A CONCEPTUAL APPROACH TO TRACK DESIGN CHANGES WITHIN A MULTI-DISCIPLINARY BIM ENVIRONMENT

Behzad Pilehchian [a 1], Sheryl Staub-French [b *], and Madhav Prasad Nepal [c]

[a] *Graduate Student and Research Assistant, Department of Civil Engineering, The University of British Columbia, Vancouver, BC, Canada; Tel: +1 604 833 2797; Email: bpl@alumni.ubc.ca*

[b] *Department of Civil Engineering, The University of British Columbia, Vancouver, BC, V6T1Z4, Canada; Tel: +1 604 822 2637; E-mail: ssf@civil.ubc.ca*

[c] *School of Civil Engineering and Built Environment, Queensland University of Technology, 2 George Street, Brisbane QLD 4001, Australia; Tel: +61 7 3138 9257; Email: madhav.nepal@qut.edu.au*

[*] Corresponding Author

[1] Currently, *Senior Civil/Structural Engineer, SNC-Lavalin, Vancouver, BC, Canada*

AN APPROACH TO TRACK DESIGN CHANGES WITHIN A MULTI-DISCIPLINARY BIM ENVIRONMENT

Abstract: Successful management of design changes is critical for the efficient delivery of construction projects. Building Information Modeling (BIM) is envisioned to play an important role in integrating design, construction and facility management processes through coordinated changes throughout the project life-cycle. BIM currently provides significant benefits in coordinating changes across different views in a single model, and identifying conflicts between different discipline-specific models. However, current BIM tools provide limited support in managing changes across several discipline-specific models. This paper describes an approach to represent, coordinate, and track changes within a collaborative multi-disciplinary BIM environment. This approach was informed by a detailed case study of a large, complex, fast-tracked BIM project where we investigated numerous design changes, analyzed change management processes, and evaluated existing BIM tools. Our approach characterises design changes in an ontology to represent changed component attributes, dependencies between components, and change impacts. It explores different types of dependencies amongst different design changes and describes how a graph based approach and dependency matrix could assist with automating the propagation and impact of changes in a BIM-based project delivery process.

Keywords: building; design management; ontology; dependency matrix; change management; level of development; model evolution, traceability

Introduction

Changes in design are inevitable due to the iterative and exploratory nature of design. The content and structure of design is not static but subject to continual changes even after construction has started, particularly on fast-track projects. Design changes often generate unanticipated side effects on project cost, schedule and quality (Lee and Pena-Mora 2007), which can become a major cause of project delay, cost overruns, defects, and even project failure (Hao et al. 2008). The real impacts of changes are often overlooked or revealed later in the process when making adjustments or identifying alternative solutions (Isaac and Navon 2009). Current tools and methods used for project planning and design provide limited support for evaluating the consequences of a specific change before the project plan and design are fully updated (Hegazy et al. 2001; Motawa et al. 2007). The timely identification and analysis of the consequences of design changes is therefore critical for the successful delivery of construction projects, particularly as more projects are being executed using fast-track approaches.

Building Information Modeling (BIM) is envisioned to play an important role in identifying the impacts of project changes. BIM is becoming more prevalent in the construction industry (Arayici et al. 2011) and change management has been identified as an important application area (Aslani et al. 2009; Hajian and Becerik-Gerber 2009). Moreover, the capability of integrating time and cost within BIM (4D or 5D integration) along with the prospects for the integration of data from multiple disciplines and sources via BIM-servers, demonstrates the significant potential of BIM to facilitate the management of changes throughout the project life cycle. However, the current reality of BIM use on construction projects demonstrates that significant challenges remain in realising this vision. Project teams still rely largely on subjective expert opinion and manual analysis of 2D drawings to identify the impact of changes due to the lack of computer-based support required by practitioners (Navon and Isaac 2009).

This paper describes a conceptual approach to represent, coordinate, and track design changes in the context of a collaborative multi-disciplinary BIM environment. We developed this approach by analyzing design changes and change management practices on a large, complex fast-track BIM project, and by evaluating the capabilities of state-of-the-art BIM tools to assist with the change management process. Our approach characterises design changes in an ontology to represent changed component attributes, dependencies between components, and change impacts. We propose a graph based approach and dependency matrix as a mechanism to assist with the automatic propagation and impact of changes in a BIM-based multi-disciplinary project environment.

Related Research

Research on change management systems has tended to focus on documenting best practice for managing changes (e.g., CII 1994; CIRIA 2001), developing change management systems (e.g., Ibbs et al. 2001), evaluating the effects of project changes on certain project elements (e.g., Lee et al. 2004), and investigating the IT-based change management approaches in construction (e.g., Soh and Wang 2000). A number of research efforts (e.g., Mokhtar et al. 1998; Hegazy et al. 2001) have focused on developing information models intended to improve the coordination of design information through the management of design changes. Mokhtar et al. (1998) presented a central database of building components data to track past changes and assist in the planning and scheduling of the future ones. Isaac and Navon (2009) developed a graph-based model for automatic identification of the possible consequences of changes prior to their implementation in the design and planning of building projects. The model utilizes available sources of project information in order to identify the impact of changes on cost, schedule and performance of the project. More recent publication of these authors focuses on extracting information about existing project elements and their relationships from BIM using the IFC platform. Their approach however considers only the spatial relationships between different systems or components of a building and does not consider the functional or operational dependencies of different parts or components within a system (this issue will be discussed further in subsequent sections).

Other research efforts have investigated the utility of BIM tools and AEC-related software for automating the tracking and management of project changes. Wang et al. (2007) presented a semi-automated approach for detecting the differences between versions of a data model. Their approach incorporated a taxonomy for classifying version differences. Akcamete et al. (2009) performed two construction case studies in order to understand the types of changes that occur during the life-cycle of a project, with a particular focus on facilities management and maintenance activities. They discuss some challenges associated with managing such changes to the successive updates of building information models. They also investigated the capability of commercially available BIM tools to address these challenges. Nour and Beucke (2010) introduced an approach to integrate object versioning (as a change management concept) and the IFC model (as a neutral building information model) to address requirements of design change management in a multidisciplinary collaborative environment. Koch and Firmenich (2011) goes further in providing an approach for integrating existing version-oriented information (i.e. state-oriented descriptions of virtual buildings) with change-oriented information using processing-oriented modelling.

Researchers have emphasized a need for adding change-based information or semantics to existing building models (Koch and Firmenich 2011), providing computational support for identifying the impact of changes (Akcamete et al. 2009), and developing computer-based tools in order to automate the change management process (Isaac and Navon, 2008). Navon and Isaac (2009) identified three general problems or shortcomings of the existing computer models of building projects:

(1) Project data are currently stored in different sub-models, e.g. requirement, design, planning, risk, etc, which are not fully integrated (the need for an integrative model).
(2) It is very difficult to update the building model following changes in the project. The model should allow continuous modification of data and the automatic propagation of changes (the need for an adaptive model).
(3) Building models should incorporate uncertainties and link the project components explicitly both to the identified sources of uncertainty, and to the assessed probability distributions of the cost and duration of the planned activities (the need for a stochastic model).

Our research mainly focuses on the first two requirements identified by Navon and Isaac (2009) which are also exemplified by (Koch and Firmenich 2011), i.e. the need for an integrative and adaptive model, and attempts to lay down a foundation to address these requirements in the context of BIM-based change management. It is argued that an adaptive and integrative BIM would be able to maintain the consistency of data throughout the building model when a specific component is modified and would support the automatic propagation of changes, thereby relieving users from having to perform the necessary adjustments of the model manually. We investigate the conceptual characteristics of design changes that have an essential role in such adaptation and integration processes and provide some insights to assist with automating the propagation of changes in a BIM environment.

This research builds on the ontology developed by Akcamete et al. (2009) to represent design change characteristics and extend it to provide richer change attributes and to represent change patterns and dependencies. We categorize the design changes encountered throughout our case study in a taxonomy of changes similar to those developed by Akcamete et al. (2009). Although their focus was more on operation and maintenance activities, the change characteristics they identified provide conceptual facets that are also applicable to building design and construction. We extended these characteristics to build a richer set of change attributes and to represent change patterns and dependencies. The graph-based model developed by Issac and Navon (2013) gets input from BIM and other information sources in order to track the consequences of changes. In contrast, our approach identifies different types of dependencies and provides specifications on what additional information is needed in a BIM to be able to comprehensively track the consequences of design changes. Our approach also builds on the classification of version differences developed by Wang et al. (2007) to support change tracking and complements the processing-oriented modeling suggested by Koch and Firmenich (2011).

Research Methodology and Process

We studied the design and construction of the Pharmaceutical Sciences Building project, a $150 million project that was constructed at the University of British Columbia (UBC), Vancouver, Canada (Figure 1 a). We conducted a long-term ethnographic field study of this large, complex, fast-track building project, with complex Mechanical, Electrical, and Plumbing (MEP) systems, to examine change management in the context of a collaborative, multi-disciplinary BIM environment. BIM was used during the design and construction of this project as a contractual requirement and because of its potential to improve the design and construction coordination and constructability.

We studied the project extensively to collect data on design changes, to document the requirements of practitioners in managing changes in design, and to evaluate the functionality of commercially available state-of-the-art BIM tools, including Autodesk® Revit®, Navisworks®, Solibri Model Checker™ (SMC), and Vico Doc Set Manager™ against these requirements. We collected data through observational studies of over 40 BIM coordination meetings, extensive field studies, formal and informal communication with design and construction professionals, and analysis of design changes and related project artifacts (e.g., BIM's, 2D drawings, RFI's, change logs, site instructions, etc.). Figure 1 (b) and 1 (c), respectively, show BIM coordination meetings in the architect's office during the early stages of the project, and at the construction site afterwards. More detailed description about this BIM project is provided in Pilehchian and Staub-French (2012).

From this observation-based extensive data collection process, we identified the common characteristics and patterns of the design changes and developed an ontology of design changes. We developed a graph-based approach to trace and facilitate the tracking of design changes across different models and levels of detail. The next sections describe the design changes in detail and the ontology of design changes we developed to represent design change characteristics and the nature and dependency between changes.

Ontology of Design Changes

We analysed the numerous examples of design changes that we documented in our study to identify and characterize the common characteristics of design changes in a taxonomy of changes. Table 1 shows a portion of this taxonomy that includes the first twenty recorded changes. The common facets identified during this classification provide the basis for developing the ontology of design changes.

The developed ontology explicitly defines a BIM-based conceptual structure to organize project changes and highlights the characteristics that are essential for tracking the impact of design changes. In developing the ontology, the most generic characteristics of changes are defined first as object-oriented, adaptation oriented and integration oriented change classes. Object-oriented classes represent change characteristics that are related to geometry, position or specification of components. Adaptation-oriented classes constitute important component characteristics that are important for updating the building model following the changes in the project by continuous modification of data and the automatic propagation of changes. Integration oriented change classes, on the other hand, represent those component characteristics that are important for integrating project data that are currently stored in different sub-models (e.g., requirement, design, planning, scheduling, risk, etc.). The development of these upper level classes is informed by previous research, as noted earlier. A bottom-up approach was then used to define the most specific classes, essentially the leaves of the hierarchy. These classes are based on the common facets we identified throughout the classification of changes encountered during the case study. We subsequently grouped these lower-level facets into mid-level concepts (subclasses) and related them to the top-level characteristics in a superclass–subclass hierarchy to develop the BIM-based ontology of changes. The developed ontology is comprised of three main classes, six mid-level classes and twenty-two subclasses, which include thirty-six facets in total. Table 2 presents this ontology and briefly illustrates the classes, subclasses and their important facets.

Attributes of Design Changes for Tracking their Consequences

This section elaborates on the attributes or specific characteristics of design changes that should be taken into consideration for tracking design change consequences. In doing so, we use three broad examples of design changes that we documented during the course of the case study and that emphasize object-oriented, integration-related, and adaptation-related characteristics. An earlier version of the first two examples was reported in a conference paper (Pilehchian and Staub-French 2012). For each example, we highlight important characteristics, describe the current practice for managing these changes, investigate the functionality of existing BIM tools, and summarize the findings in terms of requirements for tracking the consequences of design changes automatically.

Example #1: Relocation of Fire-rated Walls

This example emphasizes the object-oriented characteristics of design changes. Due to architectural requirements, the initial arrangement of two-hour fire-rated walls changed slightly. The construction manager noticed the effect of this change on the wall openings, and consequently on the arrangement of their internal framing. Thus, he wanted to know which walls would be affected to modify their assembly prior to installation. To address this issue, we investigated a range of possible methods for tracking such changes in an information model. We also evaluated the capability of a state-of- the-art BIM tool to detect such changes (Pilehchian and Staub-French 2012).

Critique of Current Practice

The implementation process of this change was comprised of two stages. The first stage was identifying any probable consequence of the change, such as the effect of the location of the walls on the openings required for the penetration of mechanical system components. In the second stage, the affected areas were detected by tracking the location of the repositioned walls.

With respect to the identifying the impacts of the change, BIM did not play a significant role in facilitating this process. The consequences of this change were ultimately identified based on the subjective expert opinions provided by different specialists, such as the architect and the construction manager. The minimal role of BIM in this decision making process was primarily due to the limitations of commercially available BIM tools in detecting most ***analytical*** or ***spatial dependencies*** between different building systems and components, e.g. dependency of the arrangement of a wall internal framing on the position of the wall and the wall opening arrangement. Spatial dependencies arise due to the geometry (e.g. shape) or position/location of components in 3D space such as the relationship between the height of a wall and elevation of the roof slab. Analytical dependencies refer to such relationships that link different parts or components of a system (e.g., HVAC system, plumbing system, steel structures) in order to perform a specific function or operation. Regarding the tracking of the location of repositioned walls, despite all the efforts expended on utilizing BIM tools to facilitate and expedite this process, the tracking of these changes were eventually performed by means of traditional methods such as manual comparison of 2D drawings by the construction team. This was mainly due to limited capabilities of the BIM tools for tracking changes between different revisions of the model and their weakness in effectively presenting the results. In addition, the utility of modern 2D software tools, which were specifically designed to track changes in 2D drawings, were another driving factor for the construction team to track changes in 2D rather than in BIM. Further details about this issue are provided in the next section.

Analysis of Existing BIM Tools

In this specific example, we examined a state-of-the-art BIM tool, Solibri Model Checker™ (SMC), to detect changes that occurred in the location of wall openings between two revisions of the model. As shown in Figure 2 (a) – 2 (c), the results of our first attempt indicated that 322 openings were ***added*** *to the model*, 242 openings were ***deleted*** from the model and 61 openings were ***modified*** (Pilehchian and Staub-French 2012).

Further investigation showed that many of the detected additions or deletions were incorrect as the added or deleted openings were identical. This usually happens because the wall and its openings were simply removed and then ***recreated*** at the same location. Such deletions and recreation of identical components might have occurred during modifications of adjacent components or due to ***splitting*** or ***merging*** of the existing components. A number of detected changes were also negligible adjustments in the openings location or geometry, which should not be considered as a change. In terms of modified components, we noticed that a component would be reported as modified if a change were made in any of its attributes, such as ***position***, ***geometry*** or ***specifications***. However, the concern was only about changes in a specific attribute, i.e. position of the component. Thus, many of the detected changes were not the intended target of our analysis.

In another attempt, we used *SMC* to detect changes in the location of the walls, instead of their openings only. To obtain clearer results, we narrowed down our comparison to the east side of the first level of the building. We also focused just on a specific attribute, i.e. the ***position*** of the walls, and excluded changes in any other attributes such as ***geometry*** or ***specifications*** of the walls. The results of this analysis still include many irrelevant changes, which reduce the traceability and reliability of the results. However, due to the capability of this BIM tool to present the footprint of walls in the results, actual changes could be traced, to some extent, by visual comparison of the approximate locations of the changed and existing walls.

To draw a comparison between the functionality of BIM and current 2D tools, we examined a modern 2D software tool, Vico Doc Set Manager™, to track the change of the wall locations between different versions of the drawings. This tool could overlay two revisions of drawings to specify the probable changes. The results of this comparison could be reviewed in three modes: side by side, highlight with color-coding, and slider mode (a slider bar could be dragged across the screen to reveal each of the two overlay drawings). The identified changes then could be marked with cloud marks and an RFI document could be generated for each identified change. Figure 2 (d) shows the result produced by Vico Doc Set Manager™.

Based on the feedback provided by the construction team, the results of Vico Doc Set Manager™ were more practical in terms of addressing the location of changes and traceability of changes. For example, the overlaid layout produced by Vico Doc Set Manager™ contains all grid lines, dimensions and descriptive texts that were already presented in the drawings but *SMC* could only produce a partial footprint of the walls at each floor.

Summary of Findings

The results of our investigation demonstrate that the examined BIM tool can help in tracking the history of changes between revisions of models but owing to its shortcomings, the construction team still preferred to use drawings and 2D software tools to track such changes. The main shortcomings of the BIM tool are as follows:

(1) The studied BIM tool was not able to link the characteristics of ***recreated***, ***merged***, or ***split*** components to the characteristics of the original components. In order to track changes between different revisions of a model, the following transfers of component characteristics should occur automatically:

- ***Recreated*** components should inherit the characteristics of the deleted component.
- ***Merged*** components should inherit the characteristics of their parent components.
- ***Split*** components should inherit characteristics of the original component.

(2) From the perspective of the majority of the construction team, the 3D-presentation of the changes generated by the studied BIM tools was not as effective as the 2D-presentation. Therefore, the construction team preferred to track changes by comparing drawings even though it might need to be done manually. BIM tools such as Autodesk® Revit® are able to create a 2D plan or section view and update it automatically to any change in model components. The presentation of the comparison results would be much clearer if they were marked in any plan or section view.

Example #2: Changes in HVAC Routing

This change focuses on adaptation-oriented characteristics of a design change. To avoid clashes between HVAC ductwork and a column capital in a congested space above the ceiling of the third floor, the routing and the size of the duct needed to be changed. The duct passed over the shower area so a minimum ceiling height of three meters was required (Figure 3). At the time of this change, the construction of the column and the floor slab was completed and the routing of HVAC ducts was being checked for possible clashes with other systems prior to fabricating the duct.

Critique of Current Practice

Clashes between different MEP systems were a major source of changes in the studied project. To resolve such clashes, prior to each meeting, the latest version of architectural, structural, mechanical and electrical models, which were updated separately by the relevant discipline specialists, were integrated into one single model using Navisworks®. Then, an automatic clash detection search was conducted and the results were reviewed during the meetings. Alternative solutions for each clash were explored by the discipline specialists and the overall consequences of each alternative solution were investigated based on subjective expert opinions. It was a significant challenge to develop a solution for each clash due to the numerous constraints imposed by the different engineering disciplines, the accelerated rate of progress in the building's construction, and the geometric and building system complexity of the building. In this particular example, changing the route of HVAC duct and reducing its size in the proximity of the column capital was the final solution. Figure 3 shows the clash report prepared for this change. In this report, the top picture is a screenshot of the Navisworks® model showing different building systems at the location of the clash and the bottom picture shows the annotations on the LCD screen captured in the form of a screenshot. The annotations demonstrate the solution discussed during the BIM meeting. These annotations and the screenshot were captured by the smart tools mounted on each LCD screen in the BIM trailer.

Despite the capabilities of Navisworks® for automatic clash detection, the consequences of the change discussed in this example were ultimately identified based on expert opinion, with little use of BIM in automating any aspect of this resolution process. The primary reason is due to the shortcomings of BIM tools for detecting most ***analytical*** or ***spatial dependencies*** between different building systems represented in multiple discipline-specific models, the issue to be further explained in the next section.

Analysis of Existing BIM Tools

Identification of ***spatial*** and ***analytical dependencies*** is essential for recognizing the components affected by a change and the corresponding chains of successive changes caused by an initial change. However, due to the variety of these dependencies and the technical logic behind them, this process is complicated and challenging.

Spatial dependencies between components are somewhat easier to track and visualize as they are related to the geometry or position of components. For instance, the relocation of the main air supply duct affected secondary ducts ***connected*** to it and the steel hangers that ***supported*** it. It also might influence a number of ***adjacent*** components such as pipes, cable trays, concrete walls and columns. Moreover, as the duct was ***surrounded*** in the small space between the floor slab and the ceiling, a considerable change in the location of the duct could affect its surrounding components such as the ceiling and the concrete floor slab.

Investigation of ***analytical dependencies*** is more complicated as it needs specific technical information and expertise. For instance, prior to the local change in the size of the duct, the new size should be checked against ***mechanical requirements*** such as the air change rate. Similarly, other analytical relationships, such as

architectural, ***operational***, ***maintenance***, ***structural***, and ***electrical requirements*** may also need to be examined, thus necessitating the inputs of different disciplines.

The ***level of propagation*** of a change, and thus its consequences, depends on the extent of such dependencies. The more dependencies to the other components exist, the more ***extensive*** the ***propagation*** and consequences are. For instance, as there are minimal dependencies between the location of a partition wall and the other building systems and components, a slight change in the location of the partition wall causes only ***local propagation*** of the change, which affects the element of the partition wall itself or at most the component that are connected to it. On the other hand, a change in primary design parameters such as the basement height can affect most systems and components of the basement or all other floors thereby causing ***extensive propagation*** of the change. As an example between these two extreme levels, a change in the route of HVAC ductwork on a specific floor can affect other systems of that floor, but not the other floors. Such a ***regional propagation*** is due to several special or analytical dependencies between the changed component and the other systems and components of that region.

It should be noted that a level of propagation is a qualitative identifier used to characterize the complexity and extensiveness of the propagation of changes. It is not directly used in the graph-based approach per se, but it enables to identify the complexity of a graph. It also helps to understand the type of dependencies that would influence the propagation of changes. For example, local propagation is mainly governed by geometrical dependencies whereas analytical dependencies play a more significant role in regional and more extensive types of propagation.

Commercially available BIM tools are able to detect a number of ***spatial dependencies,*** for example, if a change happens in a floor elevation, the length of columns will be automatically updated. They also have some limited capability of detecting and tracking ***analytical dependencies***. Navisworks® and *SMC*, for example, are able to check the clear distance between different components and detect components that do not comply with a minimum preset clearance requirement. While *SMC, with its* rule-based reasoning approach and rule sets is able to interpret typical relationships between components and analyze their interferences, these predefined rules, however, cannot effectively recognize a wide range of logical dependencies

Summary of Findings

Although the predefined rules in *SMC* still cannot recognize a wide range of logical dependencies effectively, they highlight the potential of BIM for such functionality. This capability is crucial in the automatic tracking of the chains of successive changes in BIM. It was found that BIM was of limited use in the automation of identifying the consequences of the change in the HVAC duct route and were ultimately identified based on expert opinion. The capability of Navisworks® for automatic clash detection and the three-dimensional presentation of these clashes facilitates this process significantly.

Example #3: Change in Basement Level

This particular change example illustrates integration-oriented characteristics. Due to the extensive and massive MEP system and limitation of space in the basement and the interstitial level, these areas were extremely congested and, therefore, subject to a vast number of clashes between MEP components and frequent changes. During the early BIM meetings, the design team noticed that the height of these levels should be increased to provide more space and resolve clashes in these areas. Considering the concurrency of the design and construction, the proper timing of this change was imperative because the change in the level of basement could affect the early stages of construction (excavation, shoring and foundation).

Critique of Current Practice

Similar to the previous examples, the consequences of this change was identified based on expert opinion during BIM meetings. The integrated Navisworks® model was reviewed in the meeting. However, this model did not contain any information about the actual construction status of the building components, i.e. whether a component is ***fabricated***, ***erected*** or ***constructed***, that is crucial in decision-making about the proper timing of changes in order to reduce the ***impact of the changes*** in construction. However, due to shortcomings of the utilized BIM tools and the BIM coordination processes in integrating such data into the model, the design team was typically informed about the construction constraints through conventional methods. Consequently, BIM was again of limited use in automating this process. These shortcomings of BIM tools are discussed next.

Analysis of Existing BIM Tools

Parameters such as the elevation of the basement floor are among fundamental design parameters that need to be set in early stages of the design process (***basic design*** or early ***detail design***). However, complexity of the design may cause uncertainty in such parameters and further changes might be required as the design evolves. In terms of the extent of propagation, any change in such basic design parameters will cause ***extensive*** consequences, termed hereafter as an ***extensive change***. Acceptable ***timing*** of such ***extensive*** changes is limited to specific milestones that should be determined based on the ***design*** *or* ***construction status*** of the affected components. In the ***design phase***, ***cost*** and ***time impacts*** of the change depend on the progress in design of other affected components and can be calculated based on the amount of rework required for the relevant modifications. In the ***construction phase***, however, based on the progress in construction of each affected component, the ***cost*** *and* ***time impacts*** associated with the change would increase significantly and sometimes to a degree that the change would no longer be feasible. In this example, the change in the basement elevation would affect the basement and all other components that have a ***spatial*** or ***analytical dependency*** with the basement components, i.e., almost all building components including foundations and the base slab. However, since none of the building components had been constructed at the time of the change, the critical milestone that determines the acceptable timing of the change would be the start of the construction in the construction schedule (foundations/ base slab). Therefore, in order to determine the acceptable timing of such changes, important project data such as ***construction schedule*** and updated ***construction status*** of components should be integrated into the BIM.

To investigate the capabilities of current BIM tools in terms of integration of construction date, such as the updated ***construction status*** of each component, we developed a 4D as-built model during construction of the project. This model only included components that were under construction or already constructed. We gathered the latest construction status of the components during our site visits or through the online pictures taken by the security camera mounted on the roof of an adjacent building. We examined different capabilities of Autodesk® Revit® and Navisworks® for development of such models by utilizing different modeling approaches such as phase-based modeling, definition of groups based on timing of construction, and the use of section boxes to prepare the model.

In our first attempt, we developed a simple 4D model using Navisworks® Timeliner that only contained actual dates and a few tasks corresponding to older, recent and new construction activities. The main challenges in this process were:

- The extensive time required for filtering and separating new constructed components from the other components as the model contained a wide range of tiny secondary components, which were split from main components due to the geometric complexity and irregularity of the structure.
- The necessity of splitting a number of components at "Construction Joints" as Navisworks® is incapable of modifying the geometry of components.
- Revising the as-built model due to revisions in the design model.

To address the first two challenges we utilized Autodesk® Revit® to split components at the construction joints, group secondary components into components and categorized the main components into construction phases. Each time we were updating the model, we were defining a new phase that was then assigned to the individual components that were recently constructed or were under construction.

The update of the as-built model due to the revision in the original BIM was another challenge, and in fact the most significant one. After each revision in the original BIM, a complete iteration of almost the whole process was required to develop a new as-built model based on the new design model. To address this challenge, we used a number of section boxes to split the model into different segments that correspond to different construction phases approximately. Although this method was rough and inaccurate at the component level, it could provide an overall overview of the construction status and its update was significantly quicker and simpler than the previous method.

Summary of Findings

In order to track the consequences of changes, the project data, such as ***construction status*** of different components, ***cost***, ***time schedule*** and ***client's objectives***, need to be integrated with a BIM. Currently, these data, if captured, are stored in different databases that are not linked to the model. Although 4D and 5D modeling help to integrate a portion of this data into the model, the BIM still does not contain all data that is essential to track impacts of changes. In addition, although BIM helped to represent the ***construction status*** of different components, the constraints imposed by such progress in construction were ultimately identified according to subjective expert opinion and BIM was of limited use in automating this process.

A Conceptual Approach for Tracking Changes

In the previous section, we explored the primary characteristics of design changes that are essential for tracking their consequences. These characteristics and their main facets were summarized in Table 2. In this section, we describe our conceptual approach to track design changes that involves three distinct aspects: (1) tracking and tracing dependencies, (2) the deduction of dependencies from a BIM, and (3) the generation of dependency graphs. This section will end with a discussion of the challenges of this approach, and in particular, the challenge of managing change as the model evolves throughout the different phases of design.

Tracing and Tracking Dependencies

In our graph-based approach, the component attributes are linked directly and indirectly through different types of dependencies in a graph. Accordingly, when an attribute of a specific component is changed, different components, which are affected by that change, can be traced in the graph. We then represent these graphs in the form of dependency matrices, which assist in developing a computational approach for automatic propagation of design changes in a BIM. Graph-based approaches to modeling problems have been already used in decision analysis (Morgan and Henrion 1990) and workflow management (Reichert and Dadam 1998) and, specifically, in identifying the implication of changes in construction projects (Isaac and Navon 2009). Isaac and Navon (2013) also explored the extraction of building project information from various documents and databases, including BIM. However, they were more focused on representing the impact of changes across the project, including the project plan and resources, and did not represent the nuances and diversity of design changes and the different types of dependences that exist between building components as we have observed on actual projects.

Different types of dependencies between component attributes exist in a BIM that are useful to track the chain of successive changes created by an initial change in a component attribute. These dependencies can be shown through a graph-based approach. Figure 4 (a) depicts, in the form of a graph, the dependencies between the attributes of the HVAC duct, the column capital, the cable tray and the sanitary pipe, which were discussed in Example #2. The graph nodes represent the component attributes that are linked directly or indirectly through different types of dependencies presented by arrows. The arrow tail specifies the changed component attribute and its head points to the affected component attribute. The three-letter abbreviation beside or over each arrow indicates the type of dependency between the two attributes (refer to Table 2 for the types of dependencies and their abbreviations). If the link represents more than one type of dependency, the abbreviation of each type is indicated and they are separated by a comma. As this figure shows, any changes in the HVAC duct ***geometry*** (i.e. its ***size***) or position will affect the position of the ***adjacent*** cable tray and sanitary pipe due to required minimum clearances (***operational dependencies***) or architectural limitations regarding the ceiling height (***architectural dependencies)***. However, a change in the ductwork does not affect the position of column capitals because column capitals are always located at the top of columns. Likewise, the duct position can be affected by any changes in the position of its adjacent components. Moreover, probable changes in the sizes of the column capital or the cable tray can affect the position of the duct but changes in the sanitary pipe size is usually small and does not affect the position of the duct. Furthermore, the column capital size may change because of changes in the column spans (i.e. its ***position***) or their specifications, such as concrete or reinforcement properties. Moreover, ***mechanical dependencies*** link the ***size*** and the ***position*** of each mechanical system, such as HVAC ducts and sanitary pipes. Likewise, due to ***electrical dependencies***, the ***size*** of cable trays correlates with their ***specifications***, such as their capacity in terms of the weight of cables they can support.

The developed dependency graph can be represented in the form of a dependency matrix in order to communicate the specifications of the graph to a computer, which could assist with the automatic tracking of changes. In essence, the rows and columns in the dependency matrices represent the nodes in the graphs and entries in the matrix indicate whether a link exists between the nodes. Figure 4 (b) depicts the development of the dependency matrix for the provided example. The size of the matrix is 12×12 as it represents the dependencies between four components (n=4) and the characteristics of each component are controlled by three sets of attributes (m=3). Likewise, we can integrate all component-based dependency graphs and develop a dependency network, which represents the relationships between all components attributes in the network.

Deduction of Dependencies from a BIM

In order to generate dependency graphs, information related to the components and their dependencies (i.e., spatial and analytical dependencies) needs to be identified and stored in a common model. Component specific information and spatial dependencies between components can be extracted from BIM. IFC is a suitable platform for the extraction of such object-oriented building information. For the extraction of spatial dependencies, modern BIM tools and related research (e.g., Nguyen et al. 2005) provide algorithms for the deduction of the relevant topological relations, such as adjacency, containment, separation, etc. The extracted information, however, needs to be stored and manipulated in the form of a matrix structure. A unique ID number would be assigned to each component attribute, which identifies its corresponding entry in different vectors of data.

In terms of deducing analytical dependencies, these dependencies are currently not stored in BIM applications, and we suggest that a hybrid of object-oriented and subject-oriented modeling approaches be utilized to store them. Design requirements will be defined as different design subjects (e.g., structural and mechanical perspectives) and each design subject will consist of the dependencies that satisfy a set of specific project requirements. Using this approach, the dependency data will not be lost when the original objects are replaced with their subcomponents while the model progresses to the next Level of Development/Detail (LOD) because the subcomponents inherit the dependencies that already exist between their parent components. We will discuss this further in a subsequent section on reasoning about model evolution.

The Generation of Dependency Graphs

Once all dependencies are derived, the dependency graph will be formed. The graph is manipulated in the form of a matrix structure based on the concept of adjacency matrix and relevant graph-theoretic data structure. Any proposed change in different component attributes will be presented as an initial "change vector." The change vector is a matrix with one row only (row vector). Each entry of this vector has a logical value (0 or 1) that defines whether each component attribute has changed or not. A change vector is defined as:

C = Change Vector = {[C1], …., [Cn]}

[Ci] = Change vector for component i = $\{c_1, \ldots, c_j, \ldots, c_m\}$

$$c_j = \begin{cases} 1: & \textit{if attribute } j \textit{ of the component } i \textit{ has changed} \\ 0: & \textit{if attribute } j \textit{ of the component } i \textit{ has not changed} \end{cases}$$

This initial change vector only determines the initial change and not the changes that happen as the consequence of this initial change. Therefore, we call it C_0. The effect of this change on the other component attributes can be determined by the product of multiplying this vector and the Dependency Matrix:

$$C_1 = C_0 * D$$

The calculated change vector (C_1) indicates the direct effect of the initial change vector (C_0) and shows the first group of affected component attributes in the series of successive changes caused by the initial change. These new changes also generate a second group of successive changes. The attributes affected by these successive changes can also be determined by the product of C_1 and the Dependency Matrix as follows:

$$C_2 = C_1 * D$$

The chain of successive changes (C_1, C_2, ...,) are thus determined by multiplying the initial change vector and the Dependency Matrix (D) and iterating this operation by using the product of the first operation as a new proposed change until no new attribute is affected by the last group of effected attributes ($C_i = C_{i-1}$). The product of the first operation will identify the components directly affected by the proposed change, similar to clustering algorithms, such as Iterative Conductance Cutting (ICC) algorithm (Kannan et al. 2004), and each further iteration will add new components to the chain of successive changes until the propagation of changes is completed. After the completion of many iterations, similar to path search algorithms, such as Dijkstra's shortest path algorithm (Dijkstra 1959), all possible change paths will be determined. More detailed explanation about this process is provided in Pilehchian (2012).

While the computer implementation of the adjacency matrix will likely have more computational costs, tracking changes using the adjacency matrix is likely to be more efficient than the existing clustering and path search algorithms. Isaac and Navon (2013) reported that poor results were obtained when the ICC clustering algorithm was applied to what they called as the Project Connectivity Model (PCM), which is the graph-based model that included complex project information. And the clustering algorithm was generally not able to decompose the graphs into distinct clusters at all (Issac and Navon 2013). However, the adjacency matrix approach does not appear to have such a limitation. Moreover, entries in the adjacency matrix can be weighted to represent particular consequences of changes, such as costs or stochastic data (e.g., probability of changes and risks).

Challenges and Future Directions

In the previous section, we investigated the dependencies between component attributes and introduced dependency graphs and their matrix representation, which could provide a basis for automatically tracking these dependencies in BIM. In this section, we discuss the more challenging and complex issue of tracking changes as the model evolves throughout the different phases of design. Specifically, we focus on the changes in the quantity and quality of the dependencies throughout the evolution of BIM and investigate how this evolution affects the tracking of changes in BIM.

Figure 5 illustrates the evolution of a BIM and depicts the formation of different types of dependencies between component attributes throughout this evolution process. In the early stages of the project (***feasibility*** and ***conceptual design***), the information model, if it exists, only includes very basic aspects of design. The ***conceptual model*** may include basic components, such as spaces, areas, floors and the main components of the structural systems and the building envelope. In this stage, incorporation of changes in design needs minimum effort and the majority of available BIM tools are able to implement them automatically since the number of components and their spatial and analytical dependencies are limited. During ***basic design***, models include the majority of main components, such as column, beams, floor slabs, doors and windows. However, models include only ***basic attributes*** of these components (***geometry***, ***position*** and probably ***material*** type) and models do not include most detailed attributes of these components (***elements***, ***semantic properties*** and ***material specifications***). During the basic and detailed design phases, the focus of modeling is on development of systems and components respectively. Thus, the increase in the number of components and component attributes causes an exponential increase in the number of ***spatial*** and ***analytical dependencies***. This reduces the capability of BIM tools in automatically tracking the consequence of changes significantly as commercially available BIM tools can only identify a limited range of ***spatial dependencies*** and do not recognize most ***analytical dependencies.*** This limitation is further exacerbated when the LOD increases during the design process as more components, component attributes and elements are created and thus their ***dependencies*** become more complicated. This increases the ***time*** and ***cost*** of incorporating changes in the model and in the design. The practical number of dependencies that can be manipulated through the approach suggested in this paper depends on the computational capacity of the utilized computer (CPU speed and memory). Given the conceptual stage of this research and the lack of empirical evidence, we are unable to provide the exact relationship between the number of components/attributes and their dependencies with the performance of BIM tools for automatically tracking the consequences of design changes.

The dependencies formed during such evolution link either the components of the model at a particular design stage or the components of different revisions of the model created while the model progresses from one LOD to the next. To distinguish between these two types of dependency, we call the former ***intra-model dependency*** and the latter ***inter-model dependency***. Overall, compared to the effect of ***intra-model dependencies,*** propagation of changes through ***inter-model dependencies*** affects a wider range of components. Thus, identifying ***inter-model dependencies*** is crucial in controlling the ***impacts of changes*** by managing their ***timing***. To elaborate on this issue, we consider the change in the basement height that was discussed in Example #3. The diagram presented in Figure 5 depicts a part of the chain of successive changes created by the initial change in the basement height. This change is propagated through ***intra-model*** (shown by green solid lines) and ***inter-model dependencies*** (shown as red dashed lines). The elevations of floors and basement height are among the primary parameters that are supposed to be finalized during the ***basic design*** phase. If such primary parameters change during further stages of the design, such as ***detailed design***, it causes ***extensive*** successive changes in a wide range of the model components created as subcomponents of the basement space, such as basement walls and foundations. This ***extensive propagation*** of changes is due to formation of ***inter-model dependencies*** between the primary component (basement space) and secondary components with higher LOD created as supplementary to this primary component during the evolution of the model from basic design to detailed design stage (e.g. basement walls and columns). Likewise, a change in the basement height at the fabrication or construction stage propagates more extensively due to formation of ***inter-model dependencies*** between the components created at detailed design stage and the vast numbers of supplementary component and elements (e.g. formwork and reinforcing bars) created by the progress to higher LOD.

The graph-based approach we illustrated in the previous section could be utilized as a base for tracking changes propagated through ***intra-model dependencies***, when we only need to track the dependencies between components of the same model at a specific LOD. However, this approach is not effective when the model progresses from one LOD to the next LOD unless we integrate the dependency information stored in the older model into the new one. This means that the secondary components created as supplements to the primary components with lower LOD should inherent the dependencies which were already formed between the primary components. For example, the basement space (a mass component) is the primary component with the lowest LOD created at the basic design stage. Basement walls and the basement floor are basement subcomponents created while the model progresses to the detailed design stage. Thus, these subcomponents inherit the dependencies that already existed between the original component attributes, such as the dependencies between parameters related to the basement geometry and position. Likewise, the reinforcing bars and the formwork of the basement floor and its walls are subcomponents with the highest LOD, which are created at the construction stage. These subcomponents inherit the dependencies that already exist between the basement floor and its walls while the model evolves to the construction stage.

As we discussed in Example #2 of the case study, BIM potentially is able to store the dependencies between components of the same LOD, i.e. intra-model dependencies. However, commercially available BIM tools are not able to detect all dependencies between components and even if they detect them, they store them as object-oriented data. Thus, the dependency data will be lost as soon as the original objects, i.e. primary components, are replaced with their subcomponents while the model progress to the next LOD. A subject-oriented modeling approach, as discussed previously, could be a solution to address this shortcoming. This approach has been introduced in the design of software systems (Clarke et al. 1999) and proposed for the development of change management tools (Isaac and Navon 2009). In this approach, the model is divided into design subjects (e.g., structural integrity and HVAC) and each design subject consists of the dependencies that should satisfy a set of specific project requirements. For example, the design subject of basement sizing consists of adjusting the position and geometry of its components (such as its floor slab and walls) to satisfy the requirements of minimum space for HVAC, plumbing and electrical systems. Accordingly, a subject-oriented modeling approach is capable of interconnecting dependencies between components of different systems with different LODs throughout the evolution of design. Hence, adopting a hybrid method consisting of both object-oriented and subject-oriented approaches can expand the capabilities of BIM tools such that they can propagate changes through different versions of models prepared at different stages of the design.

Conclusions

In this research, we examined change management in the context of a multi-disciplinary collaborative BIM environment during the design and construction of a fast-track building project. We identified the common characteristics of design changes encountered throughout the project design and construction to develop an ontology of design changes. The ontology provides a common understanding of the characteristics of design changes for practitioners who need to use BIM tools for managing changes, and identifies relationships and dependencies between different facets of changes and their impact on the project costs and schedule.

During the course of this study, we examined the capability of three, i.e., Autodesk® Revit®, Navisworks®, Solibri Model Checker™, in the context of BIM-based change management and investigated their potential benefits and shortcomings in terms of the management of changes. While state-of-the-art BIM tools are helpful in tracking the history of changes between revisions of models, they have a number of shortcomings to track changes. SMC, for example, was not able to link the characteristics of ***recreated***, ***merged***, or ***split*** components to the characteristics of the original components thereby limiting tracking changes between different revisions of a model. Furthermore, state-of-the-art BIM tools still cannot recognize a wide range of logical dependencies of design changes effectively. In order to track the consequences of changes, the project data, such as ***construction status*** of different components, ***cost***, ***time schedule*** and ***client's objectives***, need to be integrated in a BIM model. Currently, these data, if captured, are stored in different databases that are not linked to the model. Although 4D and 5D modeling help to integrate a portion of these data into the model, but BIM still does not contain all data that are essential to track impacts of changes. As a result, practitioners still prefer to use drawings and 2D software tools to track changes and rely heavily on subjective expert opinions for decision making.

Different types of dependencies between component attributes exist in a BIM, within the attributes of a component or between the attributes of different components. A graph-based approach can be very useful to show and trace the dependencies between different component attributes and to track the chain of successive changes created by an initial change in a component attribute. Our approach has the potential of addressing the challenging task of re-establishing the dependencies and updating the dependency graph after each revision in BIM during the design. The real challenge for tracking changes remains as the model evolves or progresses from one LOD to the next. BIM potentially is able to store the dependencies between components of the same LOD, i.e. ***intra-model dependencies.*** Moreover, identifying the dependencies of components of different revisions of the model while the model progresses from one LOD to the next (i.e. ***inter-model dependencies)***, is still a challenge with current BIM tools.

Additional research is required to further extend the ontology of design changes developed in order to validate its breadth, depth, and completeness. Further research is also required to verify the different spatial and analytical dependencies and to identify other important attributes that might be required for automatic recognition of these dependencies in an information model. Finally, the graph-based approach needs to be implemented and tested in a BIM environment to examine its effectiveness to manage and track changes throughout the design and construction process, which will be the focus of the next phase of this research.

References

AIA. 2008. Document E202 – Building Information Modeling Protocol Exhibit, American Institute of Architects, Washington, DC.

Akcamete, A., Akinci, B., and Garrett, J. H. 2009. Motivation for computational support for updating building information models (BIMs). *International Workshop on Computing in Civil Engineering*, Austin, Texas.

Arayici, Y., Coates, P., Koskela, L., Kagioglou, M., Usher, C., and O'Reilly, K. 2011. Technology adoption in the BIM implementation for lean architectural practice. *Automation in Construction*, **20**(2): 189-195.

Aslani, P., Griffis, F., and Chiarelli, L. 2009. Building information model: The role and need of the constructors. *Construction Research Congress 2009*, Seattle Washington, pp. 467-476.

Clarke, S., Harrison, W., Ossher, H., and Tarr, P. 1999. Subject-oriented design: towards improved alignment of requirements, design, and code. *14th ACM SIGPLAN Conference on Object-oriented Programming Systems, Languages and Application*, A. Michael Berman (Ed.), ACM, NY, pp. 325-339.

Construction Industry Institute (CII). 1994. Project Change Management, Publication, 43-1, Austin, Texas.

Construction Industry Research and Information Association (CIRIA). 2001. Managing Project Change - a Best Practice Guide, UK.

Dijkstra, E.W. 1959. A note on two problems in connexion with graphs, *Numerische Mathematik*, **1**(1): 269–271.

Hajian, H., and Becerik-Gerber, B. 2009. A research outlook for real-time project information management by integrating advanced field data acquisition systems and building information modeling. *International Workshop on Computing in Civil Engineering*, Austin, Texas, pp. 83-94.

Hao, Q., Shen, W., Neelamkavil, J., and Thomas, R. 2008. Change management in construction projects. *Institute for Research in Construction, National Research Council Canada*, London, Ontario, *NRCC-50325*.

Hegazy, T., Zaneldin, E., and Grierson, D. 2001. Improving design coordination for building projects. I: Information model. *Journal of Construction Engineering and Management*, **127**(4): 322-329.

Ibbs, C., Wong, C., and Kwak, Y. 2001. Project change management system. *Journal of Management in Engineering,* **17**(3): 159-165.

Isaac, S., and Navon, R. 2008. Feasibility study of an automated tool for identifying the implications of changes in construction projects. *Journal of Construction Engineering and Management*, **134**(2): 139-145.

Isaac, S., and Navon, R. 2009. Modeling building projects as a basis for change control. *Automation in Construction*, **18**(5): 656-664.

Isaac, S., and Navon, R. 2013. A graph-based model for the identification of the impact of design changes. *Automation in Construction*, **31**: 31-40.

Kannan, R., Vempala, S., and Vetta, A. 2004. On clusterings: good, bad and spectral, *Journal of the ACM*, **51**(3): 497–515.

Koch, C., and Firmenich, B. 2011. An approach to distributed building modeling on the basis of versions and changes. *Advanced Engineering Informatics*, **25**(2): 297-310.

Lee, M., Hanna, A., and Loh, W. 2004. Decision tree approach to classify and quantify cumulative impact of change orders on productivity. *Journal of Computing in Civil Engineering,* **18**(2): 132-144.

Lee, S., and Pena-Mora, F. 2007. Understanding and managing iterative error and change cycles in construction. *System Dynamics Review*, **23**(1): 35-60.

Mokhtar, A., Bedard, C., and Fazio, P. 1998. Information model for managing design changes in a collaborative environment. *Journal of Computing in Civil Engineering*, **12**(2): 82-92.

Morgan, M. G., and Henrion, M. 1990. Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis, Cambridge University Press.

Motawa, I.A., Anumba, C.J., Lee, S., and Peña-Mora, F. 2007. An integrated system for change management in construction. *Automation in Construction*, **16**(3): 368-377.

Navon, R. and Isaac, S. 2009. An automated tool for identifying the implications of changes in construction projects. *Construction Research Congress 2009*, Seattle Washington, pp. 1-10.

Nguyen, T. H., Oloufa, A., and Nassar, A. K. 2005. Algorithms for automated deduction of topological information. *Automation in Construction* **14**(1): 59–70.

Nour, M., and Beucke, K. 2010. Object versioning as a basis for design change management within a BIM context. *International Conference on Computing in Civil and Building Engineering*, W. Tizani (ed.), (ICCCBE-XIII), Nottingham, UK.

Pilehchian, B. 2012. Change management with building information modeling: A case study. M.Sc. thesis, Department of Civil Engineering, The University of British Columbia, Vancouver, B.C.

Pilehchian, B., and Staub-French, S. 2012. Change management with building information models: A case study. *2012 Construction Research Congress,* West Lafayette, IN, pp: 1182-1191.

Reichert, M., and Dadam, P. 1998. ADEPTflex - Supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems,* **10**(2): 93-129.

Soh, C., and Wang, Z. 2000. Parametric coordinator for engineering design. *Journal of Computing in Civil Engineering*, **14**(4): 233-240.

Wang, H., Akinci, B., and Garrett, J. H. 2007. Formalism for detecting version differences in data models. *Journal of Computing in Civil Engineering, 21*(5): 321-330.

List of Figures

Table 1. Taxonomy of first twenty recorded changes

No	Change Subject	Date	Initiating Department	Component Type	Was it Modeled?	Changed Attributes	Spatial Dependencies	Analytical Dependencies	Change Type	Extent of Propagation
1	Plumbing specification	2010-09-08	Mechanical	Document	No	Specification	None	MEC, ARC	MOD	Extensive
2	Plumbing Penetrations	2010-09-14	Mechanical	Piping/ Penetration	Yes	Position: CRD	ADT	MEC, ARC	MOD	Local
3	Elevator shaft	2010-09-17	Structural	Opening/ Floor slab	Yes	Geometry: SHP, DIM	CNT, ADT	STR, MEC	ADD	Local
4	Column at gridline 1	2010-09-17	Structural	Column	Yes	Position: CRD	CNT, ADT	STR, ARC, MEC, ELC, OPR	MOD	Regional
5	Pull Pit	2010-09-17	Structural	Pit: Wall, Floor	Yes	None	CNT, ADT	STR, ARC, ELC, OPR	ADD	Regional
6	Structural IFC revision	2010-10-01	Structural	Many	NA	None	NA	NA	MOD	Local
7	Column size	2010-11-01	Structural	Column	Yes	Geometry: DIM	ADT, CNT	STR	MOD	Regional
8	Column orientation	2010-11-01	Structural	Column	Yes	Position: ORN	ADT, CNT	STR	MOD	Regional
9	Column rebar	2010-11-01	Structural	Column	No	Specification: ELM	None	STR	MOD	Local
10	Top of wall	2010-11-02	Architectural	Wall	Yes	Geometry: DIM	ADT	ARC, MEC, ELC	MOD	Local
11	Elevator #5 opening	2010-11-08	Architectural	Wall	Yes	Geometry: SHP, DIM	CNT	ARC, STR	MOD	Local
12	Slab Acoustic Isolation joint	2010-11-09	Architectural	Joint	No	Geometry, Position	CNT	MEC, ARC	MOD	Local
13	Slab opening at A.IS. Joint	2010-11-09	Architectural	Slab	Yes	Geometry: SHP, DIM	None	MEC, ARC	MOD	Local
14	Location of plumbing wall	2010-11-10	Architectural	Wall	Yes	Position: CRD	ADT	MEC, ARC	MOD	Local

15	Slab Openings- Lecture hall	2010-11-16	Mechanical	Opening/ Floor slab	Yes	Geometry: SHP, DIM	CNT, ADT	MEC, STR, ARC	ADD, MRG	Local
16	Slope of Floor Slab	2010-11-21	Structural	Floor slab	Yes	Geometry: DIM	SRB	ARC	MOD	Local
17	Louver Block-out	2010-12-13	Mechanical	Wall/ openings	Yes	Geometry: SHP, DIM	CNT, ADT	STR, MEC,ARC	ADD	Local
18	Partition Layout	2010-12-14	Architectural	Partitions	Yes	Geometry, Position	ADT	ARC, MEC	MOD, SPL	Local
19	Ceiling Height	2011-04-21	Architectural	Ceiling	Yes	Position	CNT, ADT	ARC, MEC, ELC, OPR	MOD	Regional
20	Cable Tray Relocation	2011-04-21	Electrical	Cable tray	Yes	Position	CNT, ADT	ELC, ARC	MOD	Local

Note: Abbreviations related to different types of dependencies are defined in Table 2

Table 2. An ontology of design changes

Classes & Sub-classes			Facets: Description/Example
Object-Oriented	**Change Type**	Addition (ADD)	*Creating a new component*
		Deletion (DEL)	*Deleting an existing component*
		Modification (MOD)	*Modification in attributes of an existing component*
		Recreate (REC)	*Deleting a component then adding a new one with similar attributes*
		Merge (MRG)	*Combining two or more components to create a new component*
		Split (SPL)	*Dividing a component into two or more components*
	Changed Component Attributes	Geometry	Shape (SHP): *cubic, cylindrical, rectangular, plate*
			Dimensions (DIM): *Shape, length, width, thickness, diameter, slope*
		Position	Coordinates (CRD): *X ,Y ,Z*
			Orientation (ORN): *Rx, Ry, Rz*
		Specification	Material (MAT): *concrete, mild steel, galvanized steel*
			Elements (ELM): *Stud, Rebar: size, shape, arrangement*
			Semantic Properties (PRP): *Fire-rating, acoustic, water proof*
Adaptation Oriented	**Dependencies between Components**	Spatial Dependencies	Connected To (CNT) : *column and floors, main and secondary ducts*
			Adjacent To (ADT): *duct and adjacent pipes, duct and ceiling*
			Supported By (SPB): *duct and steel hangers*
			Surrounded By (SRB): *duct and false ceiling/ plenum area*
		Analytical Dependencies	Structural (STR): *sleeves size and arrangement of rebar*
			Architectural (ARC): *room functionality and exposed duct*

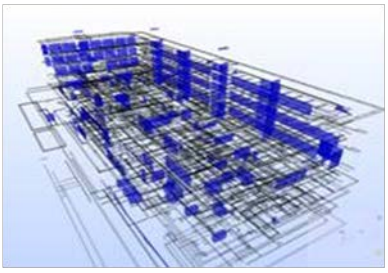
			Mechanical (MEC): *size and location of air supply duct*
			Electrical (ELC): *size of cable tray and motor power*
			Operational (OPR): *clearance around a pipe*
	Propagation of Changes	Level of Propagation	Extensive: *Substantial effects on many components*
			Regional: *Affect several adjacent components*
			Local: *Minimal effect on other components*
		Type of Dependencies	Intra-model: *between components with same LOD*
			Inter-model *: between components with different LOD*
Integration Oriented	**Change Timing**	Conceptual design	*During early decision making about the primary design aspects*
		Basic design	*During early stages of the design but prior to the full extended design*
		Detail design	*During the detailed design but prior to any procurement/ construction*
		Procurement	*After Purchase Order but prior to fabrication*
		Fabrication	*After Fabrication but prior to erection*
		Construction	*After commence of construction*
	Change Impact	Cost	*The impacts of the change on the project cost*
		Time schedule	*The impacts of the change on the project time schedule*
		Client’s Objective	*The impacts of the change on the client’s objectives*

(a)

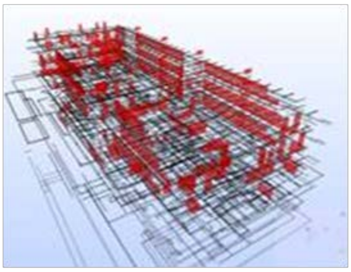
(b)

(c)

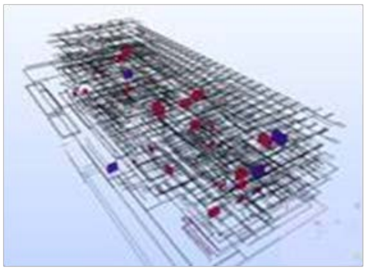
Figure 1. Rendered 3D view of the building (Source: Saucier + Perrotte Architects | Hughes Condon Marler Architects) **(a)** and snapshots of BIM coordination meetings in the architect's office during the early design stage **(b)** and research-based BIM trailer at the construction site **(c)**



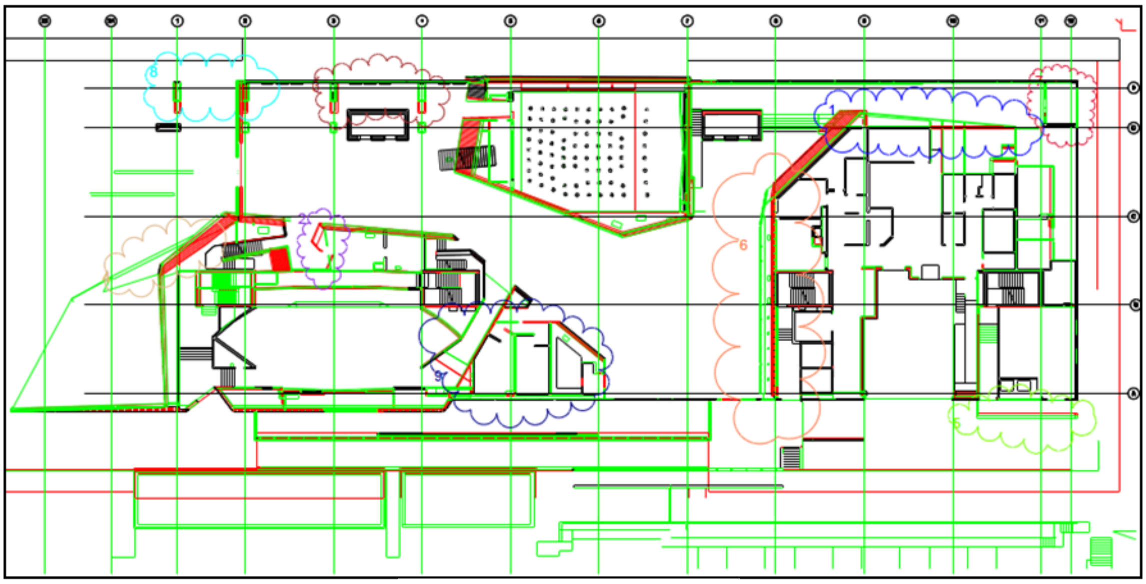
(a)
Added Openings: 322



(b)
Deleted Openings: 242



(c)
Modified Openings: 61



(d)

Figure 2. Track of changes in openings - Solibri Model Checker™ **(a)**, **(b)** & **(c)** and on 2D (drawings) - Vico Doc Set Manager™ **(d)**

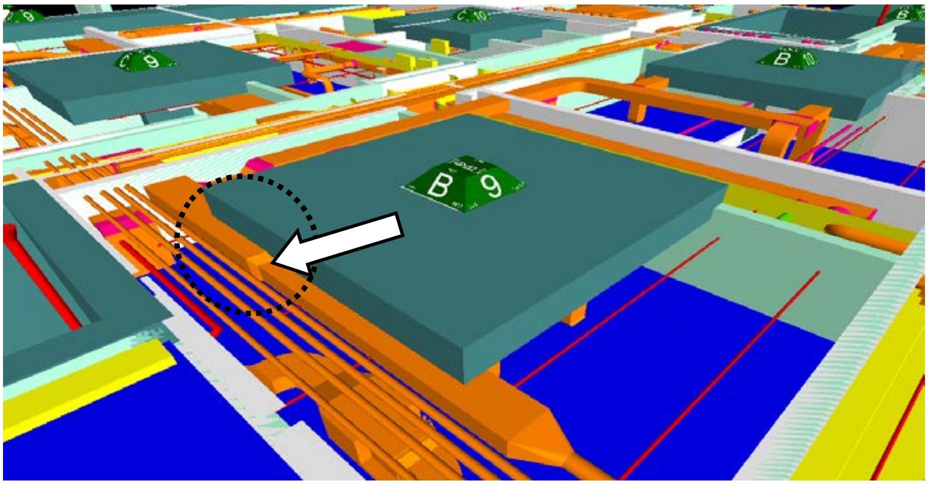
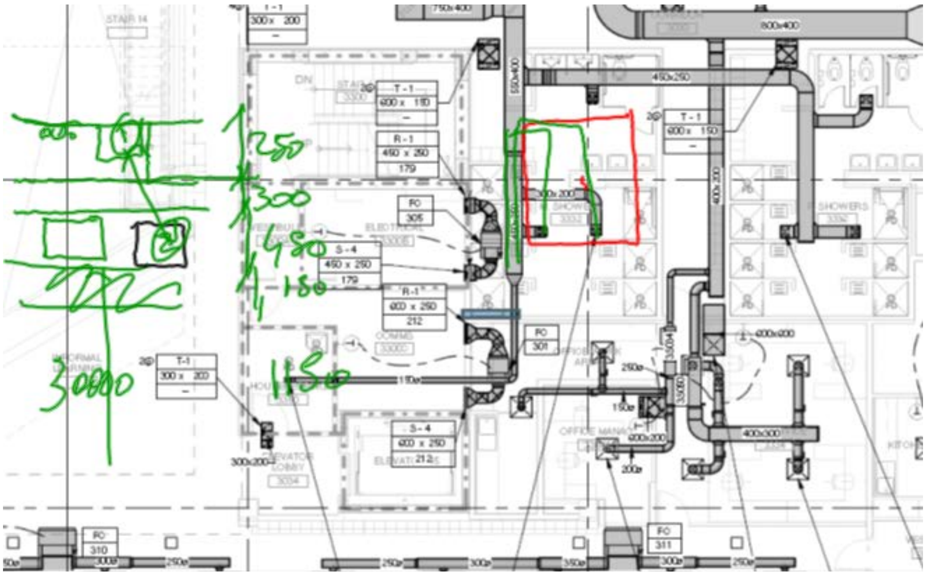
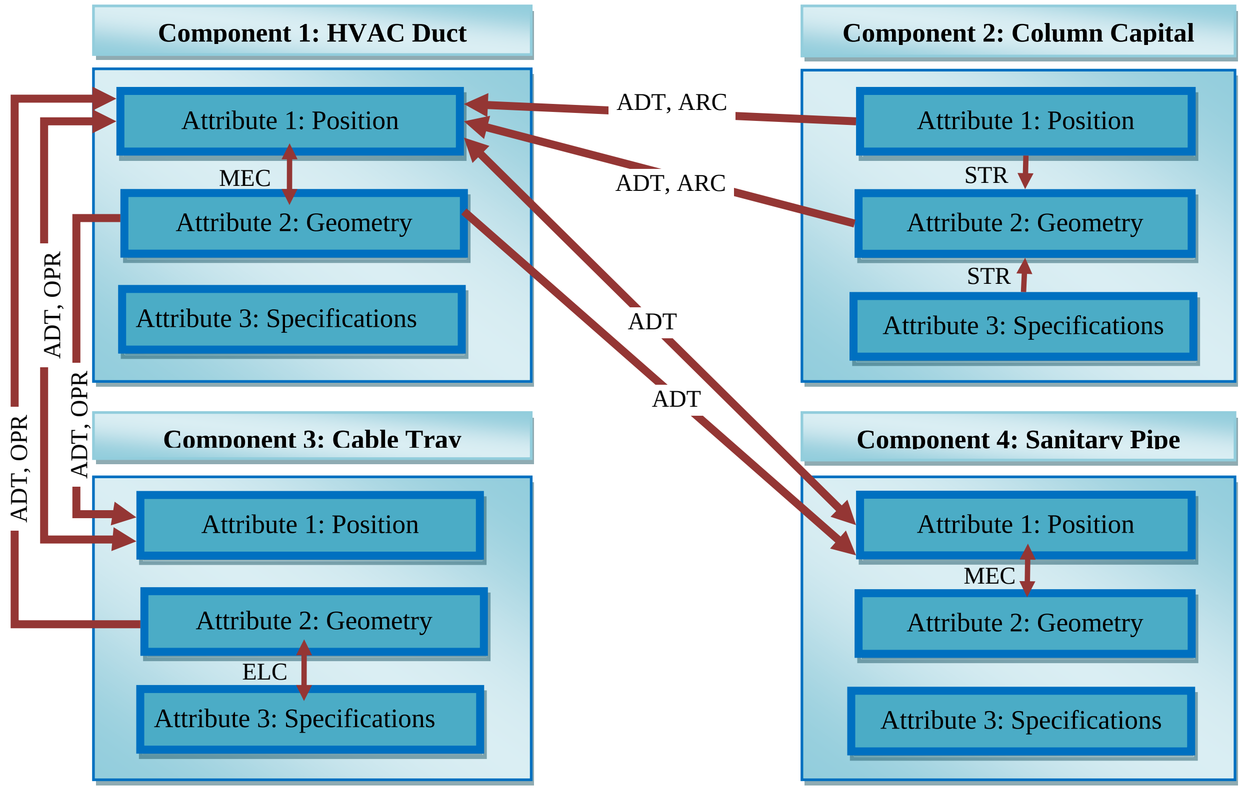
Clash ID	Status	Location	Received Date	Resolved Date
Level 3 East-CL04	Resolved	Level 3 East- Adjacent to Electrical Room- Col. B9	21 April, 2011	26 April, 2011
Reference Drawings		A2.13 E4.04 M2.08 M2.09 P2.05		
Clash Description		Ductwork clashes with column capital. Ceiling Height of 3000 mm is desired in adjacent shower area.		
Sketch Plan/ Section/ 3D Screenshot				
Solution / Changes		Ductwork and plumbing routes will change and move toward electrical room. Size of duct will reduce at column capital.		
Sketch Plan/ Section/ 3D Screenshot				

Figure 3. Clash report relating to the conflict between ductwork and a column capital



(a)

$$D = \begin{bmatrix} |D11| & \cdots & |D1n| \\ \vdots & |Dij| & \vdots \\ |Dn1| & \cdots & |Dnn| \end{bmatrix} = \begin{bmatrix} \begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} & \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \\ \begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} & \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \\ \begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix} & \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \\ \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} & \begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \end{bmatrix}$$

$$|Dij| = \begin{bmatrix} d11 & \cdots & d1m \\ \vdots & \mathrm{dpq} & \vdots \\ dm1 & \cdots & dmm \end{bmatrix} \mathrm{d_{pq}} \;\; ; \quad \mathrm{d_{pq}} = \begin{cases} 1: & \text{if change in attribute } p \text{ of component } i \text{ affects attribute } q \text{ of component } j \\ 0: & \text{if change in attribute } p \text{ of component } i \text{ does not affect attribute } q \text{ of component } j \end{cases}$$

(b)

Figure 4. Dependency graph corresponding to Example #2 **(a)** and its matrix representation **(b)**

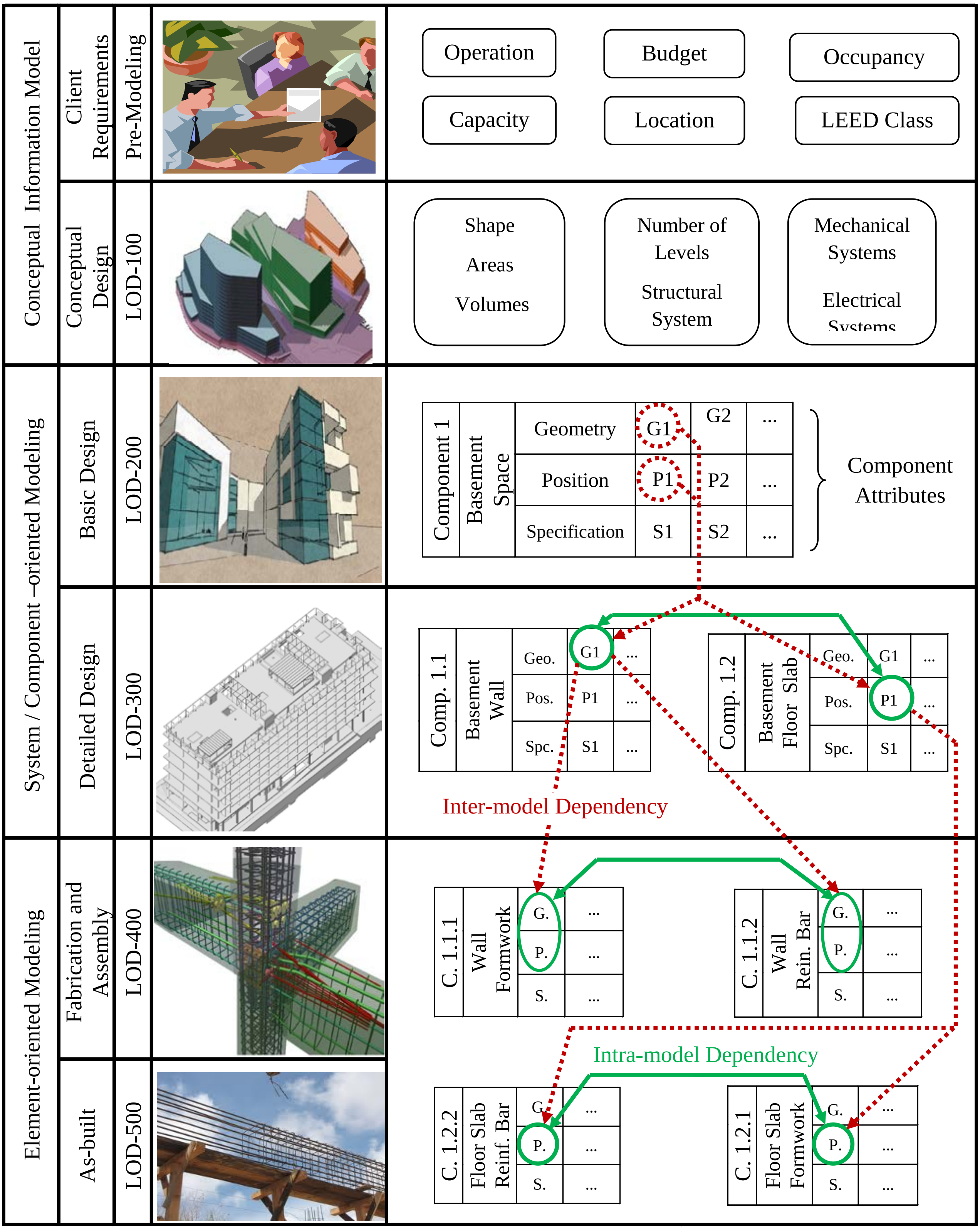


Figure 5. Formation of inter-model dependencies throughout BIM evolution [For the definition of LOD refer to AIA (2008), Document E202]