BIM-Based Life-Cycle Management for Reinforced Concrete Buildings

André Borrmann, Technische Universität München, Germany
Katharina Lukas, Technische Universität München, Germany
Marc Zintel, Technische Universität München, Germany
Peter Schießl, Ingenieurbüro Schießl Gehlen Sodeikat, Germany
Michael Kluth, Consultant, Germany

ABSTRACT

This paper introduces the concept of a BIM-based life-cycle management system for reinforced concrete buildings. The system allows one to compute a prognosis of the building’s condition taking into account the material properties of individual components, the environmental load as well as measurement data from current inspections. This prognosis then forms foundations for scheduled maintenance and repair actions in an economically efficient way. A particularly important feature of the presented system is that all input data as well as the computational results are associated with a (full) 3D Building Information Model (BIM) of the construction. In this way, an easy localization of the information is achieved facilitating both the data collection and the estimation of the building condition for engineers involved in inspection planning, inspection or the scheduling of repair actions. To further facilitate data input and interpretation, a hierarchic level-of-detail approach is employed for structuring the building model, ranging from building level down to individual hot spots. Additionally, the integration of a meta-model allows the flexible adaption of the semantic data model to specific buildings types or the particular needs of the users.

Keywords: 3D Model, Building Information Modelling, Concrete Structures, Deterioration, Inspection, Level-of-Detail, Life-Cycle Management, Measurement Data

1. INTRODUCTION

In most industrialized countries, large parts of the infrastructure were erected during the 1960s and 1970s. As a result, they are now facing an increasingly aging stock of infrastructure buildings. To keep the infrastructure safe, while at the same time keep the impact on public budget at a tolerable level, an elaborate management scheme for these buildings, planning their inspections, and their maintenance and repairs, is necessary.

DOI: 10.4018/ij3dim.2012010101
Conventionally, maintenance planning is performed manually. In recent years however, computer-aided life-cycle management systems (LMS) have been developed to support the maintenance planner in a suitable way. The most basic LMS supports maintenance planning by capturing inspection data and assigning grades according to the condition to buildings or individual components. More advanced systems perform a deterioration prognosis which enables the user to simulate the future condition of the building and evaluate different maintenance strategies. So far, the user interfaces of these systems have been mainly restricted to simple lists containing numerous building components, their individual exposure and current condition, rendering it difficult to keep an overview and identify distinct hotspots.

This paper introduces the concept of a life-cycle management system which is based on a building information model. A building information model is a digital representation of a real building which comprises of not only the detailed 3D geometry but also a semantic description of its individual components and their relationships (Eastman et al., 2008; Underwood & Isikdag, 2009). The concept of using Building Information Modelling (BIM) during the planning phase has gained increasing importance during the last few years. This paper shows how BIM can also bring severe benefit to the operational phase if it is used as the basis of a life-cycle management system. A 3D building information model allows one to easily assign material properties, environmental loads as well as inspection and measurement results to the individual components of the building. They are then unambiguously located in/on the virtual building and can form the basis for a detailed life-cycle prognosis and maintenance planning.

Due to the very flexible handling of non-geometric data, the developed system can be used for any type of construction made from reinforced concrete, including bridges and high-rise buildings.

2. RELATED WORK

There have been several life-cycle management systems for bridges and reinforced concrete buildings developed over the last few years. Examples include SIB-Bauwerke in Germany (Abram, 2003), KUBA-MS in Switzerland (Haller & Bascuro, 2006), DANBRO in Denmark (Henriksen, 1999), Eirspan in Ireland (Duffy, 2004), BridgeLife, Maintenance-Man and ServiceMan in Finland (Vesikari, 2006, 2008), Pontis (Robert et al., 2003) and BRIDGIT (Hawk, 1999) in the USA and the Ontario Bridge Management System in Canada (Thompson et al., 1999). There has also been high research activity over the last few years aimed at the development of more advanced life-cycle management systems (e.g., Frangopol et al., 2001; Frangopol & Neves, 2003; Neves et al., 2006; Hammad et al., 2006; Okasha & Frangopol, 2010).

In principal the existent systems can be categorized into four groups:

• Systems for mere data management which are not able to compute prognoses (e.g., Henriksen, 1999; Duffy, 2004). These systems store the data gained at routine inspections as well as the reports of maintenance carried out. Both DANBRO and Eirspan also have functionalities to estimate maintenance costs, Eirspan can also optimise the maintenance schedule in order to minimize the costs produced by the maintenance as well as by postponing it. The development of the condition cannot be estimated.

• Systems with deterministic deterioration models (e.g., Abram, 2003). In these systems a condition prognosis can be achieved on the basis of deterministic models. Different parts of the bridge are each assigned a lifetime, after which they have to be replaced. The system supports the user by listing repair measurements to be performed.
• Systems with probabilistic deterioration models based on Markovian chains (e.g., Vesikari, 2006, 2008). In these systems the probability of failure at a given time in the building’s lifecycle can be computed. This computation is done by Markovian chains, i.e., the failure probability in one year depends on the probability in the year before. The transition probabilities that form the core of the model are based on experience.

• Systems with more complex probabilistic deterioration models (e.g., Frangopol et al., 2001; Frangopol & Neves; 2003, Neves et al., 2006; Hammad et al., 2006; Okasha & Frangopol, 2010). In these systems the probability of failure is computed by probabilistic techniques, e.g., Monte Carlo Simulation. The probabilities are based on the history of the bridge and its traffic loads.

Almost all the systems described here are solely based on textual descriptions of the buildings. Some of them (e.g., SIB-Bauwerke and Eirspan) allow the attachment of photographs to illustrate inspection data. Only the system described by Hammad et al. (2006) makes use of a 3D geometric model of the building to assist the user’s orientation.

### 3. ARCHITECTURE OF THE BIM-BASED LIFE-CYCLE MANAGEMENT SYSTEM

The architecture of the developed life-cycle management system is depicted in Figure 1. The system 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 material 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 the structuring of the building and the assigning of 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 or building’s elements 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 the case of an inspection, the module assists in capturing newly found cracks by, for example, providing means to “draw” them or to stick digital photos on the respective surface in the 3D model.

Results from measurements carried out during an inspection (such as concrete cover mapping, carbonation depth or depassivation depth, and chloride profiles) are also stored in the LMS by means of the condition acquisition module. Furthermore, the module 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 actual 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 LMS using the repair module along with the building’s component condition after these repair measures is stored. This information is used by the prognosis module to compute a Bayesian update (Brand & Small, 1995) for all condition prognoses.

After a building’s completion and handover, the LMS 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 was obtained by using traditional 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)

### 4. The Hierarchical Building Information Model

#### 4.1 3D Building Geometry as the Data Management Basis

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

The use of a 3D building model is therefore propagated as the centre for 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

---

*Figure 1. The modules of the predictive life-cycle management system (adapted from Schiessl & Mayer, 2007)*

[Diagram of the modules of the predictive life-cycle management system]
representation of the corresponding building component. Using a 3D model, all relevant information can be easily allocated and a very good overview is guaranteed.

4.2 Multi-Level Semantical Model

A purely geometric model would not fulfill the demands on a LMS as 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 modelling approach. This is called ‘reverse’ since in the standard product modelling process the semantics of an object are first defined, and based on that, a geometric representation is assigned (Eastman, 1999).

Additionally, a multi-level, hierarchical model is realized. The proposed hierarchy consists of five different levels with each entity on one level having 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. The exception is for the last level where all child elements together form the parent element – a typical aggregation relationship (Figure 2).

The first level represents entire buildings. Since the LMS 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 of modules. Modules are groups of building components with identical functionalities. 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 in order 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.

Figure 2. The hierarchical Building Information Model in use for the LMS 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 (adapted from Schiessl & Mayer, 2007)
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 in parts of the 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 pre-stressing 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.

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 call for immediate remedial actions independent of the aggregated state of the whole structure.

4.3 Meta-Model Integration

Since the developed life cycle management system should not be restricted to the maintenance of a specific type of structure, but instead be applicable for a wide variety of building types, a functionality has been integrated that allows to dynamically adapt the available semantic data structures. This has been realized by means of the explicitly available meta-model depicted in Figure 3: On the left hand side the meta-classes CLASS, ATTRIBUTE and ASSOCIATION are depicted. 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, a class named Bridge would be modelled 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, more complex attributes often have to be modelled. For example, it might be necessary to store not only the name of the owner as a simple string, but instead use a complex Owner class comprising of 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-
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.

On the right of Figure 3 the meta-classes are shown that are capable of holding 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 side. Since all input masks are generated from the domain model defined through the meta-model (Figure 6), the input data automatically conforms to the desired data structure. However, even without the user.

Person and Figure 5 shows the corresponding object diagram. In the concept, only specialists (e.g., system administrators) should have access to the data modelling facilities, since changes in the data structures have a crucial impact on the functions of the whole system. Normal users such as bridge managers, inspectors or regulation authorities will work with input masks that are directly generated from the data model. They will not be able to change the data model.
The concept of a building information model that is on the one hand centred on geometry and on the other 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 modelling in early design phases (Steinmann, 1997; Kowalczyk, 1997), in modelling existing buildings for revivification measures (Hauschild et al., 2002), and for the realization of a Collaborative Computational Steering system (Borrmann et al., 2006).

5 PROBABILISTIC CONDITION PROGNOSIS

In the developed life-cycle management system, the condition prognosis is performed in a fully probabilistic manner. Most of the life-cycle management systems currently in use realize the condition prognosis based on deterministic deterioration models (e.g., Abram, 2003). Only Maintenance Man and Service Man (Vesikari, 2006, 2008) are using Markovian Chains to model uncertainties.
Recently, a number of research projects have been conducted aimed at developing sophisticated probabilistic deterioration models (e.g., Frangopol et al., 2001; Frangopol & Neves, 2003; Neves et al., 2006). Okasha and Frangopol (2010) further upgrade these models by including traffic loads in the computation. However, in these models the deterioration process is only dependent on the maintenance dates and the type of maintenance carried out. Effects by aggressive environments, e.g., de-icing salts, direct rain, etc. are not taken into consideration. Also, using these models it is not possible to update the prognosis when newly gained condition data becomes available, such as data from inspections, for example.

As in the approach presented, the condition prognosis is performed for each individual surface separately therefore it is possible to consider their different environmental loads: the inner surface of the wall in a parking deck has different environmental conditions than the outer surface that may be subject to rainfall and thus will deteriorate much faster. It is also possible to consider data obtained by inspections (e.g., chloride profiles, carbonation depths, etc.) and accordingly achieve a more accurate prognosis.

5.1 Applied Safety Concept

Our prognosis computation is based on the safety concept of the Eurocode for structural design (Mosley et al., 2007). This concept is illustrated in Figure 7. The basic principle is to compare the resistance of a structure $R$ with the applied loads $S$. As the exact values for both $R$ and $S$ are not known (both can vary for example due to workmanship, etc.) both cannot be handled in a deterministic way but as probabilistic distribution functions. So the dif-

---

*Figure 6. Automatically generated dialog box*
ference of $R$ and $S$, the reliability $Z$, is also a distribution function. The probability of failure $p_f$ is the probability that $R$ is lower than $S$, that is $\int_{-\infty}^{0} Zdz$. The structure is safe if $p_f$ is lower than some maximum acceptable failure probability.

In the case of a condition prognosis, $R$ depends mainly on material properties (e.g., type of cement), and $S$ depends on the environmental loads described by exposure conditions according to German norm (DIN 1045, 2008). Both depend on time, i.e., the older a building gets, the higher is the probability of its failure.

5.2 Deterioration Model

Gehlen (2000) and FIB (2006) describe a model based on Fick’s first law of diffusion (Philibert, 2005) to describe the progress of the depassivation front for both chloride and carbonation induced depassivation of the reinforcement inside the concrete, using probabilistic concepts. The velocity of the depassivation front in this model is dependent on material properties and environmental loads.

For the deterioration due to chloride and carbonation induced depassivation of the reinforcement each of the six condition states have been defined (Figure 8).

A new building (and its surfaces) is in condition state 1. It changes to condition state 2 as soon as the depassivation front of Chloride and Carbon respectively reaches the reinforcement. The beginning of corrosion marks the transition to state 3. State 4 and 5 describe failures that are noticeable on the surface, i.e., cracks respectively spalling. State 6 marks the ultimate limit state (ULS) when the buildings safety is no longer guaranteed.

For each condition change a probability can be computed based on the progression of the depassivation front: The condition state changes from 1 to 2 when the depassivation front has reached 2/3rds of the concrete cover. State 3 occurs when the front reaches the reinforcement. The remaining state changes are computed by adding time constants.

When the probabilities of the condition state changes are known for each year, the probabilities of each condition state can be computed. The condition state with the highest probability is assumed for the respective surface.

For security reasons, a threshold probability $p_t$ (e.g., 10%) can be set for low condition states (4 to 6): as soon as $p_t$ is reached for one of these condition states this condition state will
be assumed even if a better state has a higher probability.

Data gained by inspections about the current position of the depassivation front can be added to the prognosis computation by Bayesian Update (Brand & Small, 1995). So the uncertainties in the model that are high in the beginning can be minimized over the building’s lifetime.

Similar models for other deterioration mechanisms like freezing (Fagerlund, 2004), freezing in combination with de-icing salts (Lowke & Brandes, 2008) and sulphate (Müller et al., 2009) are under development.

6 PROTOTYPICAL IMPLEMENTATION

6.1 General Information

The software tool presented in this paper is implemented by coupling a Java application with a relational database management system (DBMS). All construction information, including geometric data is stored in this database. The integration of a DBMS is necessary to provide

- Persistency of the life-cycle management data,
- Fast access to the data,
- The possibility of storing large amounts of data.

For the prototype implementation, the relational DBMS MySQL was chosen. The three-dimensional representation of the construction’s geometry is realized by means of the Java 3D library (Sun Microsystems, 2006).

A screenshot of the software system is shown in Figure 9.

6.2 Implementation and Use of the Meta-Model

Realizing the aforementioned adaptable semantic model needs special requirements for the database and the graphical user interface (GUI) to be implemented. 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

Figure 8. Condition states for deterioration due to reinforcement depassivation
class can have an arbitrary number of attributes of each type, additional link relations are used to model these m:n relations (Figure 10).

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 in 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 are automatically generated during usage of the software tool (Figure 6). 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.

The following input components are automatically created by the software tool: If the user creates a new building (level 1) they have 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 (Section 4). To realize this, they create instances of the available classes and assigns one or multiple geometric objects to each of them. The user is supported during this process with a visual distinction between classified and non-classified building elements in the 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

---

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
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 using the GUI. An essential aspect of this GUI is that an intuitive and user-friendly interaction is possible both at an office desk as well as during a bridge inspection.

### 6.3 Management of 3D Geometry

The geometric representation of the building is visualized by means of the Java3D library (Sun Microsystems, 2006) (Figure 9). Using this library the user can interactively translate, rotate, or scale the model. Furthermore, a single object can be marked by a mouse click; the right-click displays the properties of this element.

The software prototype uses a geometry structure very similar to the geometry kernel ACIS (Spatial, 2006), (Figure 11). The advantages of such a structure are:

- All building elements are represented by faces. Since almost all information in this tool is stored geometry-based, it is very important to have direct access to all faces.
- The structure allows for the description of building elements with holes, like for example, box-section slabs.
In order to be able to export geometry data created in Autodesk AutoCAD, a module has been realized that creates an intermediate file containing all geometry information. This file is fed into the software system by the acquisition module. As soon as this import process is finished, all geometric entities are visualized as 3D objects. In the next step these objects are used to generate the building model’s structure using the level of details described in Section 4 and shown in Figure 2.

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

6.4 Implementation of Individual Modules

6.4.1 Acquisition Module

By means of the acquisition module (Figure 12), the user can add a new construction to the system. In a first step, the 3D geometry of the building modelled by means of an external CAD program is imported. Subsequently, the user structures the building according to the five levels-of-detail introduced in Section 4 and adds additional data such as material properties, construction dates, environmental loads and so on. They are assisted by both the 3D representation of the building’s geometry as well as a number of input masks. For every new object the appropriate input mask is generated using the meta-information associated with the assigned class (Figure 14). An example for the input mask for general information is shown in Figure 6 and for geometry based information is shown in Figure 13.

To speed up data acquisition, extendable lists for special inputs have been integrated into the software tool. These 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 a highly effective workflow can be achieved. 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

Figure 12. Graphical user interface of the acquisition module
Figure 13. Assigning material information to individual faces of a building element

![Figure 13](image13.png)

Figure 14. Assigning exposition classes to individual faces of a building element

![Figure 14](image14.png)
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.

6.4.2 Condition Acquisition Module

Using the condition acquisition module (Figure 15), inspection data is added to the building information model. The system supports the management of data from both destructive and non-destructive inspection techniques (chloride profiles, carbonation depths, etc.).

The inspection results are pre-processed and statistically evaluated in such a way that the user only has to enter a distribution type and the respective distribution parameters. This information is required for the successive prognosis computation. Similar to the acquisition module (Figure 12), a 3D representation of the geometry assists the correct localisation of the data. In most cases inspection data is assigned to individual faces of building components.

Also, the module helps in taking the decision of which kind of measurements should be used. The number of required measurement methods depends on the part’s condition and its deterioration mechanism. A decision tree is used to choose the right methods. In the framework of this research project, three inspection stages (I to III) are being developed. In Figure 16 the decision tree in the case of reinforcement corrosion is shown. The three stages are defined as follows: single pointed measures (stage I), laminar measures (stage II), and measures carried out by an expert (stage III).

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

6.4.3 Prognosis Module

In the prognosis module the condition prognosis over the building’s lifetime is computed according to the probabilistic concepts introduced in Section 5. The deterioration models we are using describe the condition on a scale of six condition states (Figure 8). In the software system, the main deterioration processes; chloride induced depassivation and carbonation induced depassivation have been implemented. However, it can easily be extended to include other deterioration processes.

The probabilistic prognosis computation is realized using the software package STRUREL (RCP, 2010). STRUREL has originally been developed to perform probabilistic computations for structural analysis. However, as the limit functions can be defined freely by the user, the program can also be used for reliability studies in general.

The communication between the LMS and STRUREL is realized by file exchange. The LMS exports the deterioration model containing distribution functions for all parameters needed for the computation. STRUREL returns the probability of transition to the different condition states over the building’s lifetime. Based on these results the probability of the building to be in a certain state at each year of the lifetime is computed and stored in the central database.

6.4.4 Assessment Module

The assessment module visualizes the condition state of the building (Figure 17). The user can choose the level of detail (surfaces, volume elements – building elements, element parts or Hot Spots – or the whole building) for which the condition prognosis shall be computed and visualised. Only the condition indices of the lowest level (surfaces) are stored in the database, for higher levels (building elements, modules, the entire building) the condition states (e.g., reliabilities) are computed by aggregation.
Figure 15. Graphical user interface of the condition acquisition module

Figure 16. Decision tree for choosing the required inspection level
The smallest unit within the acquisition of structures is a single surface. For instance, one surface can be one side of the six sides of a column. Therefore one element comprises of several surfaces and one building comprises of several elements again and so on. To generate a condition state at building level (which is called the Building Condition Index BCI), the condition states of all the corresponding elements need to be computed (Condition Index on element level CI element).

In order to weight different elements to their static and safety relevance, it is necessary to introduce weight-factors \( w_i \), which allow a functional assessment of each type of element.

How the aggregation of a condition state is computed from surface level up to element level is shown in Equation (1) wherein the \( CI_{element} \) is displayed.

\[
CI_{element} = \frac{1}{\sum_{i=1}^{n} A_i \cdot N_i \cdot w_i}
\]  

\( A_i \): Area of a single surface \( i \) [m²]  
\( N_i \): Condition state of surface \( i \) out of full-probabilistic deterioration modeling [-]  
\( w_{ei} \): Weight-factor for surface \( i \) [-]

In principle, further aggregation of condition states from the element level up to the building level can be developed analogically, but then an amplified weighting of high Condition Indices at the element level has to be considered additionally.

6.4.5 Repair Module

Whenever repair actions are taken they have to be recorded in the LMS. This is done using the repair module which provides a catalogue of repair measures for the user to choose from. A maintenance measure either resets the condition to a higher level (e.g., replacement, Figure 18) or slows down the deterioration process (e.g., coating of the surface). In both cases, a new condition prognosis has to be computed starting in the year of the maintenance. The new prognosis is again fed into the database.

7 CASE STUDY

Among other scenarios, the developed life-cycle management system has been tested on the section of a parking deck shown in Figure 19. The floor plates and the bases of columns and walls are subject to de-icing salts and therefore to chloride induced depassivation (German exposure condition XD3, DIN 1045, 2008) while the upper parts of walls and columns
are only subject to carbonation (XC3, DIN 1045, 2008).

All parts of the model belong to the same module (compare Section 4). The building elements are floor plates, columns, walls and girders. Columns and walls are further subdivided into the element parts base and upper part. To allow for the different conditions at parking and driving areas the monolithic floor plate in the model is structured into several building elements.

The structuring of the building can be seen in Figure 19. Besides, the position in the hierarchical model and the environmental loads material properties (water/cement ratio, strength category) are assigned to each building element.
In Figure 20 the prognosis for the upper surface of the floor plate in the driving area over six years is shown. In the first year the surface is in condition 1 with 35% probability, in condition 2 with 33% probability, and in condition 3 with 24% probability. Over the years the building degrades, the probabilities of the better condition indices get lower, the probabilities of the worse indices rise.

For the “good” condition indices 1 to 3, the assessment module assumes the index with the highest probability. Accordingly, for year 1 in Figure 20 the surface has the condition index 1. However, as soon as the probability of one of the “bad” indices 4 to 6 gets higher than some threshold value (here 10%) this condition index is assumed. Accordingly, the condition index jumps to 4 in year 2 of the examples.

The condition prognosis is plotted in Figure 21. The solid red line shows the condition trend over the first 30 years of the plate surface without maintenance. Planned maintenance work changes this prognosis: Surface coating “freezes” the condition index from the time of maintenance until the coating itself has corroded. The green dot-and-dash line in Figure 21 shows the condition of the surface if in year 1 surface coating is applied. For the time of the coating corrosion the surface stays in condition 1; after this the degradation follows the same curve as without maintenance.

A replacement of the concrete resets the condition of the surface to the starting point. The blue dashed line in Figure 21 shows this for a replacement done in year 7.

In year 20, chloride measurements at seven different spots of the floor plates are taken (Table 1). It is assumed that the chloride is normally distributed. With the values from Table 1 the mean value of the distribution is 1.19, the standard deviation is 0.28. With the dialog shown in Figure 15 these results are stored in the database. In a new prognosis those values are taken into account by means of a Bayesian Update.

### 8 CONCLUSION AND FUTURE WORK

This paper introduced the concept of a BIM-based life-cycle management system. The use of a 3D building model makes the system more transparent and facilitates the localization of input data and results. It also simplifies the transfer of responsibility, which occurs relatively often in the life-cycle management of constructions due to the long lifetime of buildings.

The hierarchical level-of-detail structure of the building with two sub-element levels allows for a more detailed view of the building. Using the hierarchy, areas of high risk can be given closer attention and a detailed localization of only local deteriorations is possible. The five-level approach also supports the probabilistic condition prognosis, since it allows one to assign different environmental loads to individual building elements or even single surfaces. The probabilistic deterioration model used here con-

---

**Figure 20. Prognosis for floor plate in driving area**

<table>
<thead>
<tr>
<th>Year</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.35</td>
<td>0.33</td>
<td>0.24</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.32</td>
<td>0.32</td>
<td>0.22</td>
<td>0.13</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.30</td>
<td>0.31</td>
<td>0.19</td>
<td>0.18</td>
<td>0.02</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>0.28</td>
<td>0.30</td>
<td>0.17</td>
<td>0.21</td>
<td>0.04</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0.26</td>
<td>0.29</td>
<td>0.15</td>
<td>0.22</td>
<td>0.08</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>0.25</td>
<td>0.28</td>
<td>0.13</td>
<td>0.22</td>
<td>0.12</td>
<td>0.00</td>
</tr>
</tbody>
</table>

...
siders those different environmental settings and thus can predict at which places deteriorations and damages can be expected to occur first.

One of the limitations of the presented approach is that a 3D model of the building under consideration must be available, which is rarely the case particularly for older buildings (Arayici, 2008). Future work will therefore focus on an automated generation of simplified 3D building models based on an extensive use of parametric technologies (Ji et al., 2011).

At the moment only models for Chloride or Carbon induced depassivation of the reinforcement exist. For a practical use it is necessary to develop similar models for other deterioration mechanisms, e.g., the combination of frost and de-icing salts or mechanical use and first steps in this direction have been taken (e.g., Fagerlund, 2004; Lowke & Brandes, 2008; Müllauer et al., 2009).

A further step in the development of an advanced life-cycle management system will be

**Table 1. Chloride measurements after 20 years**

<table>
<thead>
<tr>
<th>Measurement Spot</th>
<th>Chloride in 15mm [M-%/z]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.25</td>
</tr>
<tr>
<td>2</td>
<td>1.60</td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
</tr>
<tr>
<td>4</td>
<td>1.10</td>
</tr>
<tr>
<td>5</td>
<td>0.98</td>
</tr>
<tr>
<td>6</td>
<td>1.42</td>
</tr>
<tr>
<td>7</td>
<td>1.26</td>
</tr>
</tbody>
</table>
the consideration of a super-building network level where all buildings under one administration are considered. Here the objective is to find an ideal schedule for maintenance subject to budget constraints and a preferably low impact on traffic flow (Lukas et al., 2010; Lukas & Borrmann, 2011).

REFERENCES


RCP. (2010). Documentation of STRUREL. Munich, Germany: RCP.


André Borrmann received the diploma degree in civil engineering from Bauhaus University Weimar in 2003, and the PhD degree from Technische Universität München for his work on a spatial query language for building information models in 2007. He continued working at Technische Universität München as leader of the research group “Construction Informatics”, and was appointed as professor for computational modeling and simulation in 2011. His research interests include building information modeling and processing, modeling of built infrastructure, construction process simulation and optimization, computational life-cycle management, and pedestrian dynamics. He is involved in various research projects and acts as reviewer for a large number of international scientific journals.

Katharina Lukas received the diploma degree in civil engineering from Technische Universität München in 2007. From 2007 to 2008 she worked as structural engineer for railway sleepers. In 2008 she started her PhD studies at the chair for computation in engineering at Technische Universität München and changed 2011 in the newly-founded Computational Modeling and Simulation Group. Her research interests include computational life-cycle management, building information modeling, and schedule optimization.

Marc Zintel received the diploma degree in building materials engineering from Technische Universität München in 2007. Since then he works as research scientist at the Centre for Building Materials at Technische Universität München. His research interests include Epoxy coated concrete reinforcement, life-cycle management for reinforced concrete buildings, optimatized inspection strategies for reinforced concrete buildings, glued and mechanical conjunctions for reinforcement steel, and chemical additives as corrosion inhibitors.

Peter Schießl received the diploma degree in civil engineering from Technische Universität München in 1967, and the PhD degree from the same university for his work on influence of concrete cover and crack widths on corrosion of steel in concrete in 1973. From 1973 to 1985 he was head of the department “Practical Application” at the Institute for Reinforcing Steel in Munich, Germany. From 1985 to 1998 he was professor at the chair for Building Materials at RWTH Aachen and director of the Institute for Building at the same university. From 1998 to 2008 he was professor at the chair for Building Materials and Material Testing and director of the Centre for Building Materials at Technische Universität München. Since 2000 he is partner in the Prof. Schießl Engineering Office, Munich, Germany and since 2007 share holder of the Schießl Gehlen Sodeikat GmbH Engineering Office.

Michael Kluth was born in 1977. After school he did a apprenticeship in the steel working industry, beginning in 1997. In 2000 he started his studies in Civil Engineering at Technische Universität Darmstadt which he completed with diploma degree in 2006. From 2006 to 2007 he worked at a company constructing roads and pipelines. In 2007 he started as research assistant at the chair for computation in engineering at Technische Universität München where he worked on the development of a software tool for the life-cycle management of reinforced concrete buildings. In 2008 he left the chair and started working at a BIM-software company as support engineer.