A system architecture ensuring consistency among distributed, heterogeneous information models for civil infrastructure projects

S. Esser & A. Borrmann

Technical University of Munich, Germany

Proc. Of the 13th European Conference on Product and Process Modeling. Moscow, 2020/2021

ABSTRACT: The application of suitable data structures is an essential aspect for novel digital workflows in engineering and design processes of the Architectural, Engineering, and Construction (AEC) industry. Since model-based data exchange gets increasingly adopted by the industry, feasible and more effective methods must be considered to improve data exchange in the future. While the concept of federated model integration and container-based collaboration as demanded by ISO19650 is well established and widely adopted, it shows a number of deficiencies, in particular when it comes to consistency preservation across the domain models and the handling of design updates. Current practice relies on the exchange of complete domain models which requires the manual identification of design changes by all other stakeholders. Consistency is checked merely by collision detection, which however can cover only geometric aspects. To overcome these limitations, this paper proposes a comprehensive system architecture as well as techniques to identify updates in models and federate such update information by means of update patches. To this end, specific focus is put on possible mechanisms to detect changes and to integrate update patches in the receiving application.

1 INTRODUCTION

The building industry is heading towards more and more digitized processes. Especially for civil infrastructure assets, a very large number of engineers, clients, contractors, and authorities are involved during several planning stages. Each of these disciplines has specific requirements for capturing, storing, and presenting relevant information, ranging from pure semantic information over schematic system design to 3D geometry representations. In this collaboration, all experts contribute to the common goal of a coherent and error-free design of the built asset. Hence, combining heterogenous knowledge representations is a significant challenge that has not been entirely solved so far.

It is well established in today’s BIM practice, that engineers work in individual environments and upload their domain models to a common project data platform (also known as “Common Data Environment (CDE)”), thus implementing the concept of federated data models as described by UK BIM Level 2 (NBS, 2020) and ISO19650 (CEN, 2019). Such an approach enables each engineer to work with applications that suit his needs the best. The actual integration is realized by means of model coordination which is dominantly based on collision detection of 3D geometry. Many other logical dependencies between objects in the domain models remain untracked, especially as an object might have different representations in different domain models. Thus, the concept of collaborative but separated work leads to severe limitation in consistency preservation.

Linked data approaches can help to manage relationships between these various representations (Beetz & Borrmann, 2018; Zhang & Beetz, 2016). However, it is often useful or even obligatory for modeling and design tasks to fully integrate foreign models into the engineer’s design environment. The design of security equipment for railway systems is a vivid example: the security equipment engineers who design necessary signals and sensors along a new railway track must rely upon a given alignment axis and a linear reference systems (LRS). The de-sign of the track alignment, however, is typically a task of another subdomain in railway engineering, considering vehicle dynamics, existing facilities, terrain, and land ownership. Thus, consistency preservation mechanisms between the alignment designer and the security engineer are a crucial factor for a successful and faultless design of the complete rail-way system. To achieve this high level of data coherence among all disciplines, all involved parties must be notified on updates in foreign models which their domain models are based upon.

According to BIM Level 2, project participants propagate such model changes by uploading the complete updated domain model to the CDE again. This approach suits the basic needs of information provision but does not unlock the potential of truly integrated digital workflows. The main deficit lies in the fact that each participant must manually incorporate the model changes into their own domain models. This becomes particularly prone to errors, as such models contain an extensive number of components, different types of geometry, and complex relationships. Hence, incorporating update information in a domain-specific design environment.

Looking more in-depth to the specifics of civil infrastructure projects, linear reference systems (LRS) is an essential denominator among all civil infra-structure domains. Since these projects typically ex-tend over several kilometers, also data integration over spatially distributed submodels must be considered.

The currently available IT solutions for CDE are mostly based on file management. According to ISO19650, the concept of “containers” allows the bundled management of inter-related files. File-based approaches, however, have severe limitations with respect to consistency preservation as the smallest accessible information unit is the file. Individual updates on object level can thus neither be identified by the CDE, nor be propagated to other domain models. On the other extreme, the concept of a central database fully resolving and managing all objects of all domains has been discarded as unpractical, mostly due to reasons of ownership and liability.

To overcome the discussed limitations, we pro-pose a distributed system architecture that respects the principle of discipline-oriented collaboration with loosely coupled, federated domain models and heterogenous information models, but provides the means for fine-grained, object-level integration and consistency preservation by a patch-based update deployment mechanism. Heading towards an increasing number of federated models, this can significantly reduce the exchanged data and allows automated consistency preservation by integrating up-dates. Thus, instead of the full model, only the change operations must be re-evaluated. Nevertheless, a patch-based approach requires the correct interpretation of a patch, which must be ensured on the receiver’s side (e.g., providing integration rules).

Furthermore, the proposed patch-based mechanism should have a generic form that can be used independently from any specific data exchange format. Contrary to building design processes where three-dimensional geometry is the leading common denominator among domains, significantly differing information representations are in use for civil infrastructure design. These representations range from detailed 3D geometries and simple semantic information up to simple geometry but complex and heavily coupled semantic data.

Our paper summarizes existing approaches for shared collaboration in distributed systems and presents a conceptual approach for distributed CDE systems. The proposed architecture represents a possible approach towards implementing BIM Level 3 principles, realizing a deep integration of domain models. A specific focus is put on the requirements of civil infrastructure projects which have particular requirements regarding spatial placement and varying representations of individual semantic objects.

2 OVERALL SYSTEM ARCHITECTURE

The paper at hand presents a possible approach for a BIM Level 3 collaboration architecture. Several stakeholders coming from various domains are working collaboratively together. Each modeler is responsible for the domain-specific models he is creating. Such model is created in authoring applications that suit the domain requirements best.

The proposed CDE is a distributed, loosely coupled system basically following the peer-to-peer architecture (Chen & Hou, 2014). Each authoring application forms an individual peer in the system. To this end, existing software systems must be extended by connector modules that provide means for communicating within the distributed system. This module must hold a representation of the domain model, must be capable to identify changes performed by the specialist in order to propagate them into the P2P network. At the same time, it must be able to receive notifications from other domain nodes. In deviation from pure P2P architectures, the CDE has a central node that holds all shared domain models and pro-vides an information bus. A possible domain scenario for civil infrastructure projects is illustrated in Figure 1.

The integrity of the distributed, heterogenous model is preserved by a patch-based update mechanism to federate design changes. Such patches can be utilized to ensure consistency between different representations among different domains. Practically spoken, this approach enables each domain to represent the domain-specific knowledge in a data model that suits its specific needs. However, if updates are applied to a domain model, all other domains are in-formed. This allows the domains which are impacted by the update to integrate the respective changes in their own model. This integration can be performed automatically by defined rules, or with manual interventions and under active control of the respective specialist.

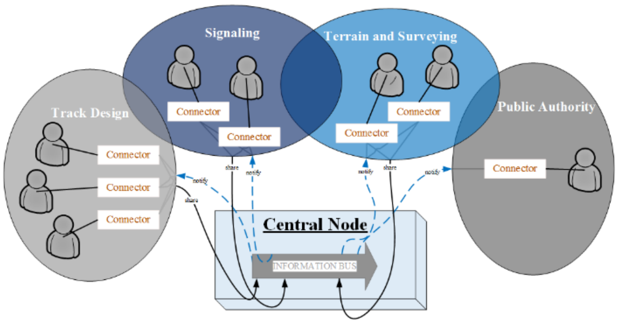

Figure 1: Collaboration System with one Central Node and distributed

A major component of the proposed architecture is the information bus. The concept is well-known from distributed application architectures and enables clients to communicate via centralized communication medium.

In the proposed architecture, each domain node can connect or disconnect to the information bus. The connection between a domain application and the information bus is managed by a connector module. This module contains methods to track local changes and enables the user to share the updates once the domain model reaches a sufficiently mature state for sharing. Both communication approaches, request and reply as well as publish and subscribe are supported (Oki, Fluegl, Siegel, & Skeen, 1993). Furthermore, domain nodes can be introduced as either master or slave nodes. Master clients can send updates to the information bus at any time whereas slave nodes can only reply to update messages that are relevant for its individual domain.

A basic assumption of the proposed architecture is the availability of heterogenous, domain-specific data models. Common anchor points define denominators that help to achieve consistency between various data models. In the context of civil infrastructure projects, such anchor points are defined using the following concepts:

Project breakdown structure: This concept splits a built asset in multiple containers. Components can be assigned to these containers to ease the navigability of the overall model. Examples are the separation of a building into building storeys or the use of several containers that separate elements along a railway track.

Reference systems: As explained in the introductory section, (linear) referencing systems are of essential importance in the design processes of linear stretched infrastructure. LRS can build a reliable concept among several data representations (Esser & Borrmann, 2019).

Classification systems: Established classification systems like Omni- and UniClass can also help the reach consistency among various data sets. Chang, Lee, & Cho (2009) have investigated research in commonly used classification systems in railway systems that assists in maintenance and operation issues.

3 LITERATURE REVIEW AND RELATED CONCEPTS

The management of complex interdependencies has been subject to numerous investigations in the context of BIM.

The UK BIM task group has defined different levels of BIM implementation (Bew & Richards, 2008): BIM level 0 describes conventional collaboration approaches to design and collaborate within a project. Data exchange among involved parties is realized by data structures that are either proprietary or their interpretation not automatable. The predominant communication medium is large printed 2D plans. BIM level 1 improves the situation by introducing file-based exchange scenarios. Such data are still not fully automatable for receivers since no or only weak requirements formalize data handover scenarios.

Nevertheless, BIM Level 1 marks the milestone from purely paper-based to file-based information exchange. Additionally, 2D geometry is often extended using the third spatial dimension. Moving on, BIM level 2 adds increasing structure to exchanged design data. Such formalization is based on object-oriented approaches to separate information in logical units. Besides, using any kind of database system also improves the quality of information exchange. All these initiatives result in model-based data ex-change, which further enhances the formalization and automatization of sending and receiving data sets. Comparing BIM level 2 and 3, the latter focuses on increasing data exchange consuming web technologies.

In comparison to level 2, exchanged information must be still structured and formulated according to an agreed data model (Counsell, 2012; Eadie, Browne, Odeyinka, McKeown, & McNiff, 2015).

The novel concept in a BIM level 3 implementation is the object-based communication including the identification and federation of updated information instead of re-sending complete instance models. Research publications have already shown that collaboration based on domain-specific models should be preferred over working in a single but centralized model. Furthermore, this principle leads to the situation that each involved party can work in their preferred design environments and only have to ensure suitable data exchange interactions with other engaged experts.

Windisch, Katranuschkov, & Scherer (2012) have defined a framework to filter domain models for further usage in foreign domain environments. However, their approach does not manage updates on the filtered models, which is one of the critical aspects of a BIM level 3 environment. Adoption of such filtering techniques for BIM models might help to identify relevant updates.

Semenov & Jones (2015) have pointed out that none of the existing BIM CDEs provide functionalities for the continuous integration of committed up-dates.

Transmitting only model updates rather than full files were already subject to earlier investigations. Researching a more generic field of (weak) coupling of distributed data systems, Crooks et al. (2016) have developed an approach called TARDiS, which abstracts update information as simple as possible and at the same time still interpretable. Their concept ensures context reasoning as well as respecting the core principles of distributed systems like ALPS and the CAP theorem.

In addition to TARDis, an older but still valid approach on branched versions among several versions of XML-based representations was introduced by Vagena, Moro, & Tsotras (2004). Their paper pro-poses a compromise of a so-called log approach and a snapshot approach. The log approach starts with an initial state of the XML document and only stores the update. In comparison, the snapshot approach always stores the full document which consumes more storage but provides faster results when querying for a specific intermediate version.

Dawood, Siddle, & Dawood (2019) are facing the problem of continuous model updates using natural language processing and updating the data model accordingly.

In the context of combining various design options, Mattern & König (2018) have proposed a graph-based data system and a possible extension to an existing data model. Their proposal intends to harmonize content from various design options authored by different domains. Since their proposal is limited to a single data model (namely IFC), the issue of deploying update information among different data models remains unsolved.

Moving from various design options to refined building designs among different project stages, Abualdenien & Borrmann (2019) have proposed a meta-model to face the challenges of emerging levels of developments in building models. This approach is an essential contribution to the presented situation in civil infrastructure projects but cannot take the variety of domain-specific data models into account that are currently in use for civil infrastructure design tasks.

In conclusion, interacting with change requests in complex building information models or other types of structured data is not an entirely new discipline. However, none of the referenced publications is fully capable of ensuring consistency between different data models that represent correlated built assets.

4 CONCEPTUAL APPROACH

The following sections provide further explanations about specific exchange steps and functionalities to reach consistency among distributed data. Upfront, a short overview of computational principles summarizes basic concepts, which are preliminary assumptions. Subsequently, these technical principals are evaluated in the context of shared data environments for railway and civil infrastructure projects.

4.1 *Concurrency principles*

It is assumed that each domain has a well-defined data model that suits its specific needs. The domain data models are supposed to follow the principles of object-oriented programming. Furthermore, it is supposed that the schema definitions of all data models are known and available (but not necessarily vendor-neutral). Therefore, also applications with a proprietary backend data model can be considered once these tools provide the implementer an accessible application programming interface (API).

In projects, design tasks require local copies of foreign domain models. Thus, updates made to such locally stored replicas can lead to inconsistencies and contradictions. The concept of concurrency control can help to resolve such contradictory model stages distributed over several clients and domains. In general, concurrency can be performed in different approaches. With the pessimistic concurrency control conflicts are avoided in advance and only certain changes are allowed. In the optimistic concurrency control, conflicts in the project information are identified and resolved eventually. Transferring these approaches to programming paradigms of database systems, the pessimistic concurrency control is aligned with the ACID principles whereas the optimistic concurrency control can be interpreted in the context of BASE principles.

4.1.1 *ACID principles*

ACID principles define baseline to interact with a database system. The abbreviation ACID represents:

Atomicity: A database interaction consists of several steps which are performed one after the other. The principle of atomicity states that all actions must be performed correctly during one transaction peri-od. If any problems occur during the transaction process, all executed actions must be rolled back to reach the initial and valid state inside the database system.

Consistency: Each interaction with the database must result in a valid state. This principle can be interpreted in two different approaches: an adequate representation in a technical manner or an accurate description in terms of the engineer’s knowledge. This set of fundamentals can be formulated in rules that validate an incoming transaction request.

Isolation: An update on the database is decoupled from any other action that is performed on the database system. Intermediate results cannot be accessed. This principle ensures repeatable queries that result in the same response.

Durability: successfully committed transactions are not reverted in case of any external effects such as power loss.

In summary, no access is granted to the database as long as any other running transaction is executed. The policy ensures consistent data at any query time (since queries are a transaction in itself) but limits the system to a few amounts of operations per timeframe.

4.1.2 *BASE principles*

Contrary to the ACID principles, the BASE approach does not guarantee permanent and up-to-date consistency of the database. The abbreviation BASE stands for:

Basic Availability: the stored data is not blocked during running transactions. Thus, datasets are available most of the time

Soft state: store operations do not have to be consistent during the writing operation and do not directly federate to all existing replicas of the data set.

Eventual Consistency: Consistency among replicas is achieved eventually at a later point in time. (Robinson, Webber, & Eifrem, 2015)

Hence, users can interact more frequently with the system. The advantage of this approach is better scalability for large data sets and improved performance. However, complex inconsistencies an contradictions (like described in the optimistic concurrency control approach) might occur during synchronization if distributed replicas have changed in themselves since the last synchronization. BASE principles are applied in all modern web systems where multiple users communicate with servers in parallel (e.g., webshops, mailing systems, video platforms, chats).

4.2 *Adoption and conclusions*

In the context of collaborative platforms for building and civil infrastructure projects, a mixture out of both approaches, optimistic and pessimistic replication, will serve the users' requirements best. In general, pessimistic replication guarantees a valid centralized dataset since federated information is directly incorporated. However, such push operations to the database might happen irregularly due to various reasons. The frequency of commits might depend on domain-specific design tasks and relevant local iterations or the individual behavior of the engineer.

An approach of optimistic replication can lead to unsolvable clashes during delayed synchronizations but make data better available using local replicas. Thus, federated update information must be instantly checked when a new change request is sent to the centralized data system.

As a result of the presented principles, the next section introduces a system architecture that advances existing CDE technologies by extending them by a bus network architecture.

5 IMPLEMENTATION APPROACH

Based on the explained definitions of consistency, the proposed system should extend existing approaches known from BIM Level 2 implementations. The management of models as the smallest unit must be overcome by a finer level of object-based communication. Information stored in a single asset is consistent in itself and represents the result of updates, which were applied to a model. Federated assets update distributed local replicas by extending existing CDE platforms by a bus network architecture. We propose a hybrid system that provides all project members with constant information and update flow. This integration can be performed automatically by defined rules, or with manual interventions and under active control of the respective specialist.

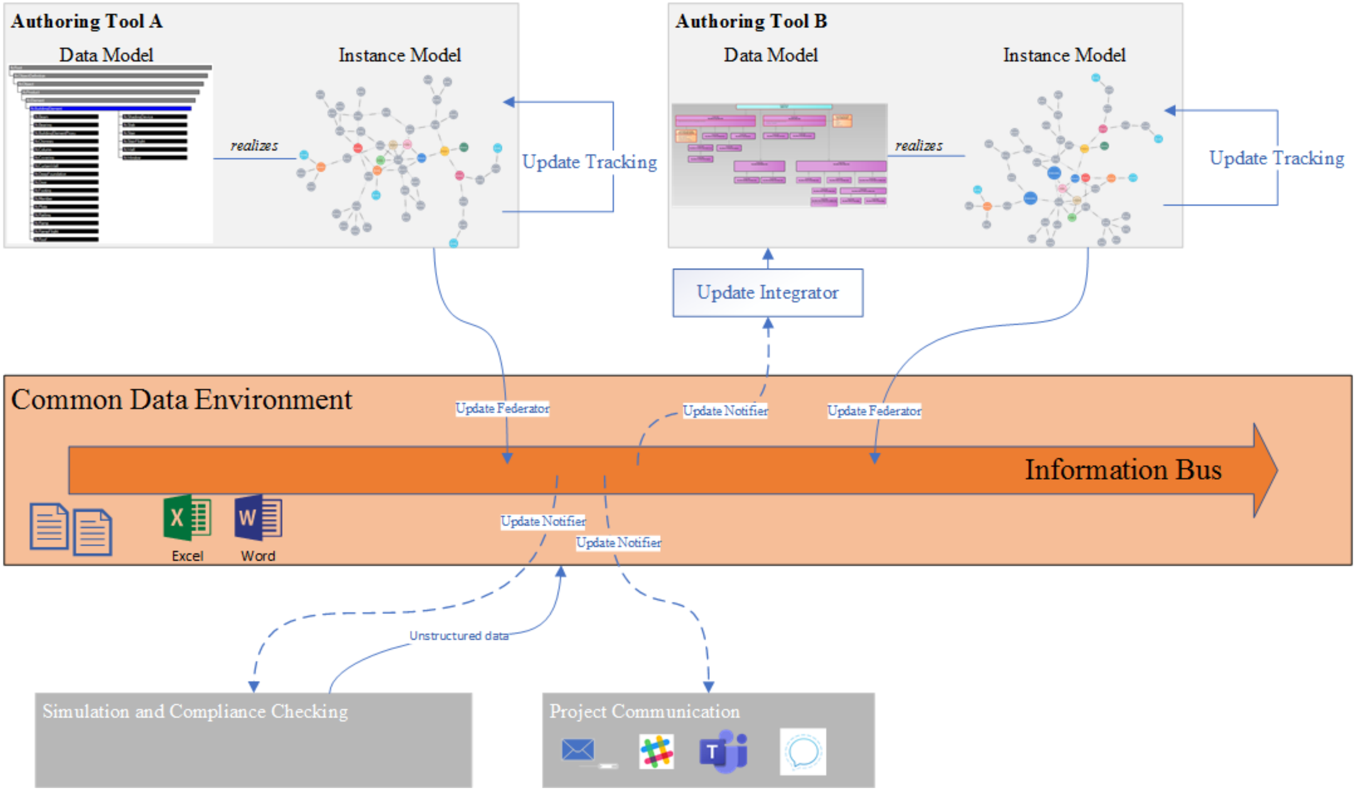

Figure 2: Technical Realization of the proposed BIM Level 3 System Architecture

5.1 *System architecture*

The complete system can be broken down into smaller units of client-server interactions. The following sections explain required operations in detail. Figure 2 visualizes the overall architecture of the proposed system.

The central part of the collaboration system comprises of a project server that contains the information bus. It is initialized once a new project is set up. Domain applications identify updates on their domain models and share the applied changes to the information bus. Various types of notifiers are used to classify update information. Domain nodes can subscribe to specific update messages and can decide whether the provided update information presented on the bus must be incorporated in their own replicas. Besides, client nodes can behave in various manners (e.g., only as notification bots in communication applications or for decoupled downstream use cases).

Thus, client nodes can either be equipped with a federator and a notifier method or only provide one of these interfaces. Authoring tools subscribe to specific subjects and get notified if events of these sub-jects happen.

The development of our system architecture focuses on the exchange of updates applied to domain models. However, additional types of data (documents, raw binary data, etc.) will exist in parallel to the actual models. Hence, the common data environment that is based on the information bus, also provides further methods to store and exchange ac-companying documents that represent results out of various design stages. Such documents include but not limited to guidelines, codes, tendering documents, or billing sheets.

The following steps build the technical base to identify, deploy, and integrate model changes between distributed replicas of domain models.

5.2 *Update identification*

To identify update information that has to be dispatched among the project, it is essential to track (a) relevant assets and changes inside such assets and (b) relationships between related assets.

An asset can be defined as a generic data resource that represents a logical piece of model in-formation (e.g., products, spatial structure containers, components of (linear) reference systems). Both assets and relationships carry additional attributes. The set of assets and relationships results in a graph representation that is aligned with but not necessarily a 1:1 replica of the tracked domain data model. Various domain data models distinguish their classes in the root layer and additional resource layers. Classes that inherit from root entities and thus have a unique identifier are suitable candidates to be represented as an individual asset node.

Two approaches for identifying updates between an initial and an updated model can be defined:

5.2.1 *Comparison of an initial an updated model state using graph representations*

Graph representations of instance models extend simple text comparisons by the extended use of relationship information between assets. Subgraph iso-morphism can help to detect changes between an initial and an updated graph representation. (Ullmann, 1976). Hidders (2001) has worked on a generic approach utilizing object-oriented data models with graph representations.

Simple graph comparison between an initial and an updated model might not supply the complete procedure of modification steps that are necessary to re-construct the modification steps in a receiving platform.

5.2.2 *Update tracking during the design process*

Contrary to the approach mentioned above, performed updates can be directly tracked inside the modeling or simulation applications. This approach requires additional plugins in each authoring and modification tool that should either identify or integrate update patches. Koch & Firmenich (2011) have defined a generic description for models and applied model changes.

Further research has to be conducted to verify the applicability of both approaches. Although the latter approach contains the potential for live collaborative work sessions, it might be beneficial to include an explicit share operation that initializes an update federation process to the information bus. Such a gate is crucial since design and modeling tasks in the civil infrastructure projects sometimes require iterative calculations during the design phase. Thus, not every iteration loop should be propagated instantly to all other project participants. Besides, the information aggregation per asset node requires further investigation to ensure compatibility with a broad range of domain data models that are currently used in projects.

5.3 *Update federation*

Besides the identification of model changes, the concept for information federation is crucial for the proposed system.

The information bus, which was introduced in section 2, federates the update information among project members. Besides the federation, the central node should also hold the latest instance of all domain models. The provision of all recent models is already implemented in various CDEs currently available on the software market. If a new actor joins the project, he can directly pull the latest model versions using existing approaches related to BIM Level 2 systems.

Additionally, each federated update patch can be tested against individual criteria to ensure the integration feasibility in receiving replicas. Besides, the provision of all recent models inside the central node enables further services within the platform (e.g., visualization, clash detection or issue management) (Oki et al., 1993).

Once a local replica on the client system is stored, the interpretation of incoming patches (modifying the fetched model state) can start. Providing a particular model state on a centralized platform is a technological approach, which is already known from BIM Level 2 systems. However, providing the latest state of the available instance model and not only the complete history of federated patches helps in several regards: Incoming patches can be instantly tested and verified before deploying them to clients. Furthermore, new project members can easily fetch the latest model to start their collaboration interactions without re-interpreting a long list of already performed updates.

Since not every single update information might be relevant for each domain or project stakeholder, participants should get the opportunity to decide on specific types of updates that have an impact on their design and modeling tasks.

5.4 *Information integration*

Due to the enormous number of components that have complex relationships within the considered models, a fully-automated update integration cannot be realized. In addition, civil infrastructure projects often comprise of models that do not only carry do-main-specific content but also have a specific spatial context (e.g., a detailed bridge model which should be integrated into a railway track model for visualization and clash detection purposes). Furthermore, different projection and coordinate systems used for georeferencing have to be taken into account as well (Jaud, Donaubauer & Borrmann, 2019).

Hence, scripting interfaces help to interpret and automate incoming update patches in receiving applications. Recurring events that don't cause any contradictions in the receiving model are automatable. Besides, additional scripts evaluate the relevance of incoming updates for specific design tasks.

6 SUMMARY

The proposed system architecture extends existing BIM Level 2 approaches towards BIM Level 3 features. The smallest unit of data exchange is not a model but a set of updated objects, which reduces data traffic and makes updates easier to interpret in receiving applications. To improve information federation, a bus architecture enhances data availability and serves as the key concept in an extended CDE platform. Each domain can subscribe to update events that are relevant for its own design tasks.

The separation of update identification, federation, and integration enables an agile approach that supports existing applications and enables users to stick with their familiar tools. Besides, established data models remain untouched and are integrated in the communication procedure.

7 OUTLOOK

Due to the legacy of fragmented and cross-disciplinary collaboration, it is essential and challenging at the same time to maintain existing and established approaches as good as possible and formulate extensions based on this heritage. However, tools from giant software vendors like Microsoft, Google and Apple already demonstrate how shared collaboration within complex data works.

REFERENCES

Abualdenien, J., & Borrmann, A. 2019. A meta-model approach for formal specification and consistent management of multi-LOD building models. Advanced Engineering Informatics, 40, 135–153. https://doi.org/10.1016/j.aei.2019.04.003

Beetz, J., & Borrmann, A. 2018. Benefits and limitations of linked data approaches for road modeling and data exchange. Workshop of the European Group for Intelligent Computing in Engineering, 245–261.

Bew, M., & Richards, M. 2008. BIM maturity model. Construct IT Autumn 2008 Members' Meeting. Brighton, UK.

CEN. DIN EN ISO 19650-2:2019. 2019.

Chang, T., Lee, S., & Cho, M. S. (2009). A Study of the Information Classification for Railway Industry. IJR International Journal of Railway, 2(1), 37–42.

Chen, H. M., & Hou, C. C. 2014. Asynchronous online collaboration in BIM generation using hybrid client-server and P2P network. Automation in Construction, 45, 72–85. https://doi.org/10.1016/j.autcon.2014.05.007

Counsell, J. 2012. Beyond level 2 BIM, web portals and collaboration tools. Proceedings of the International Conference on Information Visualisation, 510–515. https://doi.org/10.1109/IV.2012.88

Crooks, N., Pu, Y., Estrada, N., Gupta, T., Alvisi, L., & Clement, A. 2016. TARDiS: A branch-and-merge approach to weak consistency. Proceedings of the ACM SIGMOD International Conference on

Management of Data, 26-June-20, 1615–1628. https://doi.org/10.1145/2882903.2882951

Dawood, H., Siddle, J., & Dawood, N. 2019. Integrating IFC and NLP for automating change request validations. Journal of Information Technology in Construction, 24, 540–552. https://doi.org/10.36680/j.itcon.2019.030

Eadie, R., Browne, M., Odeyinka, H., McKeown, C., & McNiff, S. 2015. A survey of current status of and perceived changes required for BIM adoption in the UK. Built Environment Project and Asset Management, 5(1), 4–21. https://doi.org/10.1108/BEPAM-07-2013-0023

Esser, S., & Borrmann, A. 2019. Integrating Railway Subdomain-Specific Data Standards into a common IFC-based Data Model. 26th International Workshop on Intelligent Computing in Engineering. Leuven, Belgium.

Hidders, J. 2001. A Graph-based Update Language for Object-Oriented Data Models (University Press Facilities, Eindhoven, the Netherlands). https://doi.org/10.6100/IR551259

Jaud, Š., Donaubauer, A. & Borrmann, A. 2019. Georeferencing within IFC : A Novel Approach for Infrastructure Objects. In Y. K. Cho, F. Leite, A. Behzadan, & C. Wang (Eds.), Computing in Civil Engineering 2019: Visualisation, Information Modelling, and Simulation. Atlanta: American Society of Civil Engineers.

Koch, C., & Firmenich, B. 2011. An approach to distributed building modeling on the basis of versions and changes. Advanced Engineering Informatics, 25(2), 297–310. https://doi.org/10.1016/j.aei.2010.12.001

Mattern, H., & König, M. 2018. BIM-based modeling and management of design options at early planning phases. Advanced Engineering Informatics, 38(July), 316–329. https://doi.org/10.1016/j.aei.2018.08.007

NBS. 2020. BIM Levels explained - Definitions for levels of BIM maturity from Level 0, through Level 1, Level 2 and Level 3 and beyond. Retrieved March 28, 2020, from https://www.thenbs.com/knowledge/bim-levels-explained

Oki, B., Fluegl, M., Siegel, A., & Skeen, D. 1993. Information bus an architecture for extensible distributed systems. Operating Systems Review (ACM), 27(5), 58–68. https://doi.org/10.1145/173668.168624

Robinson, I., Webber, J., & Eifrem, E. 2015. Graph Databases. In M. Beaugureau (Ed.), Joe Celko’s Complete Guide to NoSQL. https://doi.org/10.1016/b978-0-12-407192-6.00003-0

Semenov, V., & Jones, S. 2015. IFChub : Multimodal Collaboration Transactional Guarantees Platform with Solid. Proc. of the 32nd CIB W78 Conference 2015, 27th-29th October 2015, Eindhoven, The Netherlands, 667–675.

Ullmann, J. R. 1976. An Algorithm for Subgraph Isomorphism. Journal of the ACM (JACM), 23(1), 31–42. https://doi.org/10.1145/321921.321925

Vagena, Z., Moro, M. M., & Tsotras, V. J. 2004. Supporting branched versions on XML documents. Proceedings of the IEEE International Workshop on Research Issues in Data Engineering, 14, 137–144. https://doi.org/10.1109/RIDE.2004.1281713

Windisch, R., Katranuschkov, P., & Scherer, R. J. 2012. A generic filter framework for consistent generation of BIM-based model views. European Group for Intelligent Computing in Engineering, EG-ICE 2012 - International Workshop: Intelligent Computing in Engineering, 1–12. Munich.

Zhang, C., & Beetz, J. 2016. Querying Linked Building Data Using SPARQL with Functional Extensions. Proceedings of 11th European Conference on Product and Process Modelling, (September).