5 Experimental Results[1]

BIM aims to support the entire life-cycle of buildings and structures. Damage information is mandatory during bridges' operation phase because every national guideline for inspection and assessment relies on defect information. Hence, beside generating accurate as-built models of existing buildings or structures, it is mandatory to enhance BIM with damage information in order to make it usable for the operation phase. Numerous studies focus on BIM in the context of design. The operation phase has so far been considered only sparsely. Sacks, Kedar, Borrmann, *et al.* have come to a similar conclusion as they emphasized that a damage representation is currently missing [39]. This dissertation addresses the lack of such a damage representation. Whereby, the majority of the publications propose closed frameworks for inspection automation, for example, automatizing damage registration [57], [58], [60], [75], [78], [89], [147], and less effort is invested in developing a data model to ease collaborative work of different stakeholders during the operating phase. Existing approaches either lack implementations and testing or flexibility in supporting different stakeholders [91], [100]. By considering multiple use cases, such as inspection, structural analysis, and assessment, a comprehensive information model for the operating phase has been provided.

BIM defines a digital information model to reduce costs and information loss during constructions' life-cycle. To address companies' internal processes and allow developing new processes, buildingSMART International provides documentation about how to develop new IDMs and MVDs [14], [148]. By following these guideline, it was possible to develop a DIM based on the existing BIM concept containing also a damage-specific MVD.

[1] This chapter contains republished work of a retracted article from ASCE [135]. The article has been retracted by the authors because of copy right issues [136]. All content, which was affected by the copy right issues, has been replaced, i.e., Figure 23 from the article has been replaced with Figure 5.14.

BIM is a concept that has been designed to be highly flexible because it shall be used by numerous stakeholders. This led to an extendible approach for modeling building information. An IDM for inspection purposes has already been provided by Sacks, Kedar, Borrmann, *et al.* [119]. Though, this IDM does not include geometric and geo-semantic information data, it has been extended with further information, for example, images. Based on the methodical approach of object-oriented analysis and design [137], a data model for damage information with respect to BIM has been designed. Thus, by providing DIM for bridges, BIM is capable supporting the operation phase.

Structural engineers and inspectors, among others, are involved in the operation phase of bridges. The data model designed is capable to incorporate and deliver necessary information for the operation phase to multiple stakeholders, such as structural engineers, inspectors, or owners. Hence, without manual transfers, inspectors may register damage information digitally and share it with other actors. Flexible approaches, like property sets and objectified relationships, allow to further extend this data model if required.

5.1 Data Analysis

Numerous damage types are known to nationalities [44], [46], [47], [149], [150]. Some are widespread others depend on regional climate, and hence, occur only in particular nations. Several damage types are related to specific bridge types. Germany provides an extensive catalog of defects affecting civil engineering structures [40]. An analysis of this catalog revealed numerous damage types. These types have been designed in a general way to make them applicable to other nations as well. Common damage types are corrosion, spalling, and cracks. Additional types have been defined based on the German damage catalog [40].

A statistical analysis of data retrieved from the Thuringian administration has revealed frequency and severity of defects. Cracks are most frequent in the analyzed data. Under the consideration that the analyzed data is limited to prestressed concrete and reinforced concrete, this result was expected because concrete is a brittle material. If the data set would have focused on steel or wooden bridges other damage types would have been dominant.

Divergences from specification and design occur second most frequently. One reason for this could be the age of numerous bridges - more than 30 years [151]. These 30 or more years old bridges maybe not fulfill the criteria of updated norms and guidelines. Another source of this damage type are errors during the construction process. Independently of reasons, modeling divergences from specification and design may allow a better tracking of violated norms or guidelines.

Joints or expansion joints have gotten less attention until now, hence it was not expected that joint defects play such an important role for assessing bridges' condition. However, the data analysis has shown that joint defects occur frequently with medium impact on bridges' condition.

Waste and other pollution occurs very frequently but with less impact on condition. Only if the state or functionality of bridge elements is affected, these defects become more severe. Moisture penetration may lead to depassivation and corrosion, which leads to high impacts on durability. The following list summarizes eleven damage types that have been selected for modeling after the statistical analysis.

- crack
- spalling
- material changes without material loss
- material changes with material loss
- joint defects
- missing parts
- divergences from specification and design
- foreign elements, e.g., waste or vegetation
- moisture penetration
- insufficient quality of concrete, e.g., low thickness, coarse grain, or voids
- errors in state or functionality, e.g., fixed bearing or clogged drainage

Most studies focus on cracks, spalling and corrosion [12], [39], [60], [78], [80], [89], [91]; however, these damage types do not cover all possible defects and deficiencies. Material changes are partly considered by the sector of HBIM and the given classification of defects is unique in its extent. However, most of the work in the HBIM sector focuses on brick walls and/or wood constructions [96], [98]; concrete is out of scope. The approach provided by this thesis is focused on defects at concrete bridges with respect to more damage types.

Material changes have high impacts on bridge condition states and include multiple damage sub-types from corrosion up to carbonation and alkali-silica reaction. Some of these sub-types would benefit from geometric information, such as the extend of carbonation within a concrete building element. Present studies consider primarily corrosion [39], [73], [89]. Only Kubota and Mikami has mentioned alkali-silica reactions but did not provide a comprehensible data model [128]. Due to the possibility including multiple geometries for a single defect or damaged element, as well as adding material parameters as property sets, the model proposed may also include information about material changes for subsequent structural analysis.

Furthermore, BIM extensions have been published to incorporate climate aspects or natural hazards [116]. However, impacts and defects from natural hazards and climate changes are different from common inspection defects. Natural hazards and climate change often lead to broken elements or a high degree of destruction of buildings. Normal abrasion during the operation phase leads to smaller changes in geometries and semantics of buildings. The present model focuses on defects resulting from abrasion and deterioration. Broken or missing elements were not in the focus of this work.

Joint defects may be represented via semantic information and geo-semantic information. However, joints are usually not part of building models because of their huge quantity. Belsky, Sacks, and Brilakis showed a method to automatically infer joints from given models and represent them by relationships [152]. Unfortunately, a relationship is not a product, which may be affected by a defect. A joint is a (virtual) building element and not only a relationship. If this would be included in the concept of BIM and changed in the definition of the IFC modeling joint defects could be modeled more precisely.

Moisture penetration is not directly a defect, moreover, it is an indicator or reason for other defects, e.g., missing/damaged joints or corrosion respectively. NDT methods, such as ground penetrating radar, are able to identify the extent of moisture penetration [61]. If moisture penetration including their extent could be represented by the building model, induced depassivation and corrosion may be better observable and predictable for engineers. This additional information would enable engineers to estimate the durability of structures more precisely. So far, no study that the author is aware of has considered modeling moisture penetration, although, it has high importance for durability. The DIM presented allows to include information about geometry of moisture penetration. Additionally to that, a quantitative statement about moisture may be included via property sets.

Divergences from specification or design primarily come from outdated construction practices, obsolete guidelines, or construction errors. Especially construction errors, such as missing joints, induce subsequent errors, e.g., moisture penetration. Divergences rely on semantic and geo-semantic information, e.g., related guidelines or photos. Both information elements can be stored in the model proposed in this thesis.

Related to divergences are concrete quality issues, for example, concrete graining or thickness of the concrete cover. Material parameters, for instance granularity of concrete or concrete grade, may lead to subsequent defects, such as spalling, and hence, are important for the durability assessment. This information could be stored semantically as description in the proposed DIM model. Furthermore, if such quality issues are available as 3-dimensional information, this geometric information can be added as well.

Movable parts are affected by impairments of their functionality. This could be unmovable bearings because of corrosion or dirt as well as clogged drainage. Although, these defects may lead to severe subsequent defects and have high impacts on the durability rating, they were not respected in existing data modeling concepts. Using semantic or geo-semantic information from the data model provided here, e.g., property sets or photos, allow to include such information in the bridge model.

The last damage type are missing objects or elements. These occur in case of rivets, screws, screw-nuts etc. Contrary to the intuitive assumption that this defect has no or little effect, the analysis of the German damage catalog shows that strong negative effects are to be expected here with regard to durability. Notwithstanding this fact, this damage type has not been considered for DIM in literature yet. Although it would be possible to include this defect via property sets, it would not be sufficient because such a property would need additional interpretation for visualization. Further research has to address how to visualize missing elements and how to model it properly.

Concluding, based on statistical analyses, eleven damage types have been identified to be covered by a DIM. In a prior study, twelve damage types have been used [12]. Two of those types, "Coarse grain/voids/foreign body encapsulation" and "Thickness and dimensions of concrete coatings" have been merged into concrete quality to reduce the number of types. These results helped to define an adequate DIM. The state of the art shows a rag rug of isolated applications and partly developed data models for damage information. Inspections and assessments during the operation phase of bridges have to consider several different damage types. However, most literature only cares about cracks and maybe spalling.

5.2 Information Modeling

The primary point is to add an entity for defects, which acts as a container for all related damage information, and a relationship to link defects to damaged building elements. The relationship to the damaged building element links the damage and building information. Further damage information may be added to these two elements. These central objects allow to semantically group damage information, which is inspired by the human understanding of a defect. Other literature followed a similar design [39], [60], [90], [93]. A decentralized approach would be better in case of some types of analysis methods, e.g., structural health monitoring, because they are not necessarily connected to a defect. However, traditional inspection focuses on defects. Inspectors combine several information about a structure's state to defects, for instance a lack of material and exposed reinforcement is summarized as spalling.

Borrmann, König, Koch, *et al.* have defined semantic and geometric data for buildings [10]. As far as bridge inspection and assessment highly rely on photos, sketches, or videos, and this information does neither fit into geometric nor semantic information, a new category of geo-semantic information has been added to the BIM concept. The term geo-semantic information has been chosen because images or photos consist of geometric information as the locations of pixels and semantic information as the color of them.

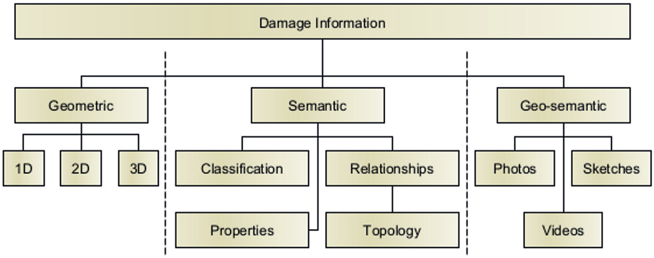


Figure 5.1: Overview of the data categories of defects. Extended categorization of [10]

Figure 5.1 shows an overview of the entirety of the data. Geometric data is used to describe 1D, 2D, or 3D geometries of damage information. Whereby, a 1D geometry information could, for example, be a point or line of a component. 2D geometries are mostly related to plans and 3D geometries contain spatial representations.

Semantic information consists of classifications, relationships and properties. Classifications of defects are, for instance spalling or crack; examples for properties are names, descriptions, and relationships. Several studies have considered that inspection photos have to be included in the data models [39], [70], [89], however, none of them has considered geo-semantic information in their model. With the data model provided, this gap has been bridged by the proposed work.

Damage's topological information are very important for engineers, for example, a crack in the superstructure over the bearing has another meaning and importance as in the middle between two bearings. Topological information contain damage positions in relation to other damages or building elements. This may be reflected via relationships as well.

Figure 5.2 shows an overview of the entire damage model, which highlights if elements have been synthesized from other studies, adjusted, or newly added. The defect annotation as central container similar to other publications [10], [39], [60], [89], [90] and has been adjusted by adding a texture and cause-effect relations. Specific measurements or additional descriptions may be added via a measurement sets that are related to the defect annotation, which is a result of including the work of Hamdan and Scherer into the model [90]. Another result of including the work of Hamdan and Scherer and [93] is to add referred documents. This allows the inclusion of inspection photos or analysis and evaluation reports [90], [93].

Some other studies showed that damage classification is required [39], [128]. The present concepts extend this approach in order to allow classifications in general. Instead of providing a fixed amount of damage types, a classification object that will be related to the defect entity is provided. With this class, the model has a higher flexibility and future damage types may be included without model adjustments. Disadvantageous is, that the semantic information of how to name damage types is missing and has to be defined. Terminology and standardized nomenclature have already been discussed in the concept of BIM in Section 2.1. In short, the bsDD may be used to define common names for required classifications.

The proposed model includes relationships between building elements and defects as well as between defects and defects. The former allows to mark elements as damaged and the latter allows to include causing and resulting defects. Both is required to allow a proper assessment process. Several studies have incorporated damaged elements [39], [93], [120], [128]. Hamdan, Bonduel, and Scherer as well as Tanaka, Nakajima, Egusa, *et al.* included to split defects into spatial part defects [93], [100], but no prior study respected the semantic relationship of a defect causing another defect.

To keep the model flexible and allow adding further damage types, the typification itself has been done via a typification object. This object consists of at least a name and description. So, without changing the entire information model, additional damage types may be included. Respecting that each damage type requires different parameters, all parameters are added as measurement sets. Again, detailed definitions of required measurement sets and nomenclature has to be defined in future.

Besides the semantic information, geometric information has been included. Dependent on the view, different geometries are possible similar to the geometries of buildings. 1D geometries could be, for instance the location of a crack on a building element. Plans or crack maps represent 2D information of buildings or cracks respectively. 3D information of defects represent volumetric geometries and are interesting in case of material changes or spalling. Several studies focused on registering defects in photos and generating the related geometry [58], [60], [153]. 3D Geometries could be used for FEAs [58], [79]. Furthermore,

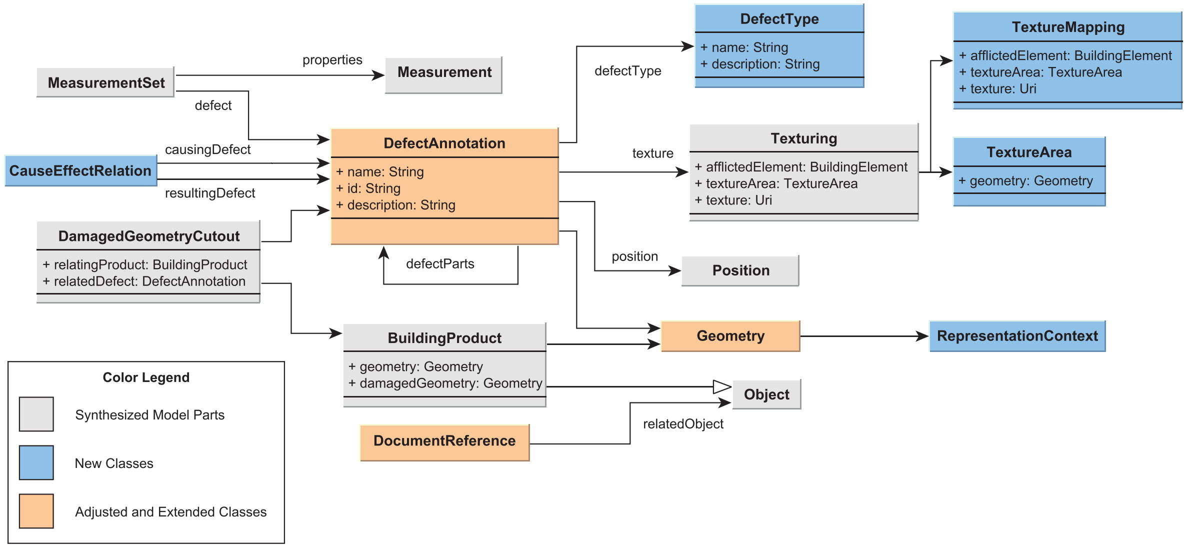


Figure 5.2: UML model of the overall DIM. Grey elements have been included and synthesized from other studies. Blue classes have been added and orange classes have been basically included from existing studies and adjusted.

defect coordinates maybe used for planning aspects or durability analyses. Planning of repair actions benefit from 2D and 3D representations in order to identify locations, plan work orders, and material quantity. All three geometry types are respected by the proposed model, which is novel compared to existing information models.

Depending on the subsequent use case, multiple geometries for the same object may be required. For example, the assessment process needs a plan of the defects to visualize defect distances depending on building element faces. In parallel, a 3D visualization is necessary to improve the spatial perception of the defect. Based on the representation context selected, the geometric representation has to be visualized. The representation context allows to add semantic context information to geometries, which helps the user choosing the appropriate context.

Multiple geometries for a single object allow to include multiple views in the same model. However, the software has to interpret that correctly, which also depends on the use case; for example, if a plan and 3D view exist, all elements that do not have a plan view should be hidden if the plan view is selected. Another usability of this concept is to include different representations depending on inspections, i.e., having one representation context for each inspection. A conceptual sketch of this is shown by Figure 5.3. Defect 1 was detected first during inspection A in 2019 and updated in 2022. Defect 2 was registered first in 2019 and not updated later. If a defect geometry does not vary between two defects, the representation should stay the same. So, the last available representation should be visualized if there is no representation in the context of the current inspection. Regarding the example in Figure 5.3, the geometry of defect 1 from inspection B has to be shown as well as the geometry from 2019 of defect 2 both as the last geometry of the respected defect. Deciding about the correct representation context requires further information or assumptions. The proposed model is capable to store this information; further definitions would be necessary to declare available contexts.

The proposed data model introduces two approaches for modeling the geometry of a damaged component:

1. either the relationship between a defect and the affected element implies geometry effects

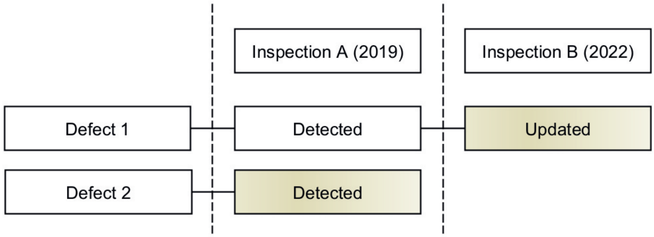


Figure 5.3: Multiple defect states over time. The yellow representation are the last ones and chosen for visualization.

2. or the relationship between a defect and the affected building element do not have consequences for the geometry and geometric aspects of the defect are modeled independently.

Approach (1) is applicable for physical damage types, such as spalling or cracks, and reduces the required information elements. Isailović, Stojanovic, Trapp, *et al.* also used the approach that a relationship implies geometric effects [60]. However, this leads to the circumstance that every defect geometry would be subtracted from a component. As aforementioned, a defect may change over time and can have different geometries. If all of those geometries are included in the same storage and related to the affected building component, several subtractions are performed, which could induce a erroneous geometry of the damaged building element. Approach (2) is beneficial for defects with geometries that should not be handled as subtraction. For this, additional objects and a conscientious modeling and management of the defect, its geometries, and resulting damaged building element geometries are necessary. None of the existing studies considered that both methods are important depending on the damage type, which has been addressed by the DIMproposed here. Although, the second approach requires more entities and management effort, it offers a better flexibility and a more specific defect definition.

Geo-semantic information information as part of DIM has been respected by prior studies already [70], [89], [98], [100]. Hüthwohl, Brilakis, Borrmann, *et al.* showed an example with a photo as texture [89]. Inspections often produce several photos of a single defect and not all of them may be used as texture. In contrast to the data model from Hüthwohl, Brilakis, Borrmann, *et al.*, the data model in this dissertation is able to include textures, and additionally, include photos via external references. Although, Hüthwohl, Brilakis, Borrmann, *et al.* provided some information how to include an image as texture, several important information were missing. To use a photo as texture three requirements have to be fulfilled:

1. the photo should be rectified
2. and as the defect is only at a part of the building element, an extra geometry for the defect texture is required to depict it at the correct position
3. a texturing algorithm is necessary to apply the texture correctly on the chosen geometry,

Rectified photos (1) are addressed in terms of image processing, which is not part of this dissertation. The model proposed in this work allows to add geometries for (2) texture placement. Also, different (3) texture mapping algorithms are supported. Besides this, multiple photos for a defect can be included, as well. Using textures and additional photos, engineers are provided with a 3D textured visualization of the model and further photos may be helpful for in-depth investigation.

5.3 Implementation and Testing

An object-oriented model is only the conceptual description, which has to be implemented for verification. The implementation of this model has to address two problems: (1) the data model needs to be transferred into a proper formal language, either a data modeling language, for instance EXPRESS, or a programming language, such as C#; and (2) software applications that operate on the data model have to be developed.

5.3.1 Implementation using IFC

For the first problem, instead of implementing the data model from scratch in a programming language, the IFC standard, which is based on EXPRESS, has been used. Thus, IFC entities, relationships and properties have been identified to be used for damage information. IFC has been developed to provide entities for the entire AEC sector and possibilities for additional data that may have not been respected yet. A typical example are the proxies within IFC, which allow to integrate objects independently from their semantic context, for instance, bridge models have been created based on the IFC 4 standard even though there were no bearings or other required entities for bridges defined in IFC 4. It is even possible to create models of AEC foreign domains with the help of IFC, like the model of a car; although no one has done it yet. This is possible because IFC includes semantic and geometric information; the parts of the car may be modeled as proxies with additional type objects for classification. Proxies may have a geometry for representation and materials may be assigned as well. This leads to the advantage that existing IFCviewers can be used for visualizing the developed and implemented DIM.

Apart from IFC other possibilities to implement the data model are proprietary data formats, for example, Revit files [69]. Proprietary data formats have the advantage that there is professional software available for editing and visualizing the data. However, existing Bridge Management Systems (BMSs) rely on their own proprietary not BIM conform data formats, and hence, would require to write several data conversion applications to exchange data between stakeholders. Furthermore, also industrial BIM authoring software has not been developed for the operation phase. So, special workarounds are necessary to add damage information.

Different from that are open standards, like IFC, which are supported by current and future applications because the interface is open to everyone. Despite IFC 4 is not designed for damage information, several entities are generic enough to include damage information and several concepts are general enough to use them also for new processes. Existing software in the AEC sector are assumed to be able importing, visualizing, and editing IFC files that are conform to the standard. Disadvantageous is that some semantics or entities are missing, for example a dedicated defect class similar to other objects. Other studies have also utilized the IFC 4 standard for damage information modeling [60], [106], [122]. Because of the missing entities, some researchers decided to extend the IFC standard with additional entities [93]. Additional efforts for developing and implementing a software that supports this extended standard are the primary disadvantage. The present study shows that with existing IFC entities and concepts damage information can be included in building models; allowing to use resulting models in available IFC software, and therefore, lowers the efforts for later software implementation.

Three possible relationships are available within the IFC standard to model the circumstance that a building element is damaged: an assignment, aggregation, and *IfcRelVoidsElement*. An assignment simply provides the information that a defect is related to a building element. A stronger relationship is the aggregation, that implies the defect is part of the building element, which is normally the case. An *IfcRelVoidsElement* is also a decomposition and is used only in combination with openings or voiding features. A voids element relationship includes the subtracting the defect geometry from the building element geometry. Adding a specialized relationship for damaged building elements, would improve the modeling possibilities regarding semantic damage information. Tanaka, Nakajima, Egusa, *et al.* suggested to add two relationships to the IFC standard [93]. After analyzing the requirements, the conclusion may be drawn that one additional relationship with a typification enumeration could be enough. Currently, the IFC standard is already complex; hence, to ease the decision for the correct relationship for a damaged component, as less entities as possible should be added. Existing associations may be used for cause-effect relations.

Hüthwohl, Brilakis, Borrmann, *et al.* have shown how to use textures for visualizing damaged components [89]. The proposed approach added the dedicated texture geometry and texture mapping algorithms to properly depict textures. For this purpose, IFC offers the possibility to apply a texture to a representation item and add mappings via texture coordinates. With these two entities, depicting textures needs fewer assumptions for correct visualization.

5.3.2 Implementation of Software

Using the IFC standard for the implementation of the model, allows utilizing and extending available IFC viewers for damage information visualization. However, two challenges occurred in parallel: (1) a software that provides the best support of the developed concepts had to be identified and (2) the implementation of the data model had to be tested. This combination led to some uncertainties because some of the IFC viewers showed different interpretation of IFC entities, especially in case of voiding features and Constructive Solid Geometry (CSG) geometries. So, if the visualization of the defect was insufficient, it could be either a problem of the software, the IFC file, or both. A viable method to check the correctness of the structure and formal propositions of an IFC file is using an IFC validation tool, like the *IfcCheckingTool* of the Karlsruher Institut für Technologie [154]. If an IFC file passes this checking and the visualization is insufficient, an error in the software was assumed most probable.

Beyond doubt, the IFC standard is an established comprehensive and open interface for sharing building information. This standard has been developed with regards to numerous different stakeholders. It covers manifold geometric design concepts, materials, structural information, and many more; indeed, software vendors are not aware of all possibilities of the IFC standard. There are several potential reasons for that. First, the standard is very complex with its numerous classes, sub-classes, attributes and property sets. To limit the scope and make this complexity manageable, most vendors focus only on their subject. However, for a proper implementation, software vendors need to be aware of other disciplines as well to implement IFC functionalities correctly.

Second, software vendors, such as Autodesk, sell structural analysis or managing software besides their authoring tool, which leads to the interest selling multiple of their products to customers. So, all products can rely on a shared proprietary data format. Primarily relying on a proprietary data format has two advantages for them: (1) developing a single data format for all applications instead of translating data over an open standard saves development time and costs. (2) all applications interact seamlessly with the proprietary data format, which yields to a better user experience.

Third, the documentation of the IFC is primarily limited to the description of instances and some formal propositions. Examples provided by buildingSMART International Ltd. cover only basic problems and concepts [155]. Domain specific example files are rarely provided. Furthermore, a well written guideline for proper implementation of the IFC standard is missing. This could be also an effect of the high complexity of the standard that does not allow one implementation only.

All test files have been edited manually by using a text editor. In order to reduce the time for manual editing of IFC files, all concepts have been individually tested with small files first. These files consist of a single damaged beam with different characteristics of defects, such as textures or geometry subtraction. This prototyping allowed a fast and distinctive testing of the software and concepts. Subsequently, an entire bridge has been set up for a case study, which includes all defects and visualization options.

Although, manual editing of IFC files is a cumbersome and error prone process it was inevitable because none of the existing BIM authoring tools support to add defects and custom building element families do not allow to subtract a geometry from another element in the building project. One possibility would have been to define damaged instances of existing building elements. However, defect geometries may vary, and hence, one family for each damaged component would be required, e.g., a cracked wall with multiple crack widths and paths, which is not representing the designed DIM, and moreover, is an even bigger effort because numerous variants of defects are possible.

In case of changing geometry information in the IFC test files, many parameters and correlations have to be respected; hence, changing the geometry of a defect manually in the IFC file, has taken many iterations of editing and viewing. To simplify this work, as often as possible geometric primitives have been used for conceptual tests. If these tests were successful, geometries have been changed to more complex variants. Furthermore, some text editors allow to define custom syntax highlighting. This eased reading blank IFC files in the text editor. Despite all of these simplifications, methods and tools, a comprehensive damage editing tool is necessary for future research.

Numerous IFC viewers and other BIM software are available. To decide about a software suitable for model verification, multiple software applications have been identified and tested with the prepared IFC files. Table 5.1 shows an overview of all tested software. Only one authoring tool, Revit, has been tested and six IFC viewers. All software application are either extensible via APIs or completely open source except for usBIM. This shows the awareness of software vendors offering options for automatizing workflows or customizing applications in the AEC sector.

None of the tested software was able to cope with all used concepts of the IFC 4 standard; independently if those software is commercial or open-source. In fact, open source programs

Table 5.1: Overview of tested BIM authoring software and IFC viewers

Software type	Name		Extension Options
Authoring software	Autodesk Revit 2019	[17]	C# API, Python API
IFC viewer	apstex IFC viewer	[23]	Open source
IFC viewer	BIM Vision	[22]	C++ and C# API
IFC viewer	Desite BIM	[19]	Javascript API
IFC viewer	Solibri Model Viewer	[24]	Java API
IFC viewer	usBIM	[20]	none
IFC viewer	xBIM Xplorer	[21]	Open Source

tend to support the IFC standard better than commercial software applications. This may also worsen the acceptance of IFC in the industrial environment because commercial authoring tools are primarily used in industry. In case of an erroneous software, the user is not able to separate whether occurring errors result from the standard or from the implementation; therefore, the standard is perceived as wrong or incomplete. Improving and/or ease the implementation of the IFC standard could raise the acceptance in industry.

Less problems occurred visualizing semantic information, which is mainly reflected via tabular or hierarchical views. This requirement also occurs during the design and planning phases, hence, it is a common requirement for BIM software to visualize data like that properly.

Although, the approach of subtracting a voiding feature from another building element is similar to modeling an opening, e.g., the opening for a window in a wall, only a few software applications interpret voiding features correctly. A possible cause of this could be missing knowledge regarding the IFC standard. Voiding features are uncommon in the AEC sector, so the related concept of voiding features is respected less frequently. Relating an opening to an element could also be used modeling a spalling or crack. However, this would be semantically wrong, which led to the decision to use a voiding feature from the IFC standard.

On the one hand, the lack of visualizing images as textures shows that images are of less interest during the design, planning, and construction phase; hence, software vendors were not instructed with implementing such cases or features. On the other hand, as BIM aims to support the entire life-cycle, new contexts and requirements have to be considered also by software vendors, which includes a correct interpretation of textures.

After all tests, the xBIM Xplorer and the Apstex IFC Viewer showed similar and most promising results. Using the xBIM Xplorer for further implementations resulted from the author's experience of several years with C#. Based on the development state of December 2020, custom extensions and adjustments have been added. To keep the code manageable and maintainable, the xBIM team has split the code into three packages for parsing IFC files, generating the geometry, and the final IFC viewer xBIM Xplorer - XbimEssentials, XbimGeometry, and XbimWindowsUI respectively. Numerous interfaces within the parser are necessary to keep the code compatible with IFC 2x3 and 4 as well as implementing concepts like selects from the EXPRESS language. For simplicity, all changes necessary for testing purposes were done in the xBIMWindowsUI package only [156]. This included mainly the possibility to select the visualization context and visualize textures via two texture mapping methods: the spherical texture mapping and explicit texture mapping, both are explained in Chapter 3.

Summarizing, the implementation effort could be limited by analyzing and extending existing software. Several IFC software lacks proper and comprehensive implementation. To support proper implementation, the IFC documentation needs further improvements in the form of examples, precise implementation guidelines, and maybe more strict certification. The open-source applications tended to have a better implementation of the IFC standard. Extending this software is possible due to the code availability. Furthermore, proper and fast support by the software vendors helped implementing missing features. All changes made to the xBIM Xplorer have been fed back to the software vendors and all except the texture visualization have been accepted. This acceptance reflects the importance and value of the developed extensions.

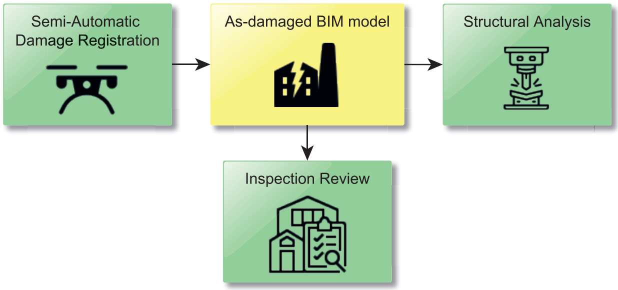


Figure 5.4: Overview of the cases for the proof of concept. Pictograms taken from icons8 [157]–[160].

5.4 Proof of Concept

The resulting as-damaged model may be utilized and processed in the context of different use cases. To provide a verification of the model, three cases have been covered as shown in Figure 5.4: an automatic damage recognition, an inspection discussion, and a structural analysis. Defects found during the automatic damage registration may be used for the inspection review and the structural analysis.

5.4.1 Semi-Automatic Damage Registration

A semi-automatic damage registration may be used to add damage information to the as-built BIM. Generally spoken, several photos of defects at the provided bridge are taken on-site and processed to segment defects and generate the defect geometry that are added to the as-built model [5], [60]. Next, this model is enriched with semantic data, such as further photos, measurements, and documents, which leads to the final as-damaged model. Figure 5.5 shows a schematic overview of this workflow. Such an as-damaged model may be used for subsequent visualization, planning, analysis or assessment.

Figure 5.5 shows an exemplary framework from Artus, Alabassy, and Koch [5]. In this framework, photos are used to generate the point cloud of the bridge, identify defects and generate damage geometries. This data is added to the as-built model. Later, further semantic data is added.

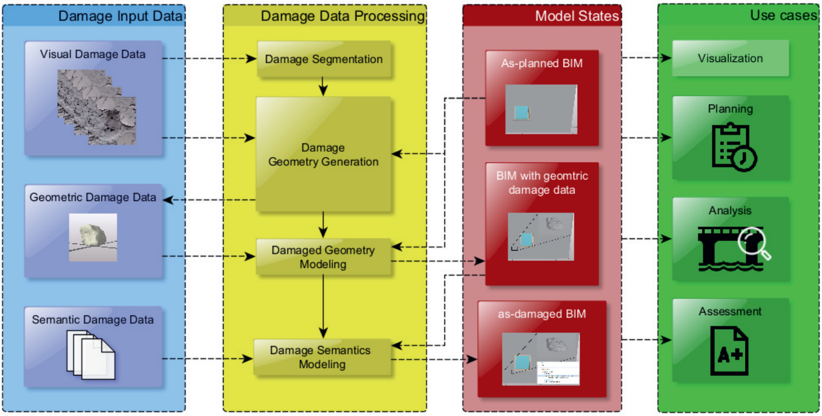


Figure 5.5: Overview of the framework with the damage input data, damage data processing, model states, and possible use cases. Originally published by Artus, Alabassy, and Koch [5]. Pictograms have been taken from icons8 [161]–[164].

Several ML methods are capable to recognize defects [5], [60], [165], [166]. These approaches may be used to identify defects in photos. Furthermore, Structure from Motion (SfM) methods offer possibilities to generate spatial geometries from photos [153]. By orchestrating these methods, a semi-automatic generation of geometric as-damaged models is possible and semantic data is added manually or automatically to the model.

Figure 5.6 shows the bridge that has been used for this case study. The photo has been provided by the "Thüringer Landesamt für Bau und Verkehr". A 3D model has been created via Revit, exported as IFC file and extended with the automatically generated damage geometry and semantics [5]. The bridge is approximately 14 m long. Because this is a short bridge, modeling was done manually. Further tests and evaluation may require long span bridges.

Figure 5.6: Photo of the bridge for case study. The photo has been provided by the "Thüringer Landesamt für Bau und Verkehr".

Figure 5.7 shows the resulting damaged bridge model. a) is a close-up view of the damaged bridge. The exposed reinforcement can be clearly seen in the form of steel stirrups. c) shows the view of section A-A that again is shown in b). A manual generation of such a geometry would be time consuming and error prone because shape consist of numerous vertices and the position may be positioned wrongly. However, a detailed geometry may be not necessary for an assessment, but can be beneficial for later structural analysis, maintenance and test planning.

Depending on the approach for including defect geometries and subsequent utilization, different geometry representations had to be tested because not every software is capable of all geometry types, e.g., xBIM can handle advanced BRep but not faceted BRep.

Elaborating on this process showed that it is possible to use the DIM model for inspections of structures in general. Another result of the process ware highly detailed geometric models of defects; however, the IFC viewers faced problems during the rendering process of geometries with huge amounts of triangles. If not necessary, the complexity of the geometries should be limited to what is effectively necessary. The number of triangles depends on the subsequent utilization of the (geometry) model.

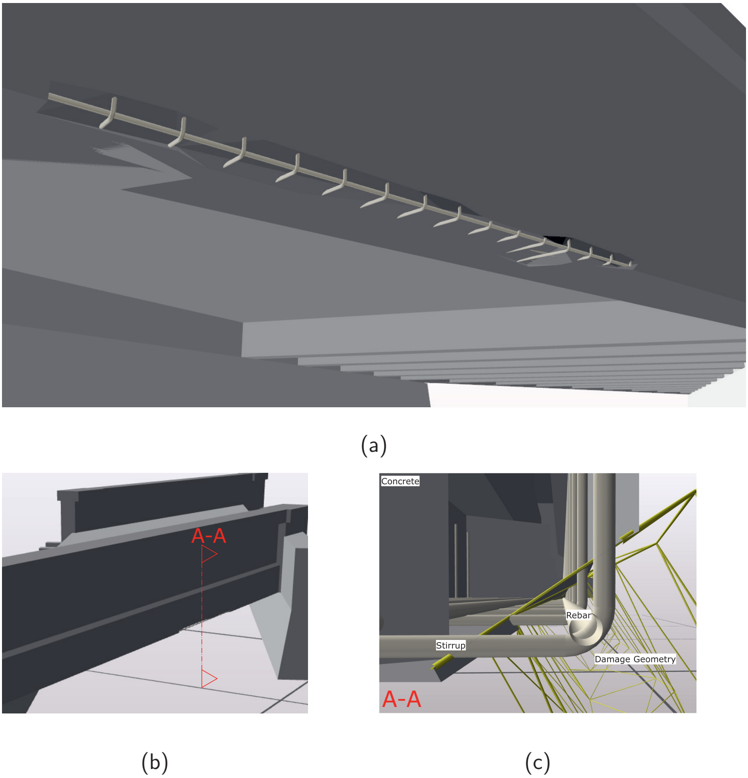


Figure 5.7: Close-up view of the modeled defect. (a) close-up view of the bridge beam with spalling subtraction and exposed reinforcement (b) bridge overview with marked section A-A (c) section A-A with the subtraction as wireframe. [5]

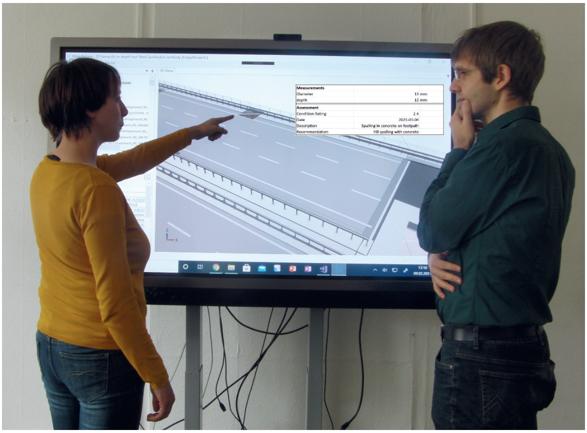


Figure 5.8: Scenario of a Model based inspection review including the discussion of detailed defect geometries.

5.4.2 Inspection Review

On the basis of the damaged bridge model, an inspection review may be performed. Figure 5.8 shows a scenario of discussing defects using the as-damaged bridge model. All defects, their properties, and related documents may be reviewed by a team of engineers. Instead of using drawings and textual descriptions only, a 3D model can be examined by moving around, selecting images, showing related data, discussing defect geometries and their impact on the condition assessment.

For illustration purposes, another bridge model made available by IFC Infra was used [167]. This model has been transformed from IFC 4.2 to IFC 4. Furthermore, damage geometries, measurements, references, and textures have been added. Figure 5.9 depicts the bridge with red markers at the positions of the defects. Starting from the left, there are two test drills at the abutment. The defect at the mid-top represents some cracks in the pavement. Third, the railing in the lower-mid is corroded. Last, there is a spalling on the lower-right at the abutment wall.

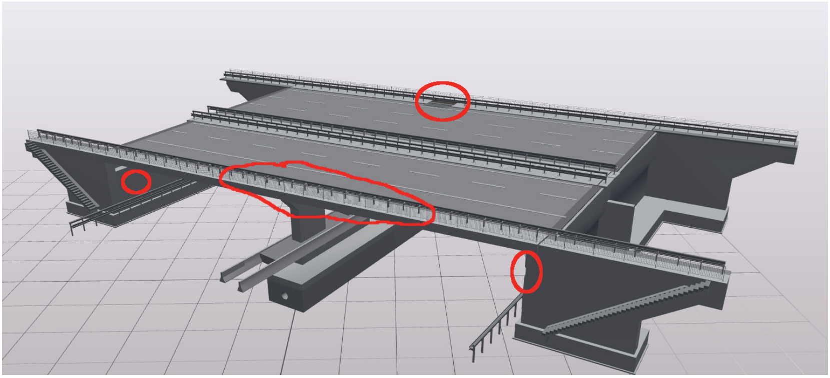


Figure 5.9: Bridge with four defects. The red markers show the places of the defects.

Corrosion

Figure 5.10 shows a photo of a corroded railing at the top-left position. Right next to this photo is the selected railing in the model. The bottom screenshot shows the properties of the railing in the model. Taking adequate photos to represent the corrosion of the entire railing is cumbersome and time consuming. Therefore, a simple property is used to represent this defect. The railing has only a body geometry, and hence, if the damaged component geometric representation context is selected, the railing is not shown anymore in the 3D view. This defect revealed that it is not fully sufficient to use geometric representation contexts for selection. Specialized views are necessary, such as a view with highlighted components or damage textures.

Cracks

Figure 5.11 shows some cracks on the pavement of the bridge. Furthermore, there is a bump in the pavement and the side walk. Only by using this photo as a texture, the inspector or engineer can get a quick impression of the defect. Aligning the texture to the 3D model can help the user gather additional information about related or near elements faster compared to studying 2D plans. However, this defect shows the problem of image

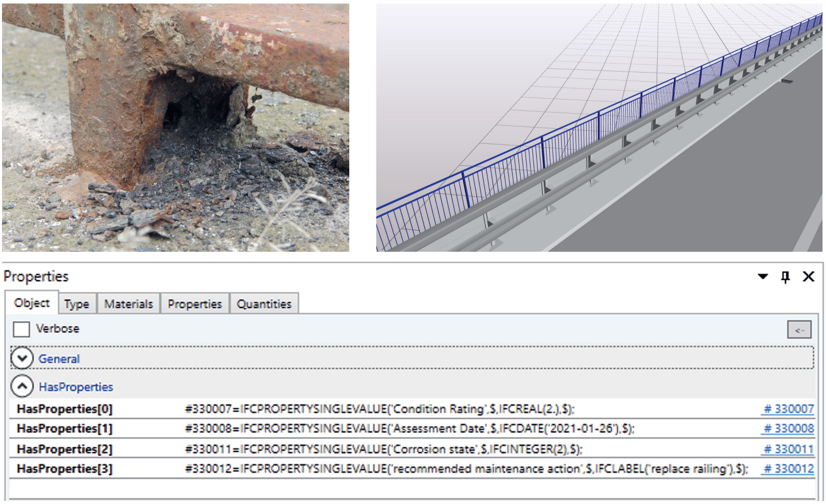


Figure 5.10: On the top left a photo of the corroded railing on-site. The selected railing is right next to the photo in the model visualized by xBIM Xplorer. On the bottom, the figure shows the related property set with a condition rating, an assessment date and further information.

Figure 5.11: Representation of some cracks at the pavement as a texture depicted on a plane.

rectification for textures. The image, which has been used for the texture, has not been rectified. Hence, the texture shows the slope of the bridge at the position of the sidewalk, which could be misunderstood.

Spalling

Figure 5.12 shows an example of a geometric representation of a defect. The left side of Figure 19 shows a photo of the spalling, and the right side shows the defect in the final model. The geometry of the spalling was generated manually within the IFC file. This leads to the visual inaccuracies, such as the different paths of the lower part of the spalling. By using SfM, damage geometries can be modeled with higher accuracy [60]. However, the example shows that the principal concept provides geometry information of a defect.

Holes from Drilling Samples

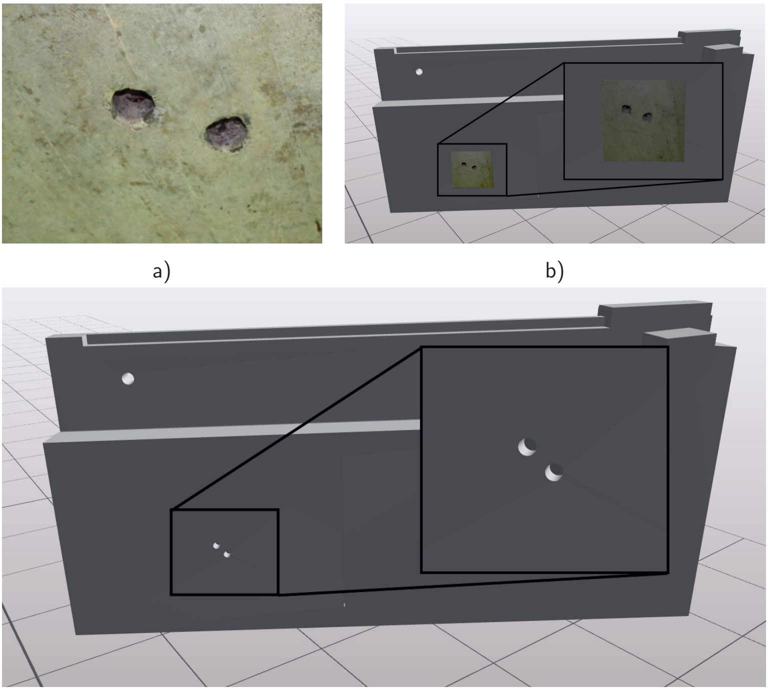
Figure 5.13 shows the representation of two holes from drilling samples in the abutment. Figures 5.13 b) and 5.13 c) show the model after selecting the context of damaged components only. Hence, the abutment with the drill holes is shown without near components. Figure 5.13 b) shows the abutment with a texture at the position of the drill holes. Figure 5.13 c) depicts the visualization of the drill holes by cut-outs. The user can switch between

Figure 5.12: Representation of a spalling at the abutment.

these visualizations by selecting the representation context. Those multiple visualization approaches would provide information about color changes or geometrical information by using the defect photos context respectively the damaged geometry context.

5.4.3 Structural Analysis

After the overall review of the bridge, some components may need further investigation. For this step, a geometry-based structural analysis, e.g., FEA is applicable to determine the impact of the defect on internal forces and stresses. Figure 5.14 illustrates an FEA in ANSYS with an individual beam. As an example, the equivalent von-Mises stresses were calculated. The top of Figure 5.14 presents the 3D model views of the beam and the spalling. The bottom part shows the colored beam in ANSYS and a close look at the beam. The color legend is shown in the lower left screenshot. For the FEA, the IFC file that contains the beam, is converted into a step file by using IfcConvert [25]. The engineer can add load conditions, bearings, and simulation parameters. With this workflow, the geometry of the beam can be imported directly instead of redrawing it. Subsequent to that, the FEA can be performed. This FE model is used as an example, not to perform an in-depth analysis but to show the capability of the information model.



a) b) c)

Figure 5.13: a) photo of some test drills in the abutment. b) photo as texture on the position of the test drills in the model c) geometry of the defect in the building model.

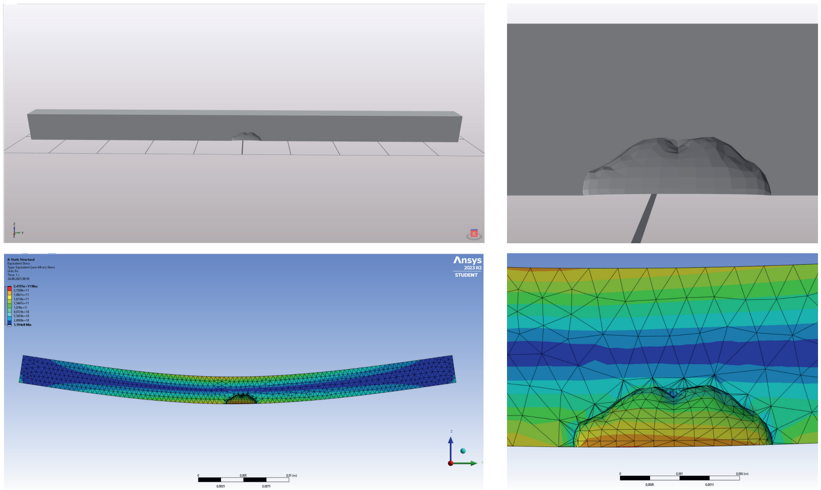


Figure 5.14: Top: BIM model of the damaged beam. Bottom: FEA model in ANSYS Mechanical showing the equivalent von-Mises stress.

To see the potential effect on the principal stress of the defect, Figure 5.15 shows the maximum principal stress of the undamaged (left) and damaged (right) beam. The cut plane for this illustration is approximately at 1m from the left side and illustrated in Figure 5.15 top. The Figure shows that there is a bigger strain in the bottom part especially near the defect. However, this example is to illustrate the use of such geometric models for structural analysis. Problems could be the different meshes of both geometries as evident from the screenshots. Furthermore, the beam has to be modeled with respect to the entire structural system of bridge, which would influence bearings, loads and so on. Advantageous is the usability of existing geometric models to analyze building components in detail. Several of such models for important bridge components may be analyzed in parallel and would provide detailed information to, e.g., predict components vulnerable to defects.

Although, automatic methods for generating FEA models on the basis of BIM models exist [79], there is still research in progress to improve this transformation. Although, the damage geometry may be included automatically by the provided DIM model, experienced structural engineers have to decide about the final assessment. The model designed provides damage information in conjunction with building information which eases the creation of structural models. This helps accelerating the model generation and lowers costs.

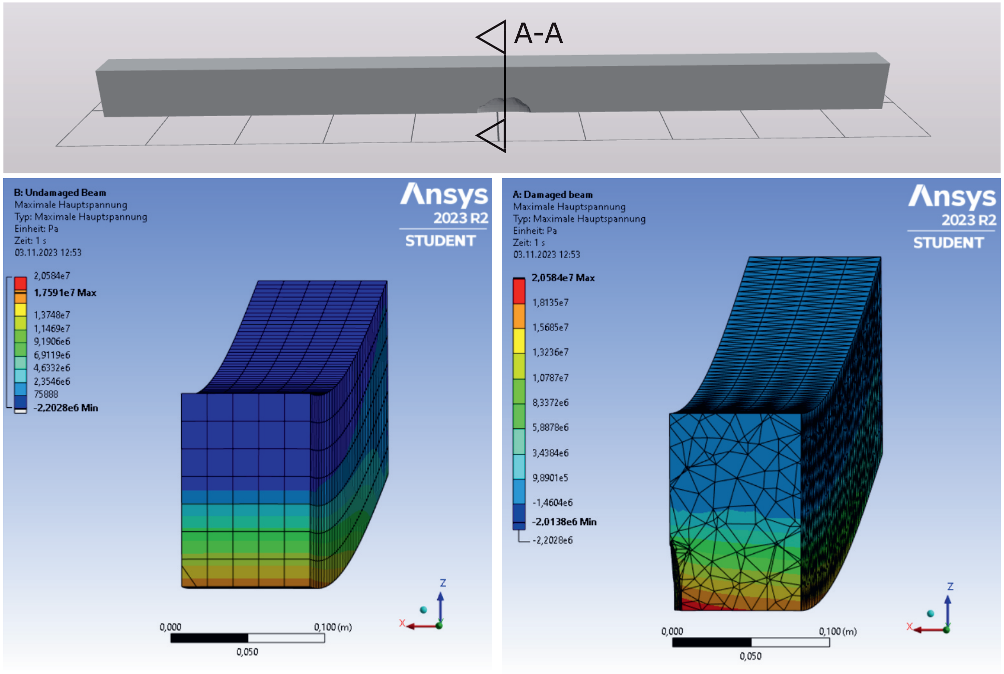


Figure 5.15: Top: cut plane. Bottom: cut view from ANSYS for the undamaged and damaged beam.