adequate inspection measurements are suggested. The actual condition data can be obtained from discontinuous or continuous, laminar or single-pointed measurements and from sensors. In case of an inspection, the module assists in capturing newly found cracks by providing means to “draw” them or to stick digital photos on the respective surface in the 3D model, for example.

Results from measurements carried out during an inspection (such as concrete cover mapping, carbonation depth or depassivation depth, chloride profiles) are also stored in the PLMS by means of the condition acquisition module. The module furthermore is able to collect and filter data from databases that are fed from sensors such as multi-ring electrodes for the determination of moisture profiles or anode-ladders for the time-dependent ingress of the depassivation front. Again all measurement and sensor information is localized within the 3D building model.

Prognosis module. Using the prognosis module, future condition changes are calculated for every building element. For the update of the original prognosis, results from non-destructive inspection methods for each building element are used. Therefore deterioration models must be defined and stored in the database.

Assessment module. The assessment module determines the optimum time for any kind of repair measures on the basis of the actually prognosticated condition changes. One aspect for repair measures is to eliminate possible damages at an early stage, thus the construction will last for a longer time and money can be saved because such a *preventive* repair measure is in many cases cheaper than the repair of a damaged building element.

Repair module. Whenever repair measures are taken they are recorded in the PLMS using the repair module. Also, the building's component condition after these repair measures is stored. This information is used by the prognosis module to compute a Bayesian update for all condition prognoses.

After a building's completion and taking over, the PLMS will be used for the first time to acquire all relevant information. In this step, the planning data is updated with the real data from the building, which were obtained by using measurement methods. A typical example is the extent of the concrete cover - in many cases it differs from the value instructed by the construction engineer due to an imprecise erection process. This first update of the building's condition prognosis is called a *Birth Certificate* (Mayer et al. 2008).

3 THE HIERARCHICAL BUILDING INFORMATION MODEL

3.1 *3D building model*

Existing life-cycle management systems only provide means for textual input of inspection data. Thus the inspection planner has the challenging and error-prone process to mentally assign material parameters, damages and measured values to real locations and individual building components.

We therefore propagate the use of a 3D building model as the center of all data acquisition and data retention activities. By means of this model, all information about material properties, environmental loadings, deteriorations, inspections, repair measures, and condition changes are stored with reference to geometric objects. Furthermore all results of non-destructive inspection techniques or photos taken during inspections can be attached to the geometric representation of the corresponding building component. Using a 3D model, all relevant information is easily allocated and a very good overview is guaranteed.

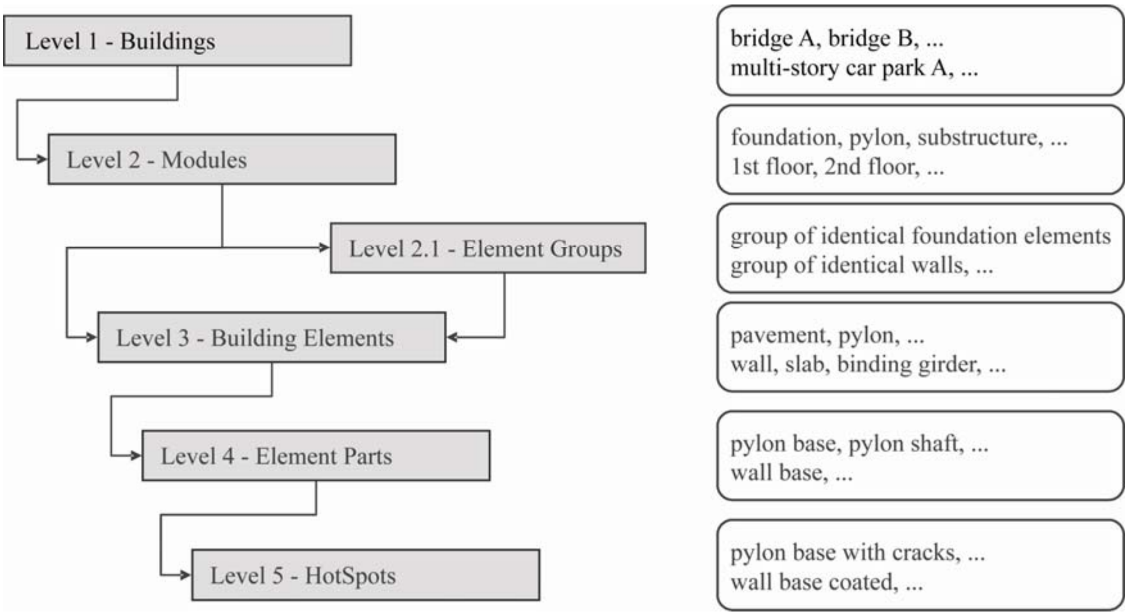

Figure 2. The hierarchical Building Information Model in use for the PLMS consists of five main levels and an additional grouping level. All child elements are physically contained in the parent element, such that their geometry forms part of its parent's geometry. The prognosis is based on this hierarchy such that the condition grades of the lower level define that of a higher level (Schiessl and Mayer 2006).

Existing life-cycle management systems only provide means for textual input of inspection data. Thus the inspection planner has the challenging and error-prone process to mentally assign material parameters, damages and measured values to real locations and individual building components.

We therefore propagate the use of a 3D building model as the center of all data acquisition and data retention activities. By means of this model, all information about material properties, environmental loadings, deteriorations, inspections, repair measures, and condition changes are stored with reference to geometric objects. Furthermore all results of non-destructive inspection techniques or photos taken during inspections can be attached to the geometric representation of the corresponding building component. Using a 3D model, all relevant information is easily allocated and a very good overview is guaranteed.

3.2 *Multi-level semantical model*

A pure geometric model would not fulfill the demands on a PLMS - the type of each building component and its role in the structural system are decisive for its influence on the condition of the overall building. The system therefore forces the user to assign classes to individual geometric elements, following a kind of reverse product modeling approach. We call it 'reverse' since in the standard product modeling process, the semantics of an object are first defined, and based on that, a geometric representation is assigned (Eastman 1999).

Additionally we realize a multi-level, hierarchical model here. The proposed hierarchy consists of five different levels, each entity on one level has none or more child elements on the next level. All child elements are physically contained in the parent element, such that their geometry forms part of its parent's geometry. Except for the last level, all child elements together form the parent element - a typical aggregation relationship (see Figure 2).

The **first level** represents entire *buildings*. Since the PLMS is designed to manage all buildings of a respective state authority or private owner, we can have multiple entities on this level, such as *bridge A*, *bridge B* etc.

The **second level** comprises *modules*. Modules are groups of building components with identical functionality. For example, reinforced concrete bridges will typically consist of *foundation*, *bridgehead*, *pylon* and *superstructure* modules.

The **third level** consists of individual building elements such as *pavement*, *pylon*, *wall*, *slab*, *girder* etc. On an intermediate level, called level 2.1, all building elements with exactly the same geometry and identical environmental loads are grouped together. This feature was integrated to avoid the multiple input of the same data for a number of identical objects. Note that individual building elements can be removed from a group whenever parameters occur which are specific for that element.

The **fourth level** represents individual parts of these elements. Such *sub-elements* like *pylon base*, *pylon shaft* or *wall base* are required to capture specific environmental stresses that occur only at parts of building components. For example, wall bases are especially exposed to splash water which may contain high concentrations of chlorides from de-icing salts and thus cause faster chloride ingress and an earlier corrosion of the steel rebars.

On the **fifth level** *hotspots* are managed. Hotspots represent sectors having a low material resistance and or extraordinarily high environmental loadings which are critical for the condition of the whole construction. Such hotspots can be set by the engineer or bridge-owner in the initial data acquisition process or added as soon as local changes of environmental loadings or extraordinary deteriorations, such as cracks, are detected. A typical example for a hotspot set by the engineer is the anchor point of prestressing tendons.

The vertical structuring of a given 3D bridge model and its subdivision into sub-elements and hotspots is performed manually but with strong support by the 3D software interface. Though the process can be seen as rather tedious, it is absolutely necessary in order to make optimum use of the fully-probabilistic deterioration models mentioned above. Only by means of hotspot entities for example, is it possible to capture local deteriorations -- an unavoidable pre-requisite for a precise prognosis of the whole structure's condition. Note that data about material resistance, environmental loads and geometric details is allocated only to elements of levels three to five.

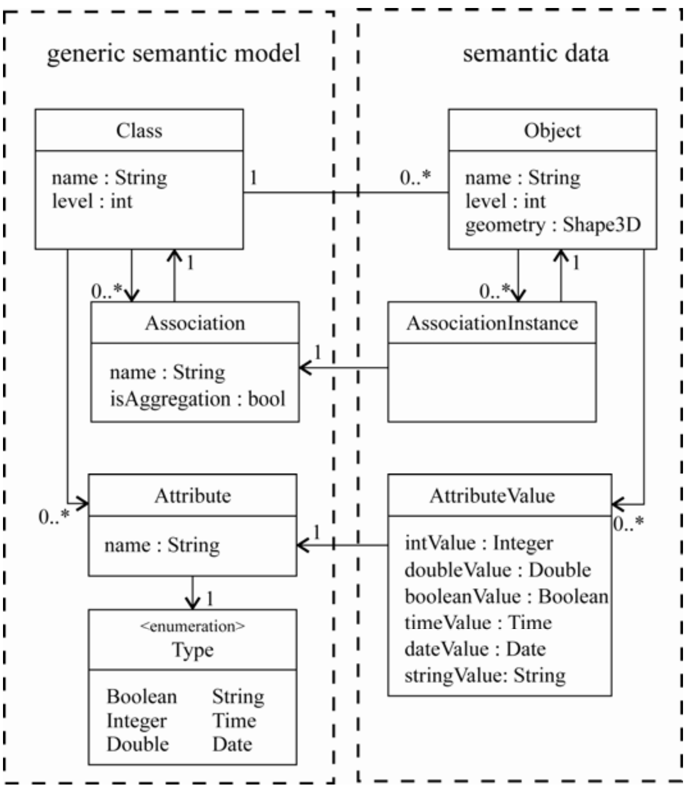

Figure 3. The meta-model used to provide an adaptable Building Information Model. Instances of the meta-classes on the left-hand side are used to model the classes of a domain-specific model and their relationships. Instances of the generic classes on the right-hand carry data of instances of this model.

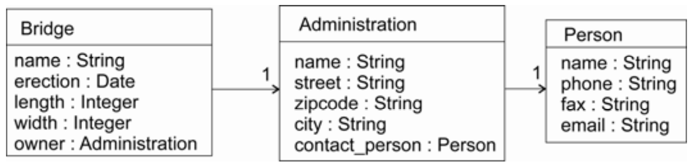

Figure 4. Sample model: bridge – owner – person

Determining the condition of a structure is realized by a bottom-up aggregation of the condition over all levels. With the described levels of detail approach, the owner of a bridge can easily acquire detailed information or assess a construction's condition from the hotspot level to the entire building. This aggregation is achieved by employing multi-attribute-decision-algorithms (MADA) that allow for a different weighting of the criteria important to the owner (e.g. financial/environmental aspects, health, safety), see (Lair et al. 2003) and (Norris and Marshall 1995). To make sure that no important information is lost in this aggregation process, additional limit states are defined for every single element, and the excess of these limit states calls for immediate remedial actions independent of the aggregated state of the whole structure.

3.3 *Making the semantic model adaptable*

Since the developed life cycle management system should not be restricted to the maintenance of bridges, but instead be applicable for a wide variety of building types, we integrated functionality that can dynamically adapt to the available semantic data structures dynamically. This has been realized by means of an explicitly available meta-model.

Figure 3 shows this model; on the left hand side the meta-classes CLASS, ATTRIBUTE and ASSOCIATION are depicted.

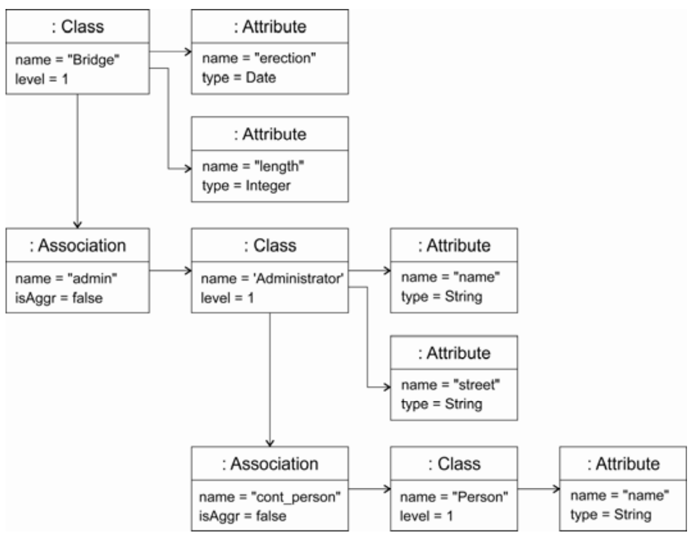

Figure 5. UML object diagram: The domain model of Figure 4 is modeled using instances of the meta-classes from Figure 3, left-hand side.

The instances of theses meta-classes can be used to model the classes of a specific domain model. For the bridge domain model, for example, we would model a class named *Bridge* having an attribute *owner* of type STRING, an attribute *erection* of type DATE etc. The available simple data types are provided by the enumeration TYPE which consists of BOOLEAN, INTEGER, DOUBLE, STRING, DATE and TIME.

However, we often need to model more complex attributes. For example, we might want to store not only the name of the owner as a simple string, but instead use a complex *Owner* class comprising the attributes *name, address, responsibility* etc. To realize this, the meta-model offers the ASSOCIATION facility: each CLASS object can have an arbitrary number of associations. Each ASSOCIATION object itself will point on exactly one other CLASS object. In our example, the latter would be the *Owner* class with the aforementioned attributes. Since the associated class can have associations itself, an arbitrarily complex data model can be generated.

An instance of the meta-model defines the Building Information Model used by the system. Figure 4 shows an instance of the meta-model representing the sample model *Bridge-Owner-Person* and Figure 5 shows the corresponding object diagram.

In our concept, only specially trained personnel (administrators) should have access to the data modeling facilities, since changes in the data structures have a crucial impact on the functions of the whole system. The normal users such as bridge managers, inspectors or regulation authorities work with input masks that are directly generated from the data model, but are not able to change it.

On the right hand side of Figure 3 the meta-classes are shown that are capable to hold instance data, i.e. the data of one particular building. They are generic in the sense that they work for any domain model. However, their structure is controlled by the connection to meta-classes on the left hand side. Since all input masks are generated from the domain model defined through the meta-model (see for example Figure 6), the inputed data automatically conforms to the desired data structure. But even without the user front-end, each data entity stays semantically structured: as can be seen in Figure 3, an ATTRIBUTEVALUE, for example, 'knows' to which ATTRIBUTE it belongs.

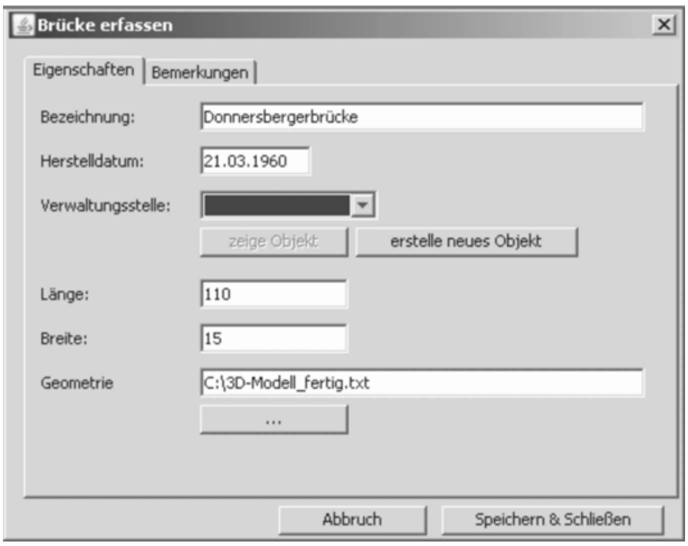

Figure 6. Automatically generated dialog box.

The concept of a BIM that is on the one hand centered on geometry and on the other hand adaptable with respect to the semantic data structure by means of an explicitly available meta-model has already been approved in a number of research projects, e.g. in the context of architectural and structural modeling in early design phases (Steinmann 1997, Kowalczyk 1997), in modeling existing buildings for revivification measures (Hauschild et al. 2003a), and for the realization of a Collaborative Computational Steering system (Borrmann et al. 2006).

4 IMPLEMENTATION

4.1 *General information*

The software tool presented in this paper is implemented by coupling a Java application (Sun Microsystems 2006a) with a relational database. All construction information, including geometric data, is stored in this database. A relational MySQL database (MySQL 2007) was chosen as the database management system (DBMS). The advantages of this DBMS are fast access to data, the possibility of storing large amounts of data, and no license fees. The three-dimensional representation of the construction's geometry is achieved with the Java 3D library (Sun Microsystems 2006b).

The advantage of the programming language Java is that applications written in this language can easily be used on different operating systems like Microsoft Windows or Linux. This aspect is very important because potential users of this application (such as administrations, construction firms with PPP-projects) may use different operating systems.

A screenshot of the software tool is shown in Figure 7.

4.2 *Implementation and use of the meta-model*

Realizing the aforementioned adaptable semantic model implies special requirements for the database and the graphical user interface (GUI). According to the description of the meta-model shown in Figure 3 the database structure consists on the one hand of relations used for the generic semantic model and on the other hand of relations containing the semantic data. In the database, a relation is needed for every attribute type. Since every class can have an arbitrary number of attributes of each type, additional link relations are used to model these m:n-relations, see Figure 8.

In the first step the administrator or some other authorized user creates classes that form the semantic model. These classes are stored in the database. In order to generate the input masks for using the software tool, the following information for each attribute is required: sequence of attributes, limit values for attributes of type INTEGER or DOUBLE, and the number of characters for attributes of type STRING. Determining the sequence of the attributes is important, because this way it is possible to topically arrange attributes of different types in the later input masks. The limit values are needed to identify erroneous user inputs such as negative dimensions.

All input masks will be automatically generated during usage of the software tool, see Figure 6 for example. To this end, the required class from the generic semantic model is loaded from the database. The attributes are placed in the given predefined sequence by the administrator. The system provides assistance for every input. This is done using tooltip texts and a support system. In addition, the user is informed whenever incorrect information has been entered or information is missing when closing a dialog box. Further information is given if complex attributes are used and there is still some input missing when closing a dialog. As soon as all entries are correct and accepted by the user, a new object is generated and stored in the database.

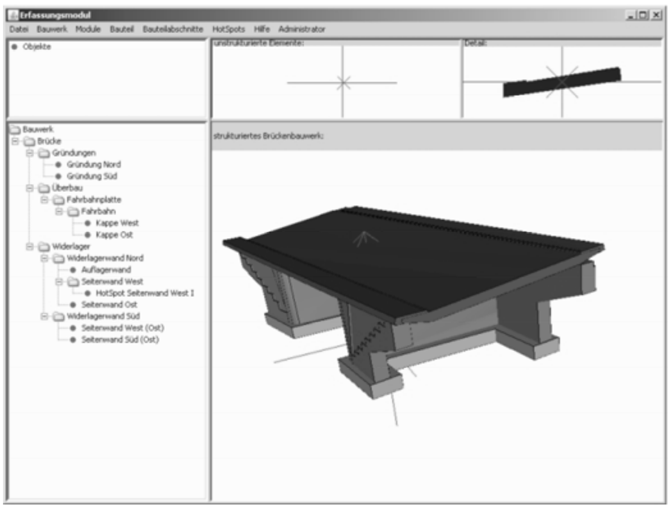

Figure 7. Graphical User Interface of the PLM system.

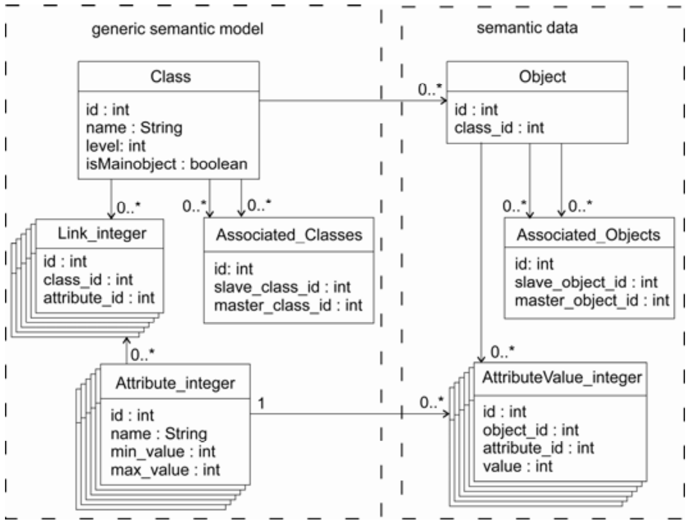

Figure 8. Database structure for the meta-model. Layered boxes describe similar relations used for other attribute types.

The following input components are automatically created by the software tool: If the user creates a new building (level 1) he has to choose a file that provides the geometry of the building as a B-Rep model. All geometry data is stored in the database, as explained in the next section. In the next step, the user has to structure the building model according to the vertical levels and the semantic data model (see Section 3.2). To realize this, he creates instances of the available classes and assigns a geometric object to everyone. The user is supported during this process by visualizing classified and non-classified building elements in different 3D views. When creating a new element part, the user may choose whether this object has a geometric representation or not. A building element does not have its own geometric representation if it is completely subdivided into sub-elements. However, for all new sub-elements and hotspots a geometric representation must be chosen. Every object with a geometric representation needs further input of material properties.

All construction information stored in the database can be visualized and edited anytime by the GUI. One special aspect of this GUI is that an intuitive and user-friendly interaction is possible everywhere – at an office desk or during a bridge inspection.

4.3 *Geometry*

Visualizing the geometric representation of the building is realized by means of the Java3D library (Sun Microsystems 2006b), see Figure 7. Using this library the user can interactively translate, rotate, or scale the model. Furthermore a single object can be marked by clicking and the properties of this element will be shown.

The software prototype uses a geometry structure very similar to the geometry kernel ACIS (Spatial 2006), see Figure 9. The advantages of this structure are: All building elements are represented by faces. Therefore nearly all information in this tool is stored geometry-based, it is very important to have direct access to all faces. Furthermore it is possible to describe building elements with holes, like box-section slabs. The export of the geometric data from a CAD application, like AutoCAD, is done by creating a temporary file containing this information. This file is read by the acquisition module into the software tool and all parts are visualized as objects. In the next step these objects are used to generate the building model's structure using the level of details described in Section 3.2 and shown in Figure 2.

The building's complete geometry representation is stored in the database. The representation of single building elements and the associated objects are linked by the attribute GEOMETRY as shown in Figure 3.

4.4 *Implementation of the PLMS*

Acquisition module. Using the acquisition module shown in Figure 7, all information about a building is entered. Beginning with level 1 (see Figure 2) the building structure in the software tool is generated step by step. For every new object the appropriate input mask is generated using the information given by the assigned class. This meta-information is read from the database. The complete acquisition is done - as everything else - using the GUI. During the acquisition process, general and geometry based data, such as material, concrete cover, environmental loads, is entered and stored in the database. An example for the input mask for general information is shown in Figure 6, for geometry based information in Figure 10.

For speeding up data acquisition, extendable lists for special inputs have been integrated into the software tool. Those lists are available for the following properties: materials, defined material parameters, environmental loads, repair measures, deterioration models, and types of building elements. Using these lists to create input masks for the software tool we can achieve a highly effective workflow. The lists are stored in the database and can be updated by the user or the administrator, respectively. For example, it is possible to define an arbitrary number of environmental loads, each of which is defined by a number of parameters. This functionality for defining environmental loads renders the systems extremely flexible, which is necessary to ensure its usability over the long life-span of a concrete building.

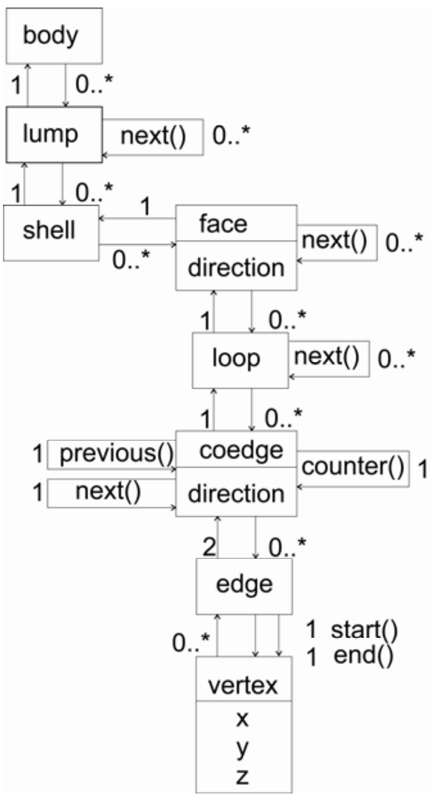

Figure 9. Geometry schema in use by the PLMS.

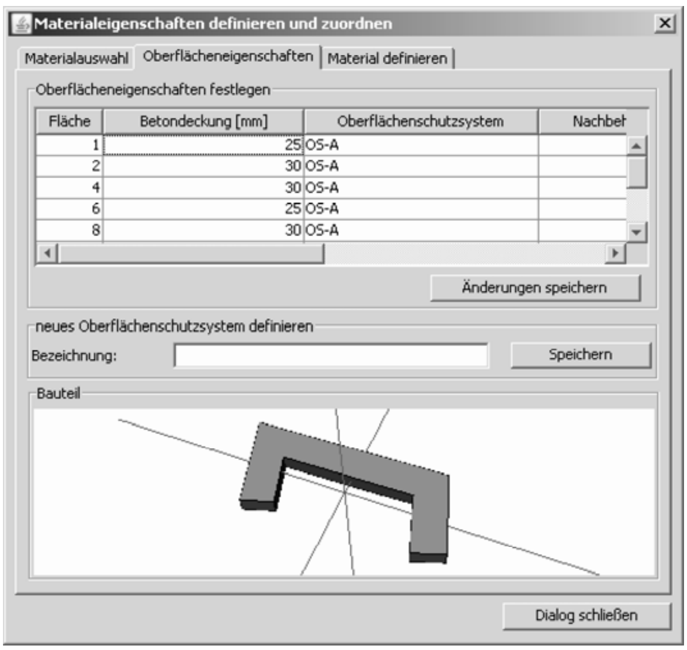

Figure 10. Assigning information to faces.

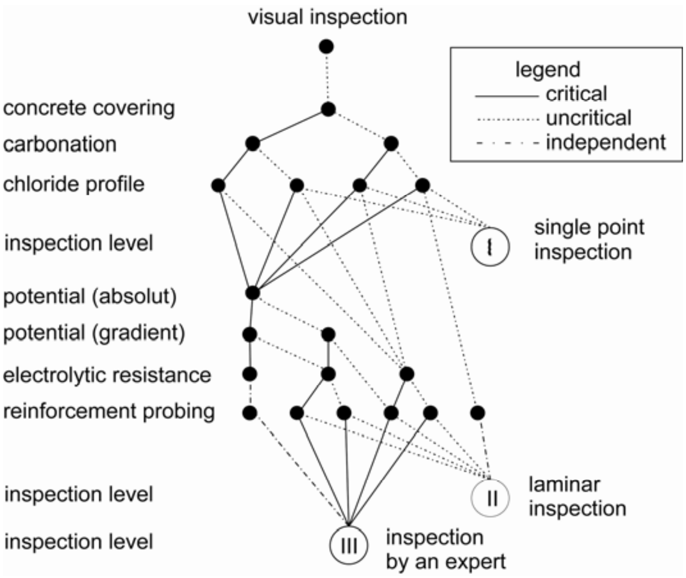

Figure 11. Decision tree for choosing the required inspection level.

Condition Acquisition module. Using this module, the results from all kinds of measurements are assigned to the appropriate element's face. Which kind of measurements should be used will also be decided within this module. The number of required measurement methods depends on the part's condition and its deterioration mechanism. A decision tree is used for choosing the right methods. In the framework of this research project, three inspection stages (I to III) are being developed. In Figure 11 the decision tree in the case of reinforcement corrosion is shown. The three stages are defined as following: single pointed measures (stage I), laminar measures (stage II), and measures carried out by an expert (stage III).

All inspection results are also stored in the database. To this end, there are a lot of different discontinuous measurement methods and sensors for continuous monitoring available, consequently, another meta-model for measurement methods and measurements will be introduced. This one will be designed in a similar manner to the one already presented in Figure 3. The main difference will be that the generic semantic model is used to describe measurement methods and sensors, and on the right hand the semantic data is used for the applied measures including date and time information.

Prognosis module. For computing the prognoses the prognosis module needs all information assigned to the building elements. In the first step the bridge manager must choose the appropriate deterioration mechanism for every part, if necessary. It is not possible to do this automatically because lots of expert knowledge is needed to select the right deterioration mechanism. Predicting the progressive decline (of a structure's condition) will be probabilistically calculated using the software package STRUREL (RCP 2007). We intend to establish a corresponding interface between the software package STRUREL and the PLMS to enable the required data exchange.

Assessment and repair module. We have not yet begun implementing these two modules.

5 SUMMARY

In an ongoing research project, we are developing a software system for the predictive life-cycle management of reinforced concrete bridges. A key feature of the PLMS is a 3D building information model which forms the basis of all data acquisition and evaluation functionality. This model serves to store all information on building elements together with their relation to the geometry. The model provides multiple levels of detail and means to associate semantic classes with geometric objects. Since the PLMS is designed for different kinds of building types, an explicitly available meta-model has been integrated which is used to generate a specific BIM.

What first distinguishes this software tool from other existing building management systems is the fact that a construction is subdivided into up to five levels (of detail). In an initial step, structures are first subdivided into modules and then into building elements. Component parts are subdivided into subelements and hotspots. The advantage of this approach is that results of inspections or photos can be directly allotted to the corresponding geometries. This subdivision is necessary in order to make use of fully probabilistic deterioration models.

For predicting future changes in the condition of a structure, these fully probabilistic deterioration models are used in conjunction with non-destructive inspection methods. The construction's condition is computed by aggregating the conditions over all five levels. It is accordingly possible to detect damage at a very early stage and plan repair measures to eliminate such damage. This consequently means a reduction in the financial outlay for the maintenance of the structure.

The predictive life-cycle management system is implemented by coupling a Java application with a relational MySQL database. The GUI provides a simple, innovative way to store new constructions in the database. All information can be entered and modified using this interface. There are special requirements for the database and the GUI by using the meta-model, which are extensively discussed in this paper.

REFERENCES

Borrmann, A., Wenisch, P., van Treeck, C., and Rank, E. (2006). Collaborative computational steering: Principles and application in HVAC layout. *Integrated Computer-Aided Engineering 13*(4), 361–376.

Duffy, L. (2004). *Development of Eirspan: Ireland's bridge management system*. Proceedings of the Institution of Civil Engineering 157: 139–146.

Eastman, C. (1999). *Building Product Models: Computer Environments Supporting Design and Construction*. Boca Raton, Florida: CRC Press LLC.

Gehlen, C. (2000). *Probabilistische Lebensdauerbemessung von Stahlbetonbauwerken - Zuverlässigkeitsbetrachtungen zur wirksamen Vermeidung von Bewehrungskorrosion*. Berlin: Deutscher Ausschuss für Stahlbetonbau (DAfStb), Heft 510.

Haardt, P. (2002). *Entwicklung eines Bauwerks-Management-Systems für das deutsche Fernstraßennetz, Stufe 1 und 2, Schlussbericht zum AP-Projekt 99 245*. Bergisch-Gladbach: Bundesanstalt für das Straßenwesen.

Haller, H. and Basurco, G. (2006). *KUBA-DB Benutzerhandbuch*. Bern: Eidg. Drucksachen- und Materialzentrale.

Hauschild, T., Borrmann, A., and Hübler, R. (2003). Techniken der Verwaltung dynamischer digitaler Bauwerksmodelle. In *Tagungsband des Int. Kolloquiums über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen (IKM)*.

Hammad, A., Zhang, C., and Hu, Y. (2006). *Mobile Model-Based Bridge Life Cycle Management*. Proceedings of the 8th International Bridge Management Conference: I–5/1–13.

Hawk, H. (1999). *BRIDGIT: User-Friendly Approach to Bridge Management*. Proceedings of the 8th International Bridge Management Conference: E–7/1–15.

Henriksen, A. (1999). Bridge Management - Routine Maintenance: Recent Experience with the Routine Management Module in the DANBRO Bridge Management System. In *Proceedings of the 8th International Bridge Management Conference: I–5/1–13*.

Kowalczyk, W. (1997). *Ein interaktiver Modellierer für evolutionäre Produktmodelle*. Ph. D. thesis, Lehrstuhl für Bauinformatik, Technische Universität München.

Lair, J., Rissanen, T. and Sarja, A. (2003). *Methods for Optimisation and Decision Making in Lifetime Management of Structures*. LIFECON Deliverable D 2.3.

Mayer, T., Schiessl, P., and Zintel, M. (2008). *Birth Certificate as an Important Tool for Public-Private-Partnership Projects*. Proceedings of the 1st International Symposium on Life-Cycle Civil Engineering (IALCCE'08), Varenna 2008.

MySQL (2007). *Documentation of MySQL Ed. 1.5*. Uppsala: MySQL AB.

Norris, G. and Marshall, H. (1995). *Multiattribute Decision Analysis Method for Evaluating Buildings and Building Systems*. Gaithersburg: US Department of Commerce NISTIR 5663.

RCP (2007). *Documentation of STRUREL Ed. 8*. Munich: RCP.

Robert, W., Marshall, A., Shepard, R., and Aldayuz, J. (2003). *Pontis Bridge Management System - State of Practice in Implementation and Development*. Proceedings of the 9th International Bridge Management Conference: 49–60.

Schiessl, P., and Mayer, T. (2006). *Lebensdauermanagement von Stahlbetonbauwerken*. Statusseminar zum Verbundforschungsvorhaben "Nachhaltig Bauen mit Beton", 65–74.

Schiessl, P., and Mayer, T. (2007). *Forschungsbericht 31006/05, DAfStb-Verbundforschungsvorhaben "Nachhaltig Bauen mit Beton", Teilprojekt A3 "Wesentliche Bausteine eines Bauwerksmanagementsystems"*. Berlin: Deutscher Ausschuss für Stahlbetonbau (DAfStb), Heft 572.

Spatial (2006). *Documentation of 3D ACIS Modeler*. Westminster: Spatial.

Steinmann, F. (1997). *Modellbildung und computergestütztes Modellieren in frühen Phasen des architektonischen Designs*. Ph. D. thesis, Bauhaus-Universität Weimar.

Sun Microsystems (2006a). *Documentation of Java™Platform Standard Ed. 6*. Santa Clara: Sun Microsystems.

Sun Microsystems (2006b). *Documentation of Java3D Ed. 1.5.1*. Santa Clara: Sun Microsystems.

Thompson, P., Merlo, T, Kerr, B., Cheetham, A., and Ellis, R. (1999). *The New Ontario Briddge Management System*. Proceedings of the 8th International Bridge Management Conference: F–6/1–15.

Vesikari, E. (2006). *BRIDGELIFE, User Manual*. VTT, Finland.