TOWARDS METHODOLOGY FOR HARMONIZATION OF SEMANTICALLY DIFFERENT BIM's

Andrej Tibaut, Danijel Rebolj
University of Maribor, Faculty of Civil Engineering, Slovenia
andrej.tibaut@uni-mb.si

ABSTRACT

Research focus of the paper are heterogeneous information systems. Heterogeneity within a set of software applications can be attributed to the fact that their collaboration is hindered due to the conflicts in software architecture, communication protocols and/or data representation. General interconnectivity and emerging interoperability have caused the fall of mainframe-based systems, which in turn led to variety of information systems with local data representations, communication protocols and software architectures. Today these information systems need to collaborate in different engineering projects. Existing approaches, such as common framework, integration with standard scheme and data mediation, try to diminish the undesired effects within heterogeneous systems. The approaches are indeed successful because they eliminate all conflicts at design time. This way collaborating applications have to abandon their local data views. In this paper heterogeneity is regarded as a property of an information system while disharmony of an information system is defined as a state of the system. Further, structural, semantical and functional disharmony is defined as part of overall information systems's disharmony. As a consequence a new methodology called DRAGOn (Disharmony Resolving with Agents and Ontology) is proposed. The methodology aims to dynamically resolve structural and semantical disharmony by preserving applications' local data views. Another novelty is the definition of conceptualization for structural and semantical disharmony (Disharmony ontology) and the use of software agents. Disharmony ontology is specified in OWL. The agents use the ontology for resolving of structural and semantical conflicts between applications at runtime. Agents communicate via shared communication space based on Java technology. The mediation is incremental, which means that agents are able to build their local ontologies. The ontologies are used as persistent meta-data repositories of concepts (structure and semantics) that are captured from applications during runtime.

Extensive applicability of the DRAGOn methodology is expected in information system clusters with rich and complex data content, namely management of construction projects.

KEYWORDS

Interoperability, building information model, quality of semantic and structure, semantic and structural difference, mediation, ontology

1. COMPLEXITY OF BUILDING INFORMATION MODELS

Today's best information modelling practices in engineering show a tendency to build a unique allinclusive product information model for a specific engineering field (like shipbuilding, car industry, building industry, road building, etc.). However, none of these attempts has been generally accepted in the civil engineering practice. Rather than that, the past development of building information models (building product model) led to a question, whether a definition and use of a standard product model has sense at all. To overcome the need to have a single product model some authors have proposed inter-model linking schemes (like in [1], and [2]), in this way, however, the complexity of the whole system hasn't decreased, even worse, it has grown.

Another problem arises from the necessity for standard building elements. The history of mankind shows that in communication the only "standard" is the diversity of standards. In other words, it seems most unlikely that the whole of mankind would use a single standard language. Even if such a language would exist, it is very likely that soon many dialects would appear, since every individual or group is seeing the same thing in its own perspective.

This problem is also present in civil engineering and construction, where many different views have to be considered through a building life cycle. Different views are leading to more or less different descriptions (data structures) representing the same entity. Notable progress has been made by the International Alliance for Interoperability [3] with the development of the common specification for information model (IAI IFC 2x3) rooted in the Industrial Foundation Classes (IFC), which can be seen as applicable building blocks tailored to cover the software interoperability within the AEC industry. IAI has made it to become an ISO “Publicly Available Specification”. [4]

STEP technology ([5]) has succeeded as an ISO 10303 standard. It was supposed to be much broader in scope than IFC and with the ambition to seamlessly integrate processes in manufacturing engineering. The ISO 10303 standard bundle contains over 100 published technical specifications. But will they prevail over IFC or vice versa?

A conflict between the concept of a single project integrated model and the need for individuality also showed up. Companies (and individuals) have a strong inclination to fully control their own data, which also form the company’s “memory”, a vital part of every company.

The main deficiencies of product information models could be summed up into the following essential points [6]:

- Product models are based on clearly defined semantics and demand unique standard basic elements, however, such elements don’t exist,
- Computers are not (yet) capable to fill up semantic inconsistencies and gaps, which show up in the integration of computer programs - a human is adapting daily to communication patterns with other humans with different mental models, and is capable to reconceptualize parts of information, which don’t fit into the whole,
- Product models are subjective interpretations, not objective representations of the real, therefore an effective uniform product definition is not possible,
- Product models only include parts from the building process and disregard some important views (social, environmental, etc.), which form the process in reality,
- Models are restricting creativity due to their complexity and rigidity,
- When implementing prototype models into the real environment they fail due to the inability to capture the rich knowledge and experience of the people,
- Although product models are basically open, they get stiff and hardly upgradeable in reality,
- In an integrated database each client’s control over his own data is limited.

Years ago construction IT scientists ([7] and [8]) already proposed some solutions to the problems they described:

- Product models should be rather small and limited to specific areas; coexistence of more models in the same field is not necessarily bad,
- Implementation of middleware tools between applications and models, which will help humans to navigate between the islands of automation,
- Gradual implementation of small models into industry,
- Development of a richer set of language constructs for model description,
- Product and process models should be linked more closely,
- new integration concepts should be developed, which would not reside on integrated semantics,
- It is necessary to allow coexistence of structured information and unstructured data and leave their interpretation to the human,
- Pure information exchange should be upgraded with communication software for collaboration support.

Furthermore, authors [9] conclude that in the last 20 years computer integrated construction research produced software solutions focused on data and application integration that in consequence lead to tightly coupled information systems and complex building information models. The latter, complexity of handling a shared building information model, is also confirmed through research case studies [10].

We can conclude that the applicability of "giant" information models is limited because of their complexity and lack of adaptability. As such they require too many agreements and change from diverse users. In the future we won't see any prevailing international standard for product modelling but number of best-practice examples commonly used at different levels (company-wide, country-wide, maybe even continent-wide) that will require collaboration with each other.

In such a world only those information model concepts that feature some degree of self-initiative and adaptability will be widely accepted. Any other concept risks the high migration cost to be paid by its users.

2. FORMALIZATION OF DISHARMONY IN PROPRIETARY BIMS

Our research tries to diminish disharmony effects within heterogeneous federated software systems (HFSS) where set of functionally complementary applications that contribute to a global goal wouldn't need to abandon their local data views. A natural assumption is that applications are autonomous in modelling and managing of structure and semantics of their data.

We define heterogeneity as an intrinsic property of an information system. Disharmony of an information system is defined as a state of the system that changes over time and we are interested in conflicts that arise from that. Further, structural, semantical and functional disharmony is defined as part of overall information system's disharmony.
Two different disharmonies can be defined:

Static disharmony is a conflict state within a set of collaborative software systems "before its first run"; at least two applications within HFSS cause heterogeneity

Dynamic disharmony is a conflict state within a set of collaborative software systems "during the run"; a new application that cause heterogeneity is added to an existing HFSS

A good example of disharmony problem demonstrates a HFSS with discipline-specific applications that are used to design, construct, and operate buildings by capturing information about all aspects of a building throughout its lifecycle. To facilitate full interoperability between the applications state-of-the-art scenario would be to use standard schema (i.e. IFC) with import/export mechanism between the internal data models of specific building modelling applications, such as ArchiCAD and Autodesk Revit for example, and server based building information model. Therefore the standard schema must define building entities that represent union of all the entities (i.e. window, wall, column, etc.) found in internal data models. Such a schema doesn't exist. Therefore disharmony problems can not be avoided in real construction projects.

2.1 STRUCTURAL DISHARMONY

Structural disharmony occurs when one application requests data from another application.

Structural disharmony is a conflict state between two applications in a HFSS because of differences in data format structure and data names

Structural disharmony can have different forms:
- Different structure for the same concept
- Equal structure for the same concept but with different names
- Equal name for different concepts

2.2 SEMANTICAL DISHARMONY

Semantical disharmony (example: Picture 1) occurs when one application receives data from another application, but interprets its meaning wrong or fails to interpret it at all.

Examples for semantical disharmony are:

Semantical disharmony is a conflict state between two applications in a HFSS because of differences in meaning of data (semantics) with equal or related structure that represent the same concept

- Semantically equivalent data with different precision
- Semantically equivalent data represented with different units

Resolving disharmony problems between building information models means resolving structural and semantical disharmony.

Structure and semantics can be further defined as *implicit* or *explicit.*

Structure or schema for data is implicit if its description is not separated from data.
Structure or schema for data is explicit if its description is separated from data.

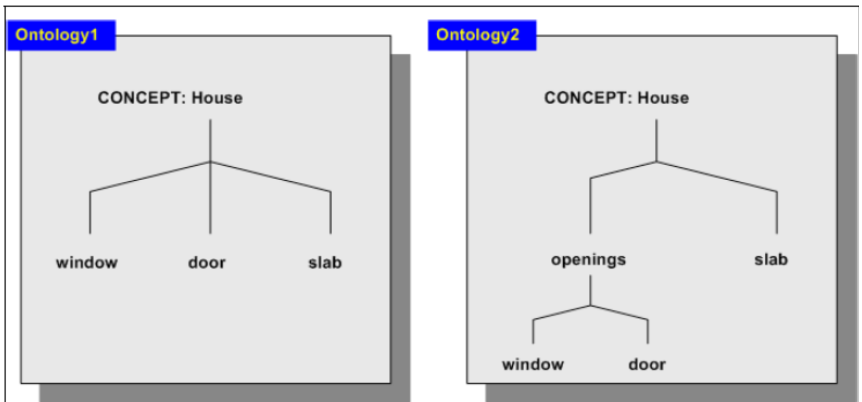
BIMs with implicit structure are exclusively managed by a software program released by a given organization. With the proprietary approach, the security is higher and reverse-engineering a proprietary algorithm is difficult, but this is compromising software collaboration within construction project. Explicit structure or schema (data in a file follows a hierarchy, grouping, example is XML with XSD) approach is well suited for machine interpretation.

Equivalently we can define *implicit* or *explicit semantic*.

Meaning or semantics of data is implicit if its description is not separated from data.
Meaning or semantics of data is explicit if its description is separated from data.

Existence of explicit structure and semantics for a BIM enables the BIM to be readable and understandable by software applications. However this is yet unreachable goal for HFSS in construction practice.

A methodology that we propose for resolving structural and semantical disharmony in HFSS applies only to BIMs with explicit structure and semantics respectively.



Picture 1: Semantical disharmony example

2.3 CLASSIFICATION OF DATA MODELS IN RELATION TO STRUCTURE AND SEMANTIC

To formalize the notion of structural and semantical disharmony we need to identify a quality of description of structure and semantic for a given building information model. For the purpose we introduce a metrics that denotes the quality of structure and semantics, *QSS* (*Quality of Structure and Semantics*), for a given building information model.

$$QSS = \left(Str_{|i,e|}, Sem_{|i,e|}\right)$$

$$Str_i = \text{implicit structure}$$

$$Str_e = \text{explicit structure}$$

$$Sem_i = \text{implicit semantics}$$

$$Sem_e = \text{explicit semantics}$$

The value of *QSS* describes a HFSS's structure and semantics as four discrete states; binary pairs (Table 1. QSS values).

Table 1. QSS values

(Str_i, Sem_i)	$(0,0)$	(implicit structure, implicit semantics)
(Str_e, Sem_i)	$(1,0)$	(explicit structure, implicit semantics)
(Str_i, Sem_e)	$(0,1)$	(implicit structure, explicit semantics)
(Str_e, Sem_e)	$(1,1)$	(explicit structure, explicit semantics)

The *QSS* has a real value if it can be determined for every building information model. How can we possibly explain that a picture has *QSS* = (0,0)? If not, than there should be an explicit schema that describes the order of pixels that compose the image, and explicit semantics description could be used by a pattern recognition algorithm that would realize, for example, that a construction site activities are behind the project schedule.

Another example is a VRML geometry model. A VRML model contains 3D geometry model together with animation path. 3D geometry is described with VRML entities like points, shapes, materials, interpolators, appearances, animation sensors, etc. The VRML model can be visualized with viewers, like
CORTONA VRML client, that use proprietary (hard wired) algorithms for processing (implicit) structure and semantic. The VRML model has *QSS* = (0,0).
The *QSS* value will be used as a trigger that determines how to approach disharmony resolving.

3. DRAGON – A METHODOLOGY FOR RESOLVING DISHARMONY BETWEEN PRODUCT MODELS

Based on the above formalization of disharmony in HFSS we propose a new methodology that dynamically resolves structural and semantical disharmony by preserving applications' local data model management. This is possible by understanding conceptualization (ontology) for structural and semantical disharmony. As a consequence a new methodology called DRAGOn (*Disharmony Resolving with Agents and Ontology*) uses conceptualization for structural and semantical disharmony, a disharmony ontology, and software agents technology. Disharmony ontology can be specified as an OWL ontology. Software agents can use the ontology to resolve structural and semantical conflicts between applications at runtime.

Software agents communicate via shared communication space, a middleware that enables mediation of semantically different BIMs. The mediation incrementally builds local ontologies. The ontologies are used as persistent repositories of concepts (structure and semantics) that are captured from applications during run-time.
Extensive applicability of the DRAGOn methodology is expected in construction industry with rich, complex and dispersed data content. The general goal of DRAGOn is to help in discovery decrease semantic and structural difference between product models. The importance of ontological harmonization of enterprise product models is also well described in [12].

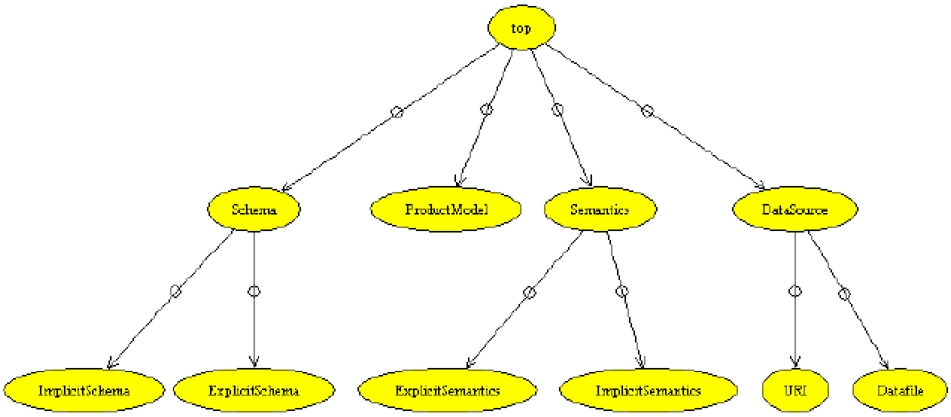
3.1 DISHARMONY ONTOLOGY

We can now formally define a conceptualization for structural and semantical disharmony of data in HFSS. We will call it *disharmony ontology*. An ontology is specification of conceptualization. An ontology is a document that describes vocabulary for communication between (humans and) software agents.

Identification of concepts for the ontology is based on the knowledge (facts and rules) about data specification languages, like XML, ISO/STEP, IAI IFC and UML. Top level hierarchy concepts are: **concept Schema** (a meta description of a data structure), **concept Explicit Schema**, **concept Implicit schema**, **concept Schema about schema** (reference to schema components from a schema document, trend is towards using combinations of schemas rather than just one; XML technology enables use of multiple inheritance with schemas), **concept Schema language** (standard STEP uses EXPRESS schema language, while XML representation of schemas is XML in IFC), **concept Semantics**, **concept implicit semantics, concept Explicit semantics, concept Product Model**, **concept Building Information Model**, **concept Product model language** (language for product model description and schema language description can be the same), **concept Software tool for schema validation** (schema validators check syntax and structure conformance of schema with schema language rules), **concept Reference to schema** (product model can have an explicit reference to a schema which can be used for product model validation or product model can be validated against a software encoded schema), **concept Tools for product model manipulation** (a complete product model can be beneficial for different business cases that go beyond usual geometry exchange and visualization and involves also non-geometry applications like structural analysis, acoustic design, facility management, CNC machines; there are known cases of robotised laying of new asphalt layer), **concept Pool of supporting tools** (there are number of tools for conversion between different schemas, example is conversion of IFC SPF to IFC XML and vice versa, **concept Egocentric technology** (a product modelling technology apparatus has a rather high gravity force for number of domains), **concept Technology implementation complexity** (a domain specific product modeling application authors are aware of the steep learning curve towards standard product modelling technology).

The conceptualization (hierarchy of concepts) is an input for definition of **disharmony ontology** in DAML-OIL, an ontology specification language. DAML-OIL is oriented towards web and is an upgrade to RDF and RDFS. RDF semantically upgrades XML. RDF/RDFS typically uses triples property, resource and value.

Part of the hierarchy of disharmony ontology is shown on Picture 2. A complete disharmony ontology written in DAML-OIL is located at http://www.tibaut.net/ontology/dragon.daml.



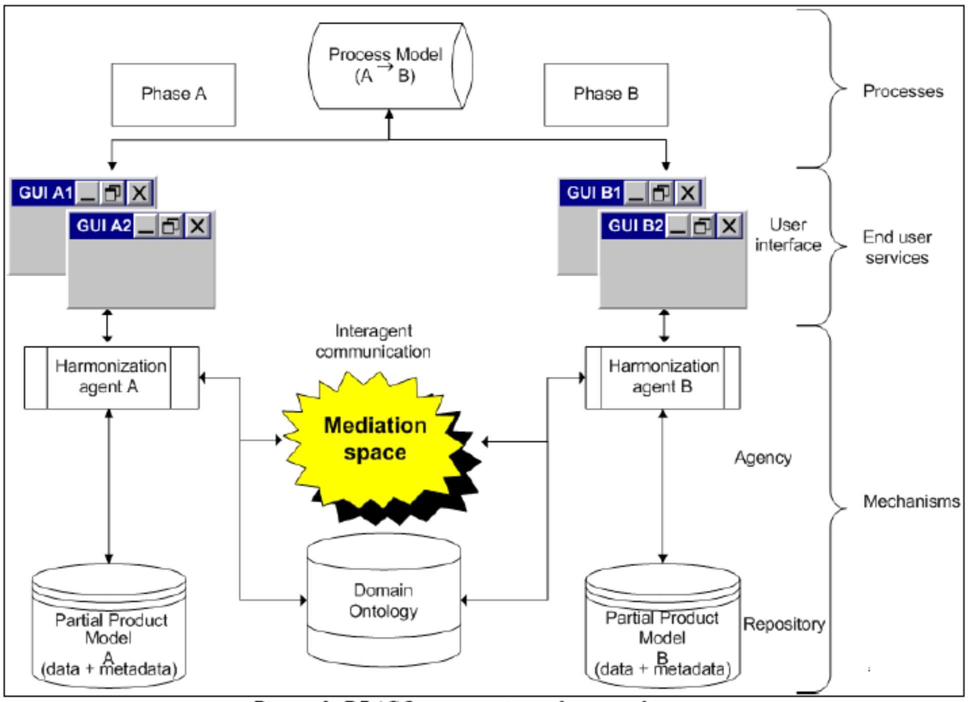
Picture 2: Disharmony ontology (partly): Hierarchy of concepts

3.2 DRAGOn ARCHITECTURE

In contrast to the traditional schema integration approach, more recent thinking is to “mediate” dissimilar data representations instead of “integrating” them. This mediation approach is typically done by using some mediation rules or specifications, which are used to resolve various kinds of conflicts among component systems at runtime. A mediated information system allows the users to see data in their own views. They can issue queries based on their own views and receive data in representations that are familiar to them. Furthermore, the mediation approach provides better support for system extensibility and scalability because, unlike schema integration approach, adding new component systems to the heterogeneous system can be done by changing or adding mediation rules or specifications instead of redesigning or modifying the integrated schema.

The virtual product model approach can be explained as a decomposed product model, consisting of the three main levels (Picture 3):

- Mechanisms including repository with data structures used by applications and multiagent system which is responsible for structural and semantical harmonization of data in repository,
- End user services that include graphical user interfaces, and
- A process model, which determines the collaboration flow for applications (the “higher sense” of applications).



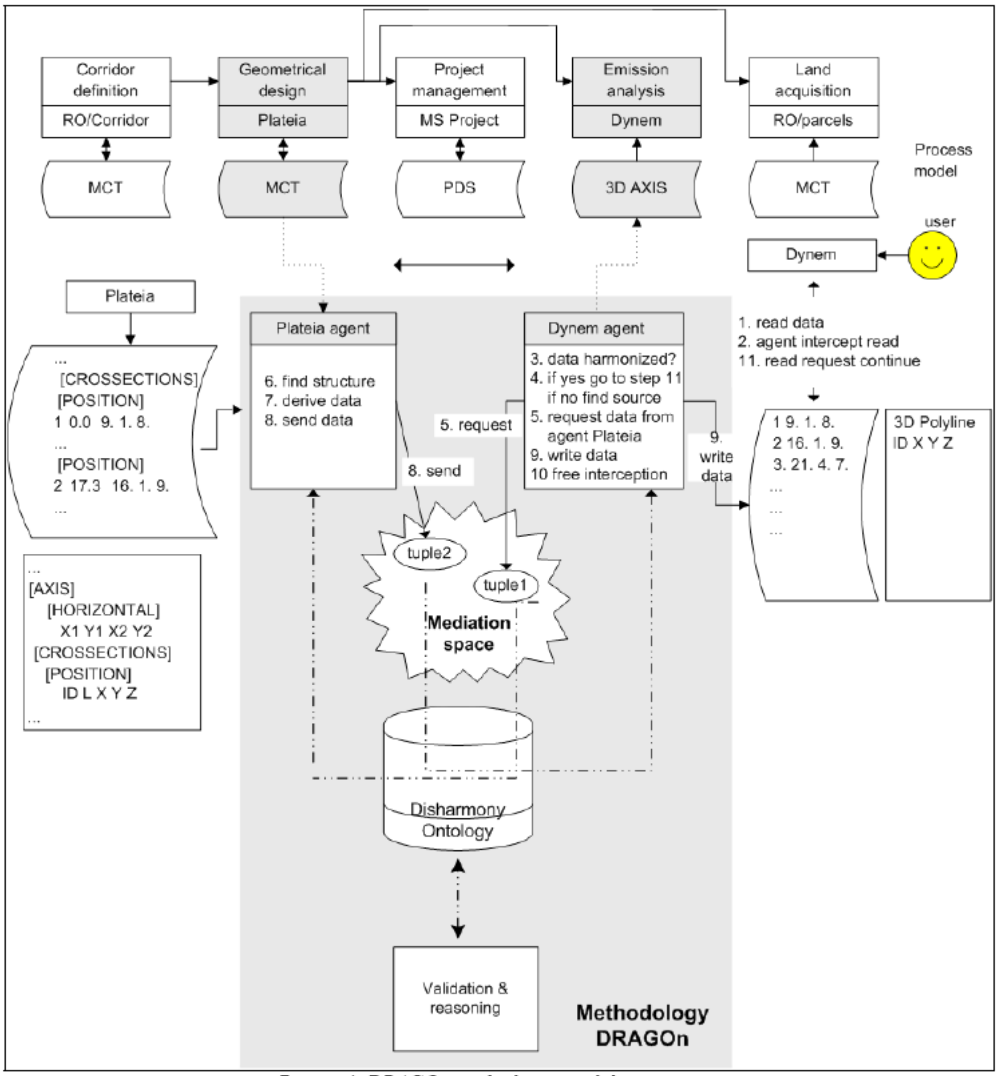
Picture 3: DRAGOn: supporting software architecture

Our integration model can also be viewed as a multilevel integration model where the mechanisms implement the end user services that support the process. The process model encodes the set of goals and constraints of the context in which the end user services will be used to meet some objective of an organization.
It is believed that such federation of heterogeneous and decomposed data sources (product models A and B in Picture 3) will decrease complexity of the product model to a manageable level and increase it's flexibility through the autonomy of partial product models - components.
Some of the important features of our concept, motivated by the shortcomings of existing technology, include:

- The ability to share data across multiple heterogeneous data sources,
- The ability to manipulate the meta-data (schema) component of a data source in the same manner as data can be manipulated,
- Increased adaptability towards foreign product models.

4. EXAMPLE

Picture 3 shows use of the DRAGOn methodology in the road lifecycle (the road has been in the focus of our research group for over a decade now [11]). A simplified schema of the process model shows a chain of tasks, with the information about the program(s) and the external data representation used in a specific task. The process model is built, or adopted, for each specific project, because every project can include slightly different tasks, carried out by different programs.



Picture 4: DRAGOn applied in a road design proces

The scenario starts with the activation of the task "Emission analysis", which is supported by the program named Dynem. When the user specifies the project name he wants to work on, Dynem tries to read relevant data. The read request is intercepted in an asynchronous way by the Dynem's harmonization agent through the mediation space that's created for the environment. The agent checks the status of the data in the project process model (implemented as a project database). If the requested data is harmonized with the predecessor components, the agent stores read request as a new tuple (tuple1, a set of objects) in the mediation space. Location of the data source is not determined in the process model since it only access (reads, writes) the mediation space. The Plateia-agent, which is responsible for the geometrical design, detects the read request (tuple1) in the mediation space. When Plateia-agent gets the description of the requested data structure through the knowledge representation repository, it tries to find it in its partial product model and returns the data as a new tuple (tuple2) in the tuple space. Now the Dynem-agent detects the content he is looking for, reads the tuple (tuple2) and updates data in its partial product model. The user can then continue to work with harmonized data.

From this example it can be seen that data in the process is not harmonized all the time, but only on demand. This mechanism simplifies the harmonization of the model as a whole, but still assures correct data when it is needed.

It is believed that the DRAGOn methodology will be especially effective in civil engineering, where processes and partners, as well as applications used, are changing from project to project. However, same applications are used more often, which makes it possible for their agents to improve knowledge repositories.

5. CONCLUSIONS

In a real scenario domain specific BIM often uses different syntax and semantics than other partners' BIMs or "central" BIM. This leads to interpretation conflicts. For practical reasons (security, IPR) construction project partners want to retain their own project data view (as opposed to project data lock-in) using existing software architecture. Industry domain BIM standards, for example IAI IFC, are rapidly extending pool of supportive applications becoming. Still data loss can happen both in importing from and exporting to the standard BIM. For the standard BIM to facilitate full interoperability between applications, it would have to be a superset of all their data models, which would be hardly realistic. Our methodology DRAGOn introduces approach that tries to understand disharmony problems in HFSS and decreases the semantical and structural differences in a construction project. Disharmony ontology helps to identify disharmony conflicts in data representations and software agents could help to resolve them. Such approach enables loosely coupled project software architecture as opposed to rigorous project discipline (to split a model to building levels). As a consequence this can support interoperability across the individual, discipline-specific applications that are used to design, construct, and operate buildings. Our near future work includes proof and analysis of our methodology based on a prototype implemenatation within the european InPro project.
(http://www.inpro-project.eu/).

REFERENCES

[1] Spooner D.L. & Hardwick M. 1997. Using views for product data exchange. IEEE Computer Graphics and Applications 17(5): 58-65.
[2] Pfennigschmidt S., Kolbe P. & Pahl P.J. 1997. Integration von Datenmodellen. Proceedings of the IKM conference. CD-ROM. Weimar.
[3] IAI, International Alliance for Interoperability, http://www.iai-international.org; {accessed 20.12.2007}
[4] IAI, International standards, http://www.iai-international.org/Resources/InternationalStandards.html;{accessed 20.12.2007}
[5] STEP on a Page, http://www.mel.nist.gov/sc5/soap/soapgrf030407.gif; {accessed 20.12.2007}
[6] Tibaut A., Rebolj D. Towards a virtual product model. International journal of internet and enterprise management. [Print ed.], 2003, vol. 1, no. 2.
[7] Eastman C. & Augenbroe F. 1998. Product modeling strategies for today and the future. Proceedings of the CIB W78 conference The life-cycle of construction IT innovations. Stockholm: The Royal Institute of Technology.
[8] Turk Ž. 1999. Constraints of product modelling approach in building. Proceedings of the 8th International conference on Durability of Building Materials and Components. Vancouver: NRC Research Press.
[9] Stefan Boddy, Yacine Rezgui, Grahame Cooper, Matthew Wetherill: Computer integrated construction: A review and proposals for future direction. Advances in Engineering Software 38(10): 677-687 (2007)
[10] Jim Plume, John Mitchell: Jim Plume and John Mitchell, Collaborative design using a shared IFC building model--Learning from experience, Automation in Construction, Volume 16, Issue 1, CAAD Futures, 2005, January 2007, Pages 28-36.
[11] Rebolj D., Tibaut A., Čuš Babič N., Magdič A., Podbreznik P. Development and application of a Road Product Model, Autom. constr.. [Print ed.], 2008, str. 1-10. http://dx.doi.org/10.1016/j.autcon.2007.12.004.
[12] Framework for enhanced interoperability through ontological harmonization of enterprise models, RICARDO JARDIM-GONCALVES, JOÃO P.M.A. SILVA, ANTÓNIO A.C. MONTEIRO, ADOLFO

STEIGER-GARÇÃO, chapter of book entitled "Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems", Springer, ISBN 978-0-387-37022-4, 2007.