Towards Automated Defect Detection: Object-oriented Modeling of Construction Specifications

Frank Boukamp, Carnegie Mellon University, Pittsburgh, PA 15213, USA
(boukamp@cmu.edu)

Burcu Akinci, Carnegie Mellon University, Pittsburgh, PA 15213, USA
(bakinci@cmu.edu)

Summary

This paper describes an ongoing research on the representation and reasoning about construction specifications, which is part of a bigger research project that aims at developing a formalism for automating the identification of deviations and defects on construction sites. We specifically describe the requirements on product and process models and an approach for representing and reasoning about construction specifications to enable automated detection and assessment of construction deviations and defects. This research builds on the previous research on modeling design specifications and extends and elaborates concept of contexts developed in that domain. The paper provides an overview of how the construction specifications are being modeled in this research and points out future steps that need to be accomplished to develop the envisioned automated deviation and defect detection system.

1 Introduction

Several different studies pointed out that errors on construction sites occur frequently (Burati and Farrington 1987; Josephson and Hammerlund 1998; Opfer 1999) and can be costly for the contractors and owners of constructed facilities. In fact, 6-15% of construction cost is found to be wasted due to rework of defective components detected late during construction (Burati and Farrington 1987) and 5% of construction cost is wasted due to rework of defective components detected during maintenance (Josephson and Hammerlund 1998). The nature of these errors is quite diverse. 20-40% of all site defects have their roots in errors arising during the construction phase (Patterson and Ledbetter 1989). According to (Opfer 1999), 54% of the construction defects can be attributed to human factors like unskilled workers or insufficient supervision of construction work. Furthermore, 12% of the construction defects are based on material and system failures (Opfer 1999). These observations suggest that a thorough inspection of construction sites is needed and that current site inspection approaches need to be improved in identifying defects on construction sites effectively. Since the main causes of construction errors, e.g. human involvement in the construction process and changing environmental conditions resulting in discrepancies in material behavior, are uncontrollable, it is critical to improve the inspection and assessment of the quality of construction projects.

Construction industry has started adopting new technologies that enable a more detailed collection of as-built information, i.e. information about the components that are built in place. For example, laser scanners are being used to generate 3D point clouds representing the geometric information about as-built conditions. However, most of the reasoning about this collected as-built information, especially the comparison of the collected information to design information and the construction specifications, is still done manually. This again introduces possible sources for errors since the people doing the comparisons need to know what to look for and what to compare in a given as-built model Thus, they have to have in-depth knowledge about design and construction specifications, which vary from project to project, from the very beginning of a project. Due to the amount of information to be remembered and due to the fact that most of this information changes from project to project, it might not be realistic to expect inspectors to know all the project information and specifications by heart from the beginning. In

addition to this, during manual inspection/analysis of the as-built data, only specific comparisons between the design and the as-built information are made. For example, only the location of corner points in the scanned data of the as-built model are compared to locations of corner points in the design model. With this manual approach, it is impossible to consider the vast amount of data available from reality capture technologies, like laser-scanners, and as a result the users end up not being able to fully leverage the data collected using these technologies.

There is a need for automating the reasoning about the as-built data collected. This reasoning involves automating the identification of deviations between design and as-built models and automating the reasoning about construction specifications to identify construction defects; i.e. deviations that violate specifications. Mechanisms for identifying deviations between two different geometric models, e.g. a design model and a laser-scanned as-built model, are already in place in several laser-scanning software packages. The challenge now is to automatically reason about the identified deviations using the knowledge of construction specifications. One of the key aspects of achieving this automation is to have the design and the deviation information available in a semantically rich, integrated product and process model. Using this model, an automated system can determine the specific contexts in which deviations appear and use the same context representation to determine the specifications applicable to the deviations.

2 Objectives and Vision

The objective of this research is to provide a framework that allows for the representation of and the automated reasoning about construction specifications to support automated defect detection. The envisioned system assumes that as-built and design information is available in an integrated product and process model, which is being updated frequently. Using this information, the system identifies deviations, searches for specifications related to the contexts of the deviations, identifies and applies the specification to the deviations to assess whether the deviations constitute defects. The system then outputs a list of deviations and defects found.
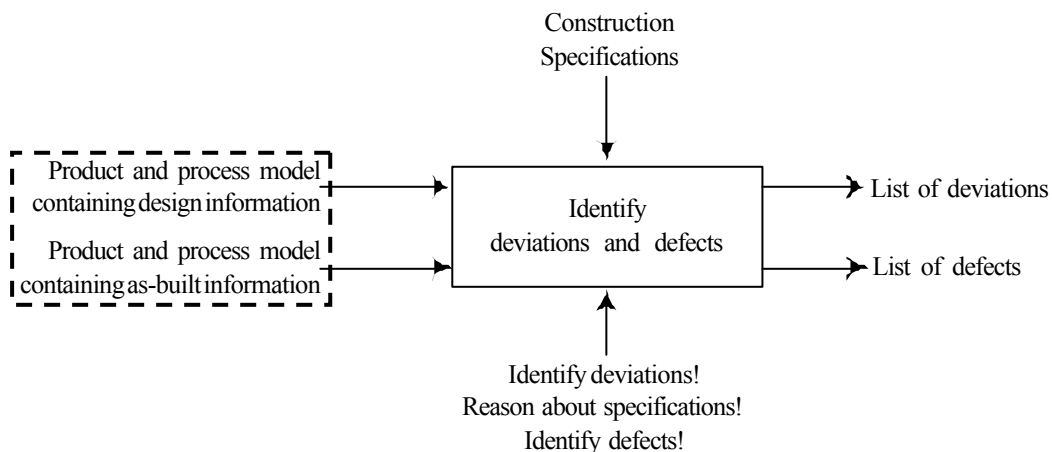

Figure 1: IDEF0 of the envisioned automated defect detection system

The research presented in this paper is part of a bigger research project, entitled **A**dvanced **S**ensor-based **D**efect **M**anagement at **Con**struction Sites (ASDMCon). ASDMCon aims at developing a comprehensive approach to automated defect detection. Different groups within the ASDMCon team work on deploying reality-capture technologies, i.e. laser-scanners and embedded sensors, for capturing as-built data, developing an integrated product and process model for representing design and as-built data in one model, developing methods to improve

the planning of inspections, and finally developing approaches for automated defect detection and management. For more information on ASDMCon, please refer to (ASDMCon 2004).

3 Overview of Construction Specifications and Requirements for Representing and Reasoning about Construction Specifications

It can be observed that construction specifications, which need to be considered for a project, consist of different sets of specifications. First, there are subsets of design specifications that not only relate to the design of a project, but also to specifics about how components should be installed on site. For example, the *Building Code Requirements for Structural Concrete (ACI 318-02)* not only is a design specification but also contains certain construction specifications, since it "…covers the proper design and construction of buildings of structural concrete." (ACI 318-02 2002) Therefore, these subsets of design specifications are an important part of the construction specifications to be considered for automated defect detection during construction. Furthermore, there are specification sets that relate to many different projects and thus are being reused amongst many different construction projects. These specification sets include regional construction specifications, which apply to construction projects in certain geographic regions, and standard specifications for different types of work, like American Concrete Institute's (ACI) *Standard Specifications for Tolerances for Concrete Construction and Materials* (ACI 1990). Finally, there are project specific construction specifications that are written for a specific construction project only. Even though, these project specific construction specifications are sometimes based on existing reference specifications, like the *MASTERSPEC* (AIA 1990), they are often changed and extended by the specification writers to account for project specific requirements.

Another observation is that construction specifications can contain requirements on tasks, components and/or the relationships between tasks, between components and between tasks and components. An example of a construction specification relating to task information and the interrelationship between tasks and construction components is as follows:

> "Do not install precast units until supporting concrete has attained minimum allowable design compressive strength."
> [Source: (AIA 1990), *MASTERSPEC*, Section 03450 - PLANT-PRECAST ARCHITECTURAL CONCRETE, Part 3 (Execution), Paragraph 3.1 B (Installation)]

4 Previous Research

This research builds on, combines and extends prior research in the areas of product and process models and specification representations. A significant amount of research has been done in product and process modeling within the Architecture/Engineering/Construction (A/E/C) industry. A major achievement in this area, which seemed to be best reusable in the research presented here, are the Industry Foundation Classes (IFC) developed by the International Alliance of Interoperability (IAI) (IAI 1996). The goal of the IFC framework is to enable sharing of information between different software systems within A/E/C and Facilities Management (FM) domain by providing a common language to describe construction project information in a semantically rich product and process model. Even though they are not fully comprehensive yet, IFCs nowadays provide the best and most general framework for capturing construction related product and process data. Thus, the IFC framework is considered as being a good starting point for the research presented in this paper. Current IFC specifications, however, do not allow for representing the design and as-built model in an integrated way and hence need to be extended to account for that (Akinci and Boukamp 2002). This research uses the modified IFC-based, integrated product and process model described in (Akinci and Boukamp 2002).

Another area related to our research is specification representation. Similar to product and process modeling, a significant amount of research has been performed in the design specification representation area. Most of the researchers have focused on representing and reasoning about design specifications to analyze a given design. The goal is to have better project designs that comply to design specifications and to help the designers make appropriate decisions if they are unsure about specifics of the design. The previous approaches for design specification modeling range from Basic Models over Condition/Consequence-Models and Object-Oriented Models to Hybrid Models (Fenves et. al. 1987; Garrett and Hakim 1992; Hakim and Garrett 1993; Kiliccote 1994; Kiliccote et. al. 1995; Kiliccote 1996; Yabuki and Law 1993). In our research, we have built on some of the previous formalisms developed for representing and reasoning about design specifications. The specific concepts that we are building on are described in Section 5.2 of this paper.

We found that formalisms for modeling design specifications cannot be directly adopted and reused for our research because of the differences between design and construction specifications. Design specifications only focus on facility components to be designed and their interrelationships. In addition to focusing on facility components, construction specifications also consider task information and interrelationships between tasks and between facility components and tasks. This results in new representation requirements for construction specification models.

In addition, the reasoning needed for design specification evaluation is different from the one for construction specification reasoning. For example, when reasoning about construction specifications, most of the times design information has to be compared to as-built information, whereas the reasoning about design specifications only needs to consider design information alone. Hence, reasoning about design specifications focuses only on one model, i.e., the design model, and reasoning about construction specifications requires focusing on both design and as-built models.

Lastly, even though some of the construction specification sets needed for automated defect detection are – as design specifications – specification sets that are being reused for different projects, there now is a much higher need for representing the specifications in a way that this can be done quickly without much effort or need for actual programming. Construction specifications change and are adjusted or developed completely from scratch for each project. Thus, specification writers need to be able to quickly develop the required specifications for a project to allow automated defect detection within that project.

Another interesting and relevant research directly addressing the structure of construction specifications is the work by (Slava 1985). Even though, this research was focusing on supporting the specification writer to allow her/him to write good construction specifications, it pointed out the general structure of construction specifications from a specification writing perspective. Parts of the identified structure overlaps with what can be seen as a useful structure of construction specification representations for automated reasoning.

A representation for construction specifications, which enables automated defect detection during construction based on product and process information, and which enables fast development of the required specifications for a project is needed. In the following, we will describe an approach that we are currently developing to satisfy these needs.

5 An Approach for Representing and Reasoning about Specifications

In the following, we will discuss our approach on the representation of and the reasoning about construction specifications for automated defect detection. We assume that a detailed integrated product and process model containing both design and as-built information is available. In our research we use the extended IFC version described in (Akinci and Boukamp 2002) as our integrated product and process model. Using this product and process model as an input, the system identifies deviations between the design and the as-built information. Some of these deviations can be expected. For example, as-built information of components that have not been scheduled to be installed yet will not be available in the as-built model but will be available in the design model. Unexpected deviations, however, are then processed through specifications for a given construction project to evaluate whether the deviations violate any specifications and thus constitute defects.

The following sections will describe in more detail the product and process model-based deviation identification and our approach to representing and reasoning about construction specifications.

5.1 Identifying Deviations

As as-built information is captured, it should be stored within an integrated project model to enable others to review the as-built information captured and eventually to compare it to the design information. The comparison of design and as-built information is also necessary to identify deviations and hence, it is an initial step in identifying defects on the construction site.

The identification of construction deviations can be performed either entirely based on the information within the integrated design and as-built project model or it can be performed during the reality capturing process. These two different approaches and their strengths and limitations will be discussed next.

5.1.1 Model-based Deviation Identification

In the first approach, the deviations are solely identified by comparing the as-built information stored in the integrated project model to the respective design information in the same integrated model. There are different approaches to identify deviations based on these two digital information sources. We will discuss a simplistic approach that we initially implemented to identify deviations, and will point out the general requirements that we could derive from our experience.

A simplistic approach to identify deviations would be based on the simple comparison of the values of the design attributes to the values of respective as-built attributes. However, this approach proves to be generally impractical, mainly because this approach assumes that the structure of the design information representation is identical to the structure of the respective as-built information representation. There are several issues about this assumption. First of all, some integrated project model, such as IFCs, usually provides several ways to represent the same information. For example, there are different ways of representing geometric information in IFCs, like using, surface models, solid models, bounding boxes, etc. The algorithm developed should be general enough to map possible different ways of representing the same information in design and as-built models. Moreover, the design and as-built model can have different levels of detail and different decomposition hierarchies. For example, a frame might be represented as being composed of two columns and beams in the design model and in the as-built model it might be represented as one object. In addition to this, the as-built model might contain additional items for temporary structures existing in a scene at a given time while they might not have correspondences in the design model which can result in flaws in detecting

deviations. Finally, this approach requires processing all the existing attributes of all the existing objects within the model since it doesn't use any intelligent rules as to which attributes should be considered and which don't need to be considered. Thus, all objects and their attributes within the integrated project model need to be processed to identify deviations. This makes this approach very slow because of the large number of objects to be expected in the integrated model and the even larger number of attributes to be compared.

A problem of the model-based deviation identification that was identified when deploying this simplistic approach is the challenge of dealing with missing as-built information or missing design information. As-built information of a built component might not have been captured yet, or the information might be missing in the model because the component actually wasn't built yet. A model-based deviation identification approach should be able to use the process information included in the model to intelligently decide whether a component is expected to be built at the time of the search for deviations. If a component is expected to be built, as-built information of the component should be available. If it is only expected to be built in the future, as-built information can be expected to be missing and thus the design information of the component does not need to be processed to identify deviations. Design information maybe missing if the level of detail in the design is not as high as the level of detail of the captured as-built information, leading to as-built information items that are missing a design peer. Also, design information may appear to be missing if for example a construction component was installed that never was intended to be part of the project. For these cases, we currently let the users of the system decide what to do with the deviations identified. An option for the first case is to let the user update the design information so that the level of detail in the design is the same as the level of detail in the as-built information. In the second case, the user will most likely tag the deviation as a defect, since the component was never designed.

In order to address the limitations associated with the simplistic approach's assumption on having identical representations of design and as-built models, we are developing an ontology-based approach which focuses on describing the features of components from a construction specification and defect identification point of view. These ontologies enable the development of model-based deviation identification mechanisms that are not dependent on identical design and as-built information representation structures. The deviation identification mechanisms will only focus on comparing features and attributes of components. At the same time, additional mechanisms need to be developed for extracting the requested features and attributes from the underlying integrated design and as-built model. These additional mechanisms would have to have capabilities to extract features, which result from the grouping of attributes contained in the model, and attributes from the different possible information representations in a given model. For example, the extraction mechanism for a "line"-feature or a "height"-attribute of a column-component needs to be able to deal with the different possible representations of a column, like surface models or solid models.

The development of these ontology-based approaches first seems to be a complex and tedious task due to the expected extensive functional requirements to enable feature and attribute extractions independent of the actual representation of the information in the project model. However, it is found that the extraction itself is not always as complex as expected, due to the highly repetitive usage of information representation structures in the underlying IFC-product and process model structure. For example, construction products in IFC all build on the same possibilities of geometric representations. Thus, a geometric-feature extraction mechanism can be reused among all the different construction products in the integrated product and process model. Also, the extraction of attributes sometimes does not require complex processing since some of the requested attributes maybe represented in the product and process model explicitly and do not have to be derived from other attributes.

With feature and attribute ontologies in place, automated model-based construction deviation identification mechanisms can be developed. However, the deviation identification mechanism itself now becomes more complex, because of the knowledge-intensive nature of the approach. Due to this expected complexity, the research team simultaneously works on ideas for outsourcing the deviation identification mechanisms to the data processing agents of the reality capturing technologies.

5.1.2 Outsourcing the Deviation Identification

Instead of implementing internal deviation detection algorithms, the deviation identification can be outsourced to the data processing agents of the reality capturing technologies. For example, currently there are systems in place that process the data captured through laser scanners and identify deviations between a design model and the scanned as-built model. Thus, it makes sense to outsource the deviation identification to the reality capturing agents and their software tools. However, this imposes the need for a deviation representation within the integrated product and process model so that the deviations can be made available to the automated defect identification and assessment system that will process these deviations and compare them to construction specifications. Such a deviation representation does not exist in product and process models, such as IFCs. The research team is currently investigating, how such deviations can be represented in an integrated project model. This includes identifying what information about a deviation needs to be represented and how the representation can be attached to the entities in the model.

The possibility for visualizing deviations and defects was found to be critical to help users of the automated defect detection system to better grasp the nature and the context of a deviation or a defect. Thus, inserting only the deviation information into the integrated product and process model is not sufficient since the deviation then cannot be visualized effectively. Therefore, the actual captured as-built information still also needs to be inserted into the model, to enable later review of the captured as-built information.

5.2   The Pluggable Specifications

Deviations between design and as-built conditions, once identified, need to be compared to requirements imposed by the construction specifications to identify any specification violations, i.e. defects. To enable automated defect detection, construction specifications need to be represented in a way that automated reasoning about them is possible. This includes automated identification of specifications applicable to identified deviations and automated assessment of the deviation.

Our approach on representing construction specifications for automated defect detection is based on some of the concepts developed in previous research on design specification modeling. A major idea reused and altered in our approach is the idea of using the notion of contexts in an object-oriented specification representation approach, introduced by Kiliccote in his *Context-Oriented Design Representation Approach* (Kiliccote 1994).

A *context* in Kiliccote's approach is an object to which the design specifications applying to that context are bound. The context object thereby is capable of identifying whether a given designed object is within a specific context and thus can identify whether the specifications bound to that context are applicable to the designed object. This usage of contexts eliminates redundancies in defining conditional parts in specifications and avoids complex and highly specialized class hierarchies of specification classes.

Kiliccote's approach structures the specifications around contexts, which in return are classifiers of genera, i.e. physical or conceptual objects in the design. For example, a context for "column" existed with subcontexts for specifying the types of columns further, e.g. "concrete column" or

"steel column", to which design specifications bound to these more specified context-objects were applicable (see Figure 2).
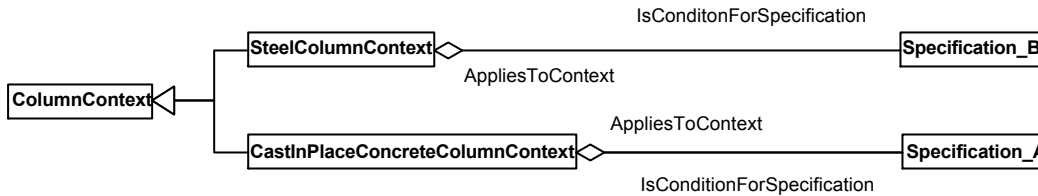

Figure 2: Usage of Specification Contexts in Kiliccote's Approach

Having the contexts only describing specific genera and thus having a detailed classification of physical and conceptual objects, however, was found to be insufficient for construction specifications because of two reasons. First, construction specifications often do not apply to a specific genus. An example for that is the following specification:

> "ACI 117-90: Standard Specification for Tolerances for Concrete Construction and Materials
> Section 4 – Cast-in-place concrete for buildings
> Section 4.1 – Vertical Alignment
> 4.1.1 For heights 100ft or less
> *Lines, surfaces and arrises......1in.*"
> [Source: (ACI 1990) *Standard Specifications for Tolerances for Concrete Construction and Materials (ACI 117-90)*]

As can be seen, no specific construction object is mentioned within this specification. It applies to any construction product as long as it is built out of cast-in-place concrete and situated in buildings. Thus, representing this specification using Kiliccote's approach does not work completely since the context object now would not be specifying a genus, like a concrete column, but specify the characteristics of any genus, like the above specification specifies that the genus has to consist of cast-in-place concrete without specifying the type of the genus. Another reason why Kiliccote's concept of context needs to be modified is that many context descriptions of specifications are sharing the same conditional parts. For example, the above specification uses the condition "cast-in-place concrete," so do the specifications in Section 7, entitled "Cast-in-place, vertically slipformed building elements," of the same ACI specification. Thus, it is necessary to allow specifications to combine different contexts to define more specific contexts rather than to use very specific contexts from a deep context-hierarchy. For example, the context "cast-in-place concrete column" would be represented by combining a "cast-in-place concrete" context with a "column" context (see Figure 3), instead of having one "cast-in-place concrete column" context as a context subclass of a "column" context, as it would be the case in Kiliccote's approach (see Figure 2).
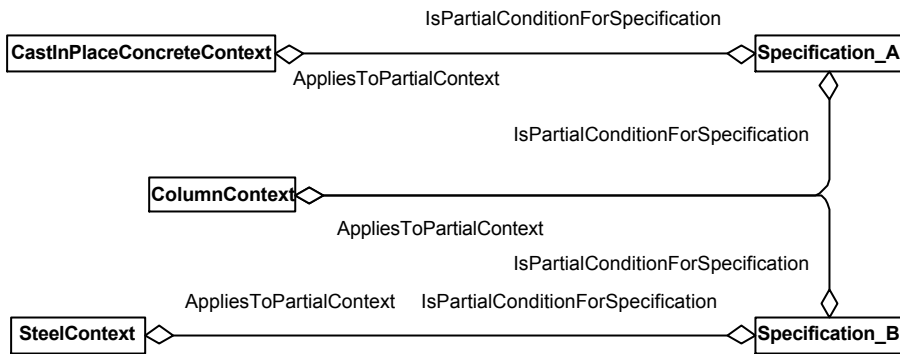
Figure 3: Proposed Usage of Specification Contexts

Besides having context objects assigned to the specification object, we found that a specification object needs to have three additional parts:

1. Requirements: In the example specification above, the requirement to be represented would be "For heights 100ft or less".

2. Targeted features: In the example specification above, the features targeted by the specification are "line", "surface" and "arris" (see 1).

3. Targeted behaviors: In the example specification above, the feature-behavior targeted by the specification is "Vertical Alignment".

The requirements will consist of conditions on certain attributes, like the "height" in the above example. To define requirements and targeted features, we will draw upon the proposed attribute ontologies introduced in section 5.2.1. Finally, the targeted behavior of the features or attributes need to be expressed for the specifications. In reasoning about geometry, these behaviors would be for example "Vertical Alignment", "Placement". For material information these behaviors would include for example "meet or exceed" behavior for the "compressive strength"-attribute of concrete components. Different behaviors addressed in construction specifications also need to be identified in a behavior ontology.

The different libraries of *contexts*, *attributes*, *features* and *behaviors* allow us to simplify the process of creating construction specification representations for automated defect detection for construction specification writers. Construction specification writers will be able to simply "plug together" specifications by choosing *contexts* from a library of contexts, defining requirements by choosing from a library of possible *attributes*, choosing targeted *features* from a library of features and finally choosing the targeted behavior of those features from a library of behaviors. This approach for developing pluggable specifications will potentially be very valuable since for every construction project new specifications have to be developed and/or existing specifications have to be adjusted based on the specifics of the new project.

Once the specifications are modeled, the processing of them will start when a deviation is found. For each deviation, all specification instances will be searched, the relevant specifications will be identified using context, and those specifications identified will be processed to assess whether they are being violated.

A different approach would be to relate specifications directly to each of the building components in the project model. However, this linking process is a tedious task and can be quite error prone since many specifications can be applicable to many different types of objects. Approaches have been developed that allow to automatically link text-based specifications and other text-based documents to objects within IFC-based product and process model (Caldas and

Soibelman 2003). However, this linking process would have to be performed for every new project and would have to be repeated every time the design is updated, which happens frequently during the construction process. The latter imposes new potential sources for errors, since one might forget to update the linking of specifications to objects in the product and process model after changing the design. This error source is eliminated in the proposed approach since the linking is performed when deviations are being identified and evaluated, which again is assumed to be done based on the updated design information.

The first step in determining the applicability of specifications to a deviation is evaluating the context objects that are assigned to the specification objects. Thereby, the context objects will be processed in sequence. If a context object finds a deviation to satisfy the context conditions, the context object stores this information and the next context object of the specification is being processed. If a context is found to be not applicable to a deviation, the processing of the specification is stopped immediately and the information "not applicable" is stored in the context object. Storing the information, whether a context is applicable or not speeds up the searching and processing of the specifications, since the context-objects are shared among the different specification objects. This means, that a context only needs to be processed once for a given deviation. Other specifications that use the same context can be processed faster, since the context-object will immediately return "applicable", without the need for repeating the actual context evaluation process. All the context objects will be reset to "not processed" when a new deviation out of the list of identified deviations is being evaluated.

If all the context objects assigned to a specification are found applicable to the deviation in question, processing continues by evaluating the requirement part of the specification. If the requirement is satisfied, the features, which relate to the deviating object in the product and process model and which the specification targets, will be extracted and the deviation will be evaluated. If it is found that the deviation violates the specification, the deviation will be categorized as a defect.

6   Future Steps

Additional work will be required in different areas to enable automated defect detection. Ontologies for features and attributes of products and processes required from a construction specification viewpoint have to be developed. Also, ontologies for describing deviations and defects will be needed to allow for the development of meaningful descriptions of deviations and defects. Product and process models need to be extended to allow for including the representation of temporary work, like formwork. Laser scanners' data processing systems currently identify temporary work only as obstacles, or in the case of formwork, as deviations in the shored component, since no knowledge about the temporary work can be gathered from the product and process model. Also, it will allow for including specifications that target temporary work. Especially for formwork, this is important to allow for the automatic identification of problems, like displacement, before the concrete is actually poured into the formwork, making it more difficult and expensive to correct the problem. Also, an automated defect detection system that only focuses on product and process information will not be able to highlight issues early on for which the source lies in the construction procedure itself. Thus, novel approaches need to be developed to describe procedural information digitally and to monitor the procedures on site.

Finally, the A/E/C industry has to progress in the usage of product and process models for construction projects during the design and construction phase. All information about a construction project needs to be stored in the project's model. Issues with different levels of detail required by different parties or by the automated defect detection system need to be resolved. Only with the information in place, a comprehensive automated defect detection system can be achieved.

7   Conclusion

The objective of the system described in this paper is to identify defects early on and to improve the inspection process by a more detailed and more thorough inspection. Also, the system could educate the users about deviations that seem to be critical because they are close to violating specifications. This would allow the construction personnel to react to these deviations accordingly and thus possibly prevent future defects or deviation propagation. It is expected that such an automated defect detection system will help educating the construction industry, enable the construction industry to reduce costs and risks and to improve overall construction.

8   Acknowledgements

9   Endnotes

1) An "arris" is defined in the specification *ACI117-90 Standard Tolerances for Concrete Construction and Materials* as "…the line, edge, or hip in which two straight or curved surfaces of a body, forming an exterior angle, meet; a sharp ridge, as between adjoining channels of a Doric column…".

10   References

AIA (1990), "MASTERSPEC", American Institute of Architects, 1990

ACI (1990), "Standard Specifications for Tolerances for Concrete Construction and Materials (ACI 117-90)", American Concrete Institute, Reported By ACI Committee 117, 1990

ACI (2002), "Building Code Requirements for Structural Concrete (ACI 318-02) and Commentary (ACI 318R-02)", American Concrete Institute, Reported By ACI Committee 318, 2002

B. Akinci and F. Boukamp (2002), "Representation and Integration of As-Built Information to IFC Based Product and Process Models for Automated Assessment of As-Built Conditions", Nineteenth International Symposium on Automation and Robotics in Construction (ISARC 2002), September 23-25, 2002, Washington DC, USA.

ASDMCon (2004), "Advanced Sensor-Based Defect Management at Construction Sites – Early Defect Detection and Management at Construction Sites", <http://www.ce.cmu.edu/~itr/>, April 2004

J. L. Burati, Jr. and J. J. Farrington (1987), "Construction Industry Institute – Costs of Quality Deviations in Design and Construction", Clemson University, Source Document 29, August 1987

C. Caldas and L. Soibelman (2003), "Integration of Construction Documents in IFC Project Models", Proceedings of the 2003 ASCE Construction Research Congress, Honolulu, HI, March 2003

S. J. Fenves, R. N. Wright, F. I. Stahl, K. A. Reed (1987), "Introduction to SASE: Standard Analysis, Synthesis and Expression", Technical Report NBSIR 872513, National Bureau of Standards, Washington, D.C., 1987

J. H. Garrett, Jr., M. M. Hakim (1992), "Object-oriented Model of Engineering Design Standards", Journal of Computing in Civil Engineering 6(3), pp. 323-347, 1992

M. M. Hakim, J. H. Garrett, Jr. (1993), "Using Description Logic for Representing Engineering Design Standards," Journal of Engineering with Computers, Vol. 9, pp. 108-124, 1993

IAI (1996), International Alliance for Interoperability, <http://www.iai-international.org/iai_international/>, April 2004

P. E. Josephson and Y. Hammerlund (1998), "Causes and Costs of Defects in Construction: A Study of Seven Building Projects", Automation in Construction, 8(9), pp.681-687, 1998

H. Kiliccote (1994), "The Context-Oriented Model: A Hybrid Approach to Modeling and Processing Design Standards", M.S. Thesis, Department of Civil and Environmental Engineering, Carnegie Mellon University, 1994

H. Kiliccote, J. H. Garrett, Jr., B. Choi and K. Reed (1995), "A Distributed Approach for Representing and Processing Design Standards," Proceedings of the 6th International Conference on Computing in Civil and Building Engineering, pp. 343-350, Berlin, Germany, 1995

H. Kiliccote (1996), "A Standards Processing Framework", Ph.D. Thesis, Department of Civil and Environmental Engineering, Carnegie Mellon University, 1996

N. D. Opfer (1999), "Construction Defect Education in Construction Management", ASC Proceedings of the 35th Annual Conference, California Polytechnic State University – San Luis Obispo, California, April 7-10, 1999

L. Patterson and W. Ledbetter (1998), "The Cost of Quality: A Management Tool", Excellence in the Constructed Project, Proceedings of Construction Congress I, San Francisco, California, pp.100-105, 1998

M. T. Slava (1985), "Structure and Organization of Project Specifications", M.S. Thesis, Department of Civil and Environmental Engineering, Carnegie Mellon University, 1985

N. Yabuki and K. H. Law (1993), "An object-logic model for the representation and processing of design standards", Engineering with Computers, 9, pp.133-159, 1993