

## INTEGRATION OF SEMANTIC TEMPORAL INFORMATION IN BIM USING ONTOLOGIES

Albrecht Vaatz<sup>1,2</sup>, Al-Hakam Hamdan<sup>3</sup>, Nidhal Al-Sadoon<sup>1</sup>, Martin Wogan<sup>1,4</sup>, Karsten Menzel<sup>1</sup>

<sup>1</sup>Technical University, Dresden, Germany

<sup>2</sup>DB Bahnbau Gruppe GmbH, Berlin, Germany

<sup>3</sup>A+S Consult GmbH, Dresden, Germany

<sup>4</sup>B/M Consult GmbH, Braunschweig, Germany

### Abstract

Model-based planning requires an extensive amount of information about the environment, inventory and requirements of the planned object. Thereby, temporal information is important to be considered for planning, construction and maintenance, but in current approaches temporal information is only defined through non-semantic datasets, which lack contextual meaning. To semantify time-data that is used in Building Information Modeling (BIM) projects, the authors propose the concept of combining existing models using Information Containers for linked Document Delivery (ICDD) and adding an Ontology for Chronological Construction Processes (OCCP) to record and implement structured temporal information of the entire life cycle and use it from the start of planning until deconstruction and even beyond.

### Introduction

Obtaining all relevant information is a recurring challenge in planning projects. Difficulty and effort scale with the increasing complexity of the project, e.g., with the number of people involved or the number of disciplines to be considered. The infrastructure sector places special demands on planners due to its public relevance and the resulting large number of stakeholders involved. To reduce the planning effort, the efficiency of information procurement must be optimal, because it is the only way to avoid iteration loops that usually result from poor data bases or poor communication.

One approach for creating an optimal information base is the development of comprehensive As-Built databases in the form of digital twins of the so-called As-Built or As-Is models in the context of BIM. A modularized structure of individual, self-contained models and the subsequent, georeferenced linking of these models enables the step-by-step construction of large models of high complexity. Nevertheless, even a 'perfect' digital twin still harbors the risk of planning errors, for example when unknown and therefore unmodeled old structures (e.g., old foundations or pipelines of a preexisting construction left remaining in the ground) only come to light during construction.

Working with models for planning and operation, a lot of temporal information can be added to models, but so far this information is not semantically structured. Therefore,

interpretation of these data is either processed manually by human experts, which could result in a time-consuming and error prone task or parsed by software tools that need to know the exact data structure in which the time information is defined. The latter solution requires continuous updating of the parser whenever changes are made to the interpreted data structure.

Also, in the course of planning, several versions are often created in order to record different variants of planning and construction states. When As-Is models are used in operation, several versions of a structure or area may also exist, since the model is used as an As-Built document and is continuously updated. Maintaining multiple versions may result in greater storage requirements and can lead to problems identifying specific conditions at a given point in time. This problem can be addressed by logging versions or using version control systems, but the more different points in time or versions are archived, the larger the number of versions that need to be managed. In the context of planning, versioning is useful because all versions that are no longer needed can be deleted once planning is complete. In relation to As-Is models, versioning is not optimal for the above reasons.

The introduction of chronological models provides an approach to counter the described problems by reducing the number of versions needed to differentiate between varying states of constructions and components. In chronological models, temporally relevant information is added to the model with the help of an ontology, making versioning superfluous and semantically enriched. At the same time, an unambiguous mapping of construction model states at arbitrary points in time is enabled and redundant data storage is favored, which can ensure a comprehensive provision of relevant planning information.

### State of the art

#### Implementation of temporal information

Possibilities for defining time-dependent information already exist in Industry Foundation Classes (IFC) of ISO 16739-1, which is an open BIM standard. A specific entity for defining time explicitly is provided in the form of IfcTimeSeries and its related subclasses. Thereby, an instance of IfcTimeSeries defines a time period that is detailed through multiple points in time. However,

IfcTimeSeries is intended to be linked only with external references outside of the IFC model that could be referenced via an Unique Resource Identifier (URI), e.g., dictionaries or documents. Furthermore, IfcTimeSeries just provides functionality for defining raw time data, but not assigning this data to a semantic meaning, e.g., the lifetime of a building or the point in time of its deconstruction. Therefore, meaningful temporal information is defined as attributes of other IFC entities. For instance, process related entities that are subtypes of IfcProcess, such as IfcEvent or IfcTask, possess attributes for explicitly binding a point of time to an event or task. By utilizing these entities, it is possible to structure various processes in the construction life cycle, such as the design or inspection and annotate them further information. However, since no entities or attributes for specific life cycle processes are supported by IFC, e.g., the submission and review of a design document, the interpretation of this information must be processed either manually by humans or by software applications that are aware of the concrete meaning of the non-standardized annotations. An approach that proposes utilizing IfcProcess and its subtypes for defining time-related information about structural damage inspections has been developed by Artus (2021). In his publication Artus (2021) also discusses the advantages and drawbacks of using the existing IFC without extensions for describing time information. The main advantage of this approach is that the model would be directly readable by many IFC-supporting applications. However, the main drawback would be that some entities would not be used in the semantically correct context. To overcome these drawbacks, IFC extensions have been proposed, e.g., by Tanaka et al. (2018) for describing component degradation over time or by Eftekhari et al. (2018) to develop an IFC-based system for fire emergency real-time management. In this regard, new IFC entities were developed for describing a certain object or behavior in a specific point in time. Similarly to IfcEvent or IfcTask, the definition of time is assigned to the entity instances via a corresponding attribute. Nonetheless, a processing of these extensions in publicly available BIM applications is currently not supported.

A valid option for describing time semantically, so that software applications could interpret this information in runtime without any prior knowledge about it would be the utilization of Semantic Web Technologies to develop ontologies. Through utilizing these technologies, it is possible to represent information in structured graphs and efficiently integrate heterogeneous data from various information sources as well as performing complex search queries on these sources for an enhanced information management and interoperability in the Architecture Engineering and Construction (AEC) industries through the whole construction life cycle (Pauwels et al., 2017). In addition, the underlying logical foundations of the languages used by Semantic Web Technologies such as the Web Ontology Language (OWL) or the Shapes Constraint Language (SHACL) allow for logical

inferences and proofs of asserted information. Since the knowledge is linked to the corresponding information inside the ontology, it could be exchanged between different software applications without bilaterally adjusting the applications to new types of information. Instead, the new information is interpreted in runtime by utilizing a compatible reasoning engine. Furthermore, ifcOWL, the OWL representation of IFC, allows the use of Semantic Web Technologies on an IFC-based model (Pauwels & Terkaj, 2019).

An established ontology for defining time concepts is the OWL-Time ontology (Cox & Little, 2022). It supports the definition of points in time through the class `time:Instant` as well as for time intervals via the class `time:Interval`. Furthermore, multiple object properties for structuring time sequences as well as aggregating time objects in intervals are provided. Additionally, the declaration of time values is possible through various data properties. However, the OWL-Time ontology does only enable the definition of general time concepts. Consequently, domain specific time concepts, especially those used in AEC, are missing. An alternative solution has been developed by Milea et al. (2009) through proposing a language extension of OWL called tOWL, which provides additional language constructs for defining time. However, ontologies formalized in tOWL require a specific reasoning engine different from common OWL reasoners, which prevents its usage in existing OWL applications.

Several approaches reuse concepts of the OWL-Time ontology for AEC-related knowledge representations. For example, Mignard & Nicolle (2014) developed an ontology for merging BIM and GIS information in which the temporal concepts were based on the OWL-Time ontology. Iadanza et al. (2019) used the OWL-Time ontology to describe model phases through time and represent the building evolution.

Furthermore, Zheng et al. (2021) developed an approach for representing the construction workflow utilizing the OWL-Time ontology. Similar to the process definitions used in IFC, process classes are defined in an OWL ontology. By using the existing concepts of OWL-Time time components are added to process representations.

### Linking data with information containers

One way to integrate an ontology in a model is the use of information containers. To avoid interoperability dilemmas and support collaboration, the information containers are one of the recently developed approaches proposed for the management of heterogeneous and distributed building models. The ISO 21597 series has been developed to fulfil a requirement for multiple documents to be delivered as one information package within the construction industry. The international standard ISO 21597-1:2020 provides a framework for the creation, management, and delivery of linked documents. Over the last few years many studies were conducted to investigate the implementation of the ICDD. Hagedorn et

al. (2022) proposed the concept of the Toolchain Framework to facilitate the modeling of project-specific workflows by linking individual software tools based on a standardized process notation to enable seamless information exchange between applications that integrate an openCDE-compliant web interface. Just recently the author proposed a BIM-based solution for Infrastructure Asset Management System (AMS) for road owners. Considering the requirements of stakeholders across domains in the operational phase, the proposed approach provides asset managers with a strategy for the dynamic use of Information Containers (Hagedorn et al., 2023). Moreover, Liu et al. (2021) developed the Building Concrete Monitoring Ontology (BCOM) based on the ICDD. It allows IFC-based bridge models to be configured with properties about concrete works that can be processed by predefined queries in asset management software. An approach was developed by Hamdan et al. (2021) in which an IFC model, representing an existing bridge, was linked with ontologies that semantically represent the construction and affecting structural damage as well as other related data, e.g., photos, protocols or structural analysis models. Thereby the models and links were stored in an ICDD. Ye and König (2021) presented a framework for automated billing by combining the BIM Contract Container (BCC) with Smart Contracts using Blockchain technology and the ICDD. Furthermore, recent research by Werbrouck et al. (2022) proposed a Linked Building Data (LBD) server that can link heterogeneous linked building data in a Federated CDE by combining the Solid initiative for web decentralization with the ICDD standard. Going beyond the ICDD implementation, Al-Sadoon et al. (2022) proposed an ontology-based extension that enables allocating of multiple values for elements in the IFC files to provide dynamic building model for simulation tools using the ICDD.

## The concept of chronological models

To create models with structured temporal information - not only for models of entire buildings, but for models of each delimitable component of a structure (in logically meaningful gradation, i.e., the largest possible independent component group) - the implementation of Chronological Models (cMod) is proposed. The concept of cMod is based on the use of IFC-based models of components that are linked inside an ICDD with other data and other ontologies, such as the Building Topology Ontology (BOT) by Rasmussen et al. (2020) or the Bridge Topology Ontology (BROT) by Hamdan et al. (2020). The temporal information is connected to the IFC model via ICDD by adding the proposed Ontology for Chronological Construction Processes (OCCP), containing the chronological structure shown in Figure 1 and specific, component related time information. This information provides statements about:

A - relevant times of planning of the component (e.g., start of planning, data procurement, submission for review, changes (resulting in multiple submissions for review))

B - relevant times of review (at least the time of acceptance and rejection – the latter causes another planning cycle, adding more changes (that are tracked again with the according temporal information as described in ‘A’))

C - relevant times of construction (start of execution, completion, issuance of notice of defects, completion of defect rectification, acceptance)

D - the times of commissioning of the component (if different from completion) and decommissioning (or start and end of use, multiple entries possible (e.g., to note temporary closures))

E - the dates of the beginning and end of the warranty period

F - the times of beginning and end of the design life (multiple entries possible, e.g., due to maintenance or repair works)

G - the scheduled and actual times of inspections and the length of regular inspection intervals (the cycles must be specified)

H - the times of special events (severe weather events, accidents, etc.)

K - the times of detection of damage and the start and completion of repairs (to keep track of how long a damage existed)

L – the times of repair and maintenance works (repair work is to be treated as a new component, for which the temporal information is given in categories A to M as needed.). Thus, if necessary, temporal information of other components with reference to the repair work is updated or supplemented with corresponding new values (acceptance, new predicted service life, warranty and so on))

M – the times of beginning and end of deconstruction

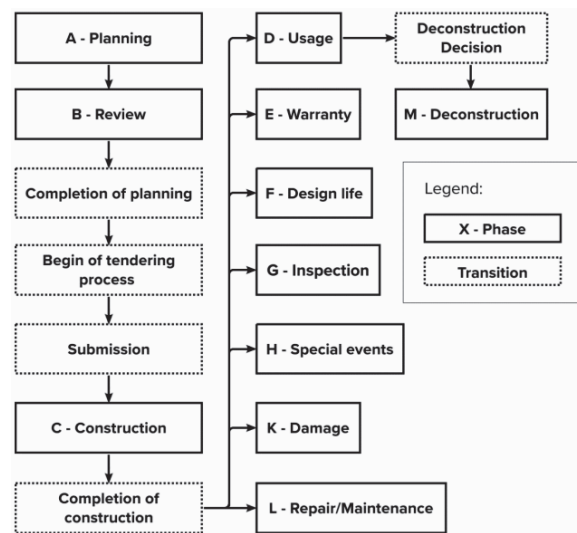


Figure 1: Temporal structure of the OCCP

The structure shown in Figure 1 is a suggestion for the sensible and practicable recording of time-specific component information. An important prerequisite for the use of temporal information is the agreement in the industry regarding the relevant temporal information. The proposed structure is based on the life cycle of a construction and divides the processes at common milestones. The process in the shown concept is kept simple and general to promote transferability and interoperability for all AEC disciplines. Further specifications, additions and smaller process steps can be developed individually according to requirements and established in exchange with experts.

A cMod differs fundamentally from the versioning of (partial) models, as it offers far more flexibility due to its component reference and makes multiple storage of individual elements in several versions of a model obsolete. The state of a structure can be mapped unambiguously by specifying the point in time under consideration.

In contrast to versioning, each component would be modeled once and supplemented with the chronological temporal information. In the case of a repair measure, the component model must be updated by separating the remaining part from the damaged part that is to be repaired. The damaged section is treated as a new cMod that goes through the phases from planning until construction and is then to be modeled as built and spatially connected to the remaining, undamaged, and unchanged rest of the component. At the end of the life cycle, the deconstruction date must be noted, but the model must not be deleted. This ensures that all information within the cMod is preserved and can be made available for planning purposes in the future.

Starting from a database in which all components including the respective information and their references to other components are stored, the Linked Data approach is used to generate the overall model of a construction from many individual modules. By specifying a certain

area, it is possible to limit the components that must be considered for the further query (e.g., building X, 1st floor). By specifying the time of viewing, the elements to be displayed for the selected time are filtered out from all elements in the viewed area by evaluating the chronological component information and are then activated for display.

## Ontological basis within ICDD for the implementation of temporal information

For the structuring and later linking of the temporal features with models or model components, the OCCP is proposed, which extends existing classes and object properties of the before mentioned OWL-Time ontology (Cox & Little, 2022). Figure 2 shows the general class structure of the OCCP. The right side of Figure 3 shows

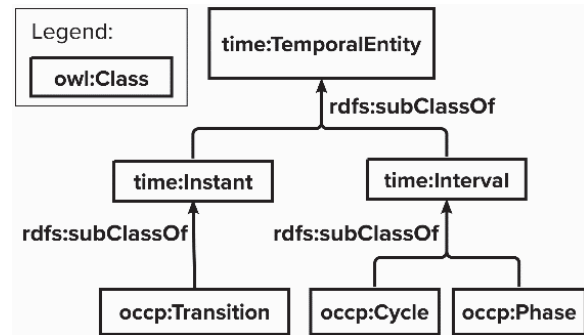


Figure 2: General class hierarchy of OCCP

the structure of the ICDD used to link the OCCP with the IFC models of a construction and its components, thus creating a cMod.

The left side of Figure 3 shows an individual of a construction (IND:ExampleConstruction) with several components linked with BOT to connect and describes the topological relation between the construction and its components. The temporal information of each of these individuals is added using the OCCP and linked to the

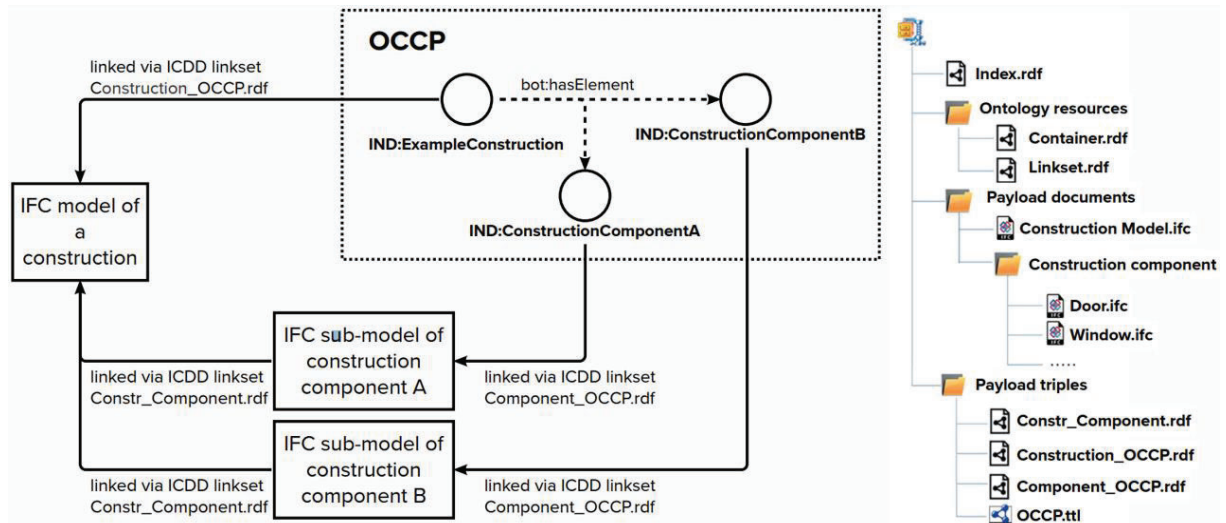


Figure 3: Integration of the OCCP with an ICDD

corresponding IFC model with an ICDD linkset. The IFC sub-models of the construction components are linked to the IFC model of the construction via ICDD linkset.

In the OCCP the classes `time:Instant` and `time:Interval` of the OWL-Time ontology are extended. Both classes are subclasses of `time:TemporalEntity`, which is a general class used for describing temporal concepts. Thereby, `time:Instant` describes a specific point in time and thus a temporal entity with zero extent or duration. Contrary to this, `time:Interval` describes a temporal entity with an extent or duration (Cox & Little, 2022). With the OCCP the temporal entity `occp:Phase` is introduced as a subclass of `time:Interval`. Instances of `occp:Phase` represent major life cycle stages of a building or construction element and are used to categorize its temporal information. Furthermore, to differ between standard intervals and cycles that are periods of time with several cascading intervals within, a new type of interval called `occp:Cycle` is introduced, which is also a subclass of `time:Interval`. This class is used to better describe iterating processes like regular inspections and their intervals. For instances of `occp:Cycle` a data property called `hasCycleNumber` can be utilized for defining the number of iterations that need to be processed by a cycle. Additionally, a new subclass of `time:Instant` called `occp:Transition` is provided in the OCCP, which is used for representing milestones between the life cycle phases of constructions.

For managing the life cycle phases and temporal entities that are associated with them, additional object properties have been added, which are shown in Figure 4.

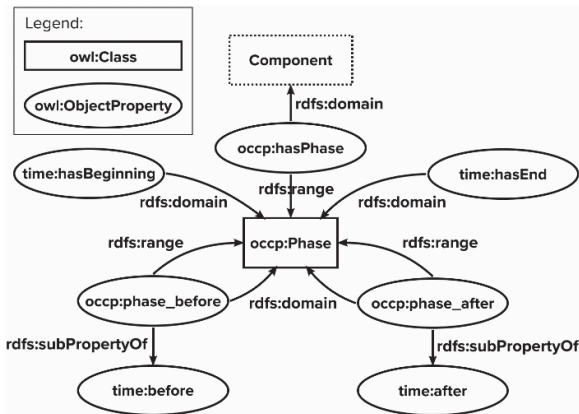


Figure 4: Object properties for managing construction life cycle phases

The chronological order of the phases is defined using `occp:phase_after` and `occp:phase_before`, which are subproperties of `time:after` and `time:before`. Each phase has a beginning and an end, both of which are defined by referencing a corresponding time instant using the OWL-Time ontology's object properties `time:hasBeginning` and `time:hasEnd`. Additional time instants within the proposed ontology mark typical key points within phases. An important part of the OCCP is the semantification of AEC-related time data. For this reason, various domain-specific subclasses of `occp:Phase` and `time:Instant` have

been developed, based on the concept of chronological models discussed in the previous chapter. In this regard, subclasses of `time:Instant` that belong to a certain life cycle phase, are categorized in corresponding superclasses, such as `occp:PhaseA_Instants` or `occp:PhaseB_Instants` and so on. Additionally, subclasses of `occp:Transition` provide more specific meanings to milestones in between life cycle phases. Furthermore, additional semantic information of a construction or component can be added to a model (e.g., topological information via BOT) and thereby extend the possibilities of information retrieval through queries.

In their current states both the cMod and the OCCP are concepts and we envisage to work out test cases in future research work.

### Example of ontology application

Figure 5 demonstrates the semantic structure of temporal information within a cMod using the OCCP, but for demonstration purposes, not all existing connections are shown. A construction component is represented by the individual `IND:ConstructionComponent` and it is linked via `occp:hasPhase` to two individuals of phases. While `IND:PhA_Planning` is an instance of `occp:PhaseA_Planning`, `IND:PhB_Review` is an instance of `occp:PhaseB_Review`, both of which are subclasses of `occp:Phase`, which in turn is a subclass of the `time:Interval`. Linked to the component are several individuals, all of them marked with an "A" that either belong to the planning and the review phase. The links are established using the object property `time:hasTime` with the according individual, e.g., `IND:PhA_Start` and `IND:PhA_Measurement`. Block 2 in Figure 5 shows the semantic connection of the A-individuals. They all have the object property of the according class within the OCCP, e.g., `IND:PhA_Start` is an instance of `occp:BeginningOfPlanning`, and `IND:PhA_Measurement` is an instance of `occp:DataProcurement` and so on. The rest of the semantic chain is the same for all A-individuals, as they are classified as `occp:PhaseA_Instant` or `occp:PhaseB_Instant` (depending on the phase they belong to), which is a subclass of `time:Instant`. The chronological structure is established by defining the succession of time instances and phases using object properties, such as `time:after` or `time:hasBeginning`. The beginning of the planning phase is marked by the start of planning, in the OCCP this connection is made by using `time:hasBeginning` and referencing `IND:PhA_Start`. Analog, the end of planning is defined by linking `IND:PhA_Planning` using `time:hasEnd` and referencing `IND:PhA_SubmissionToReview`. The order of instances within a phase is established with the object property `time:after`, e.g., `IND:PhA_Measurement` is defined as `time:after` `IND:PhA_Start` and so on. The same principle is used to describe `IND:PhB_Review` (and all following phases - not shown for simplicities sake) in order to link and structure the according individuals of the according instances. The planning phase ends with the submission of the component's plan. The case of rejection of a plan

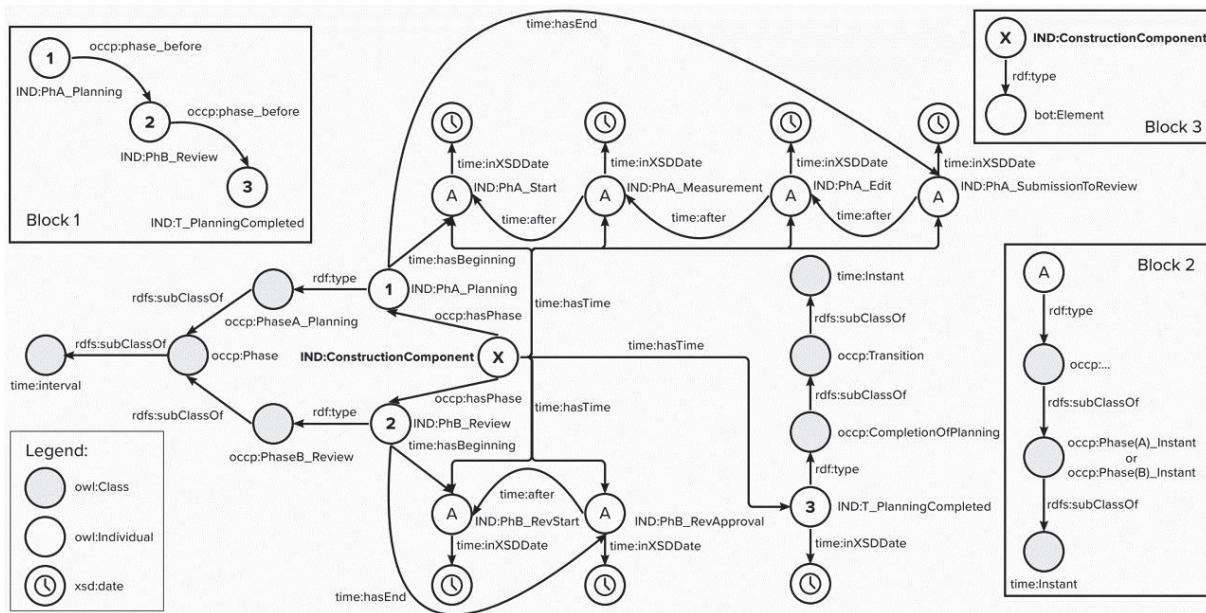


Figure 5: Example of a partial cMod of a construction component showing Phase A and B until completion of planning

and the resulting iteration can also be described using the OCCP, but is not shown in this example to keep Figure 5 readable. The review phase ends with the approval of the plan and IND:ConstructionComponent is linked to the transition instant IND:T\_PlanningCompleted, which has the `rdf:type` of `occp:CompletionOfPlanning`. All individuals in this example, aside from the component's and the phases' individuals, receive an individual temporal information using the data property `time:inXSDDate`. Block 1 in Figure 5 shows the temporal order of the planning and review phase and the transition marking the completion of planning.

Using this semantic structure for temporal information provided by the concept of cMod for each component of a construction enables a variety of options. The planning progress is tracked for each component individually and delays are potentially more obvious (because they are queryable), including the identification of the component causing the delay. This and all other temporal information can be stored within the OCCP, linked to the IFC-model via ICDD and therefore accessible for future planners, working with the model as a basis for the new planning project. All data used as a planning basis, like the measurement of the land or geological data, is linked to the construction model and the time of procurement is stored in the OCCP of the construction model. That way the actuality of all linked data can easily be checked and in case of a certain demanded actuality, the latest point in time to acquire a new set of data can be identified and used to remind the involved personnel in time. Also, the exact point in time and order of events can be of interest to answer questions of legal responsibility and it often comes down to the question, who had which information and when. The temporal aspect can be answered using a cMod.

## Potentials and applications of chronological models

The concept of cMod that is proposed in this paper could be implemented in software applications and enhanced with further reasoning, querying and validation functionalities in the future. Provided that this chronological component and model information has been implemented, it is possible for planners to obtain an overview of the old construction development in an area relevant to planning, e.g., to view old construction conditions or to identify construction remnants that may possibly collide with the planning object. By specifying the area of a model to be displayed and a point in time, which could be the current state or any state in the chronological model's past, provided there is temporal information in the point in time of interest, the combination of partial models that existed at the specified point in time can be displayed via a query of the chronological module information. By concatenating the chronological information and querying time periods (e.g., from a certain point in time until today), building states could be displayed in arbitrary increments (1 week, 1 month, etc.) and thus extensive information regarding the building states, building sequence and any backlogs could be visually displayed in one cMod. By introducing time as a linear dimension within cMods, the space related freedom from contradiction or collision is eliminated and replaced by the freedom from contradiction in time. The result is that - looking at a fixed part of a model over a certain time period - several components (e.g., an old and a new window) can be at the same position within a building model but cannot be at the same position at the same time. This enables the continuous use of existing models and thereby reduces the time and effort needed to

collect all necessary information for planning the next construction, as it will already be integrated in the OCCP within the cMod. This also implies that a versioning of models is no longer necessary, because through the difference in their temporal information, both the old and the new component can exist within the same model and in the same place. For the actual versioning process, existing approaches could be considered such as the Ontology for Property Management (OPM) by Rasmussen et al. (2018), which could enhance the querying time compared to timestamps.

Further potential applications arise from the operation of buildings and constructions, as administrations have a quick overview of warranty periods that can be used. An automated query of the remaining warranty periods can generate reminders with sufficient time to still be able to perform an investigation before the warranty expires.

Another area of application is the evaluation of types of construction, materials or construction elements regarding their suitability for the respective application by comparing the design service life to the actual service life or by monitoring the needs of related companies for repair measures. The effectiveness of repair measures can also be evaluated by following up the service life.

The chronological model could be validated against national standards, by utilizing SHACL. For instance, certain nation-specific process sequences are defined in corresponding standards and must be executed in a specific order for which SHACL shapes could warrant the correctness of their implementation in the model. Furthermore, the implementation of additional rules that could be defined in rule languages like Jena Rules, SHACL or even OWL itself, could allow for a more automatized logic-based application of the cMod. Thus, implicit knowledge, subsequent phases or required measurements or processes could be inferred through reasoning an ontology using the OCCP. Since the current proposed concept just defines the base taxonomy based on existing expert knowledge, the need for future features needs to be identified, e.g., through developing competency questions or use cases via expert interviews. Based on the identified application needs, additional rules and queries could be developed and implemented in OCCP-compatible software applications.

## Outlook and conclusion

To make the advantages of working with cMods available for all stakeholders involved in the lifecycle of a construction, the concept of cMod and OCCP would first have to be implemented in the software used by these stakeholders, covering all aspects of time relevant information across all phases. After the implementation, the access, usage, and application of the time ontology related features and information should be intuitive and as easy as the spatial assignment of model information. Temporal collisions and logical contradictions must be either prevented or indicated by the software. An integrated query function should use the temporal information of cMods and make it easily accessible, e.g.,

by presenting a selection of information depending on the temporal information given by the query. Input data for the queries must always be a point in time or a time span and a specific location (three-dimensional delimitation of the space or specific components) to be considered.

In case of a fixed spatial viewing area, a timeline function could be implemented in the software, whose extension maximum results from the earliest and last entry of temporal information. Analogous to the playtime display in music and video players, a corresponding slider could allow the control of the considered point in time and thus visualize all states of the considered object over the entire period. The corresponding model information is also displayed or retrievable depending on the considered point in time.

A potential loss of temporal information could occur when working with the IFC model of a cMod without using OCCP. To address this problem, one solution could be the use of an API to implement an export function for temporal information.

Further research should be done to

- investigate possibilities to securely store and manage temporal information to guarantee manipulation security and to answer legal questions respectively avoiding legal problems using the temporal information (e.g., by using automatically generated time stamps that are permanently unchangeable or by integrating the block chain technology).
- ensure that temporal information is free of contradictions (e.g., plausibility checks could be established by utilizing description logic in OWL).
- develop a rights management for the entry of temporal information (e.g., by assigning processor rights in combination with the 4-eyes principle (or more), whereby the authorized persons must be determined project-specifically and by mutual agreement (e.g., one representative for each client, contractor, and an independent expert / construction supervisor / BIM manager)).
- establish practical and meaningful rules for the storage of model information to create historical models from the past and for the future (e.g., by working out an agreement for indefinitely storing a defined minimum of model information).
- create an API to import and export temporal information stored within cMods to improve interoperability.

The use of chronological models, created by the permanent integration of semantically structured temporal information in models of constructions and their components using ICDD and the OCCP, reduces the need of versioning due to the possibility of spatial coexistence of multiple components that is made possible by the temporal differentiation. The continuous use of existing cMods can prevent the loss of information and has the potential to save time for planners, as the time needed for data collection for new planning projects is minimized. With the adaption of OCCP, all temporal information can

be semantically stored and used for queries, project managing purposes, and to help solve legal issues.

The authors share the concepts of the cMod and the OCCP at an early stage of development to discuss it with the AEC community and to receive feedback before making the first version of the OCCP and a minimal example of a cMod public.

## References

- Al-Sadoon, N., Katranuschkov, P. & Scherer, R. J. (2022) Extending ICDD Implementation to a Dynamic Multimodel Framework. In: Proceedings of the Conference ECPMM (doi.org/10.1201/9781003354222-15).
- Al-Sadoon, N., Scherer, R. J. & Menzel, K. (2023) From Static to Dynamic Information Containers. In: Proceedings of the Conference EC3 2023 (in press).
- Artus, M. (2021) Modeling Physical Damage Information at Concrete Bridges Considering Time Aspects. In: 32. Forum Bauinformatik 2021. Darmstadt, Germany.
- Cox, S. & Little, C. (2022) Time Ontology in OWL. <https://www.w3.org/TR/owl-time/>. Accessed date: 25<sup>th</sup> April 2023.
- Eftekharirad, R., Nik-Bakht, M. & Hammad, A. (2018) Extending IFC for Fire Emergency Real-Time Management Using Sensors and Occupant Information. In: 35th International Symposium on Automation and Robotics in Construction (ISARC) 2018. Berlin, Germany.
- Hagedorn, P., Block, M., Zentgraf, S., Sigalov, K. & König, M. (2022) Toolchains for Interoperable BIM Workflows in a Web-Based Integration Platform. Applied Sciences. 12(12):5959.
- Hagedorn, P., Liu, L., König, M., Hajdin, R., Blumenfeld, T., Stöckner, M., Billmaier, M., Grossauer, K. & Gavin, K., (2023) BIM-Enabled Infrastructure Asset Management Using Information Containers and Semantic Web. Journal of Computing in Civil Engineering, 37(1), p.04022041.
- Hamdan, A.H., Taraben, J., Helmrich, M., Mansperger, T., Morgenthal, G. & Scherer, R. J. (2021) A semantic modeling approach for the automated detection and interpretation of structural damage. Automation in Construction, 128, p.103739.
- Hamdan, A. H. & Scherer, R. J. (2020) Integration of BIM-related bridge information in an ontological knowledgebase. In: Proceedings of the 8<sup>th</sup> Linked Data in Architecture and Construction Workshop (LDAC) Iadanza, E., Maietti, F., Ziri, A.E., Di Giulio, R., Medici, M., Ferrari, F., Bonsma, P. & Turillazzi, B. (2019) Semantic web technologies meet BIM for accessing and understanding cultural heritage. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W9 (2019), pp. 381-388
- Liu, L., Hagedorn, P. & König, M. (2021) An ontology integrating as-built information for infrastructure asset management using BIM and semantic web. In: Proceedings of the 2021 International Conference on Computing in Civil and Building Engineering, 99-106.
- Mignard, C. & Nicolle, C. (2014) Merging BIM and GIS using ontologies application to urban facility management in ACTIVE3D. In: Computers in Industry 65 (2014), pp. 1276-1290.
- Milea, M., Frasincar, F. & Kaymak, U. (2009) tOWL: Integrating Time in OWL. In: Semantic Web Information Management, pp. 225-246.
- Pauwels, P. & Terkaj, W. (2019) ifcOWL ontology (IFC4x1). In: <https://standards.buildingsmart.org/IFC/DEV/IFC4/>, Accessed date: 25<sup>th</sup> January 2023.
- Pauwels, P., Zhang, S. & Lee, Y.C. (2017) Semantic web technologies in AEC industry: a literature overview. In: Automation in Construction 73 (2017), pp. 145-165
- Rasmussen, M. H., Lefrançois, M., Schneider, G. & Pauwels, P. (2020) BOT: the Building Topology Ontology of the W3C Linked Building Data Group. In: Semantic Web. DOI: 10.3233/SW-200385.
- Rasmussen, M. H., Lefrançois, M., Bonduel, M., Hviid, C.A. & Karlshøj, J. (2018) OPM: An ontology for describing properties that evolve over time. In: 6<sup>th</sup> Linked Data in Architect and Construction Workshop. London, United Kingdom.
- Tanaka, F., Tsuchida, M., Onosato, M., Date, H., Kanai, S., Hada, Y., Nakao, M., Kobayashi, H., Hasegawa, E., Sugawara, T. & Oyama, T. (2018) Bridge Information Modeling based on IFC for supporting maintenance management of existing bridges. In: 17<sup>th</sup> International Conference on Computing in Civil and Building Engineering (ICCCBE) 2018. Tampere, Finland.
- Werbrouck, J., Pauwels, P., Beetz, J. & Mannens, E., (2022) Lbdserver-a federated ecosystem for heterogeneous linked building data. Semantic Web Journal (submitting).
- Ye, X., & König, M., (2021) Framework for automated billing in the construction industry using BIM and smart contracts. In Proceedings of the 18th International Conference on Computing in Civil and Building Engineering: ICCBE 2020 (pp. 824-838). Springer International Publishing.
- Zheng, Y., Törmä, S. & Seppänen, O. (2021) A shared ontology suite for digital construction workflow. In: Automation in Construction 132 (2021), pp. 103930