Theme:

Title: **More Knowledge for less Complex Product Models**

Author(s): Andrej Tibaut, Danijel Rebolj

Institution(s): University of Maribor, Faculty of Civil Engineering, Construction IT Centre

E-mail(s): andrej.tibaut@uni-mb.si

Abstract: *Emerging digital interoperability causes pressure and uncertainity within the labor-intensive industries. This is particularly true for construction industry. Therefore ITC faces great challenges. One of the most teasing challenges is the problem of transparent data exchange within product lifecycle. Data exchange based on concepts like data import/export, single standard agreement and common scheme does not perform satisfactory.*

In the paper integration through repository of concepts - ontologies within application domain is promoted. Manipulation of concepts is based on logical derivations of explicit knowledge. Integration issues that arise in a federation of heterogeneous data sources, possible storing related information, are investigated. Using our approach it is possible to compose a product model from heterogeneous data sources. We also suggest the use of a multi-agent system where agents take care for queries from multiple sources.

Keywords: *complex product model, knowledge management, ontology, mediation, agent*

Introduction

From the large international projects that promote use of IT in construction we can learn that technology push approach still dominates over process driven approach. In the first case commercial IT tools and standards are applied to cover the particular sub processes, information is modelled in a standardized way (IFC, STEP) and exchanged between the sub processes. Unfortunately, data exchange between sub processes is based on a "single standard agreement" that promotes unity in data exchange. As such communication is limited to sub processes that "speak" an agreed data modelling language. It seems that such restriction suits most of the project teams in construction. However, in distributed environments where construction project team consists of heterogeneous data sources (loosely coupled contractors) a single standard agreement is a-priori an obstacle in collaboration.

Today many researchers, working in the field of engineering information technology, recognize the problem of modelling complex structures, and many are asking themselves whether an all-inclusive-product-model is a solution for an integrated information environment that should efficiently support the life-cycle of a product. It seems that rich experiences in product modelling in the last decade lead not to better and better models but rather to the awareness that the more complex the product models are, the more rigid and the less usable they become in practice. These recognitions have already led to some suggestions for the future integration methods and product modelling.

Before analysing the deficiencies of complex product models, a short history of product modelling should be considered. This started when the first data interface had been implemented, which linked the output of one computer program to the input of another. After that, researchers started to develop more sophisticated integration methods. According to the principle we can divide them in the following groups:

- Integration of different stand-alone programs with the help of information interfaces, as for example in the “software fixing” method [29]. These methods have two main deficiencies: they don’t enable fluent information flow, and it is necessary to implement a new interface stub for every new program we want to include.
- The use of a common medium for information exchange between programs. “Blackboard” is one such method [32], which enables a fluent information exchange through a common “blackboard”. The “Object shell” method [25] supports a fluent information exchange as well, however all these methods still require implementation of new interfaces to include new programs.
- The integrated database concept, where all included programs use a common data repository. There have been many projects, which have developed and used this concept: RATAS [6], ATLAS [5], COMBINE [4], COMBI [2], and in the last years SPACE and OSCON, which set the fundamentals of the technologically advanced integrated environment in civil engineering, described in [12]. Among earlier, but less known systems, the CIS, Construction Information System [24] introduced an integrated geometry-construction database. Many authors published more detailed reviews of relevant projects and systems, including the listed ones (e.g. [3], [11]).

Nowadays the integrated database concept is recognized as the most effective method for integration of computer programs in the life cycle of a building object. The integrated database contains the complete description of a product, therefore such data models are known as product models. Although the database technology is the mature one, its wide adoption in the

field of product modelling is yet in its infancy. The reason being the accepted complexity of product models. Ambitious projects have to deal with huge complexity, while small projects are limited in that they don't bring together all of the different stakeholders involved in a construction project. The Gallicon prototype system [13] demonstrates an all-encompassing approach (Figure 1) based on OSCON. Their integrated database includes complete descriptions of objects from housing and wastewater treatment. The effort needed for the system to be fully operational in a live commercial situation would require description of about 100 objects in the database. The time frame for the development of each object depends on the complexity of the object and takes between 2-4 days. Object definition requires involvement of the information owners.

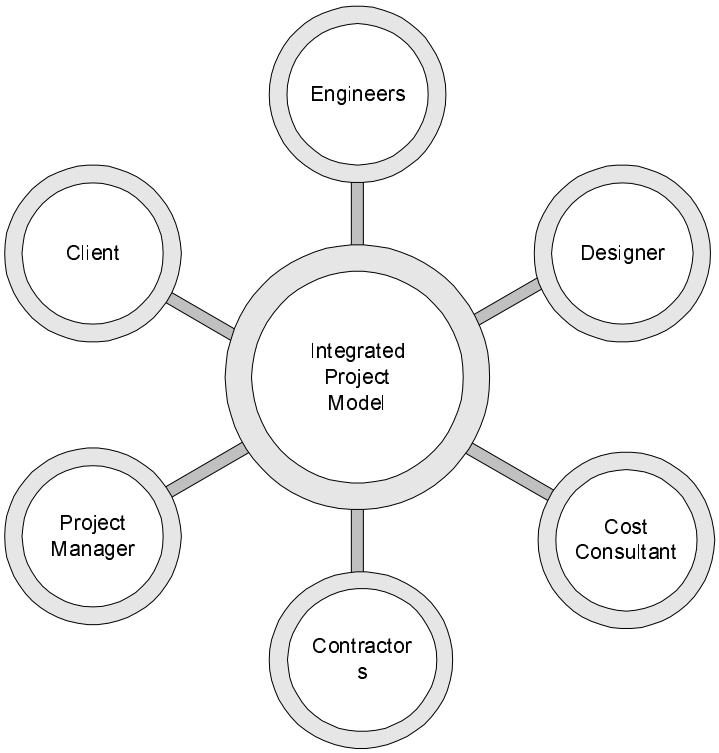

Figure 1.Integrated project model (Galicon 2001) is stored in an integrated database

The integrated database approach is a subtype of the repository-based integration, which views the issue primarly from the data sharing point of view [7]. An alternative integration approach is focused on control rather than shared data. In the control integration approach, a system is viewed as a collection of services provided by different components [28].

Applicability of complex product models

Present examples of product models show a tendency to build a unique all-inclusive complex 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 product models 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 [27], and [23]), 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 even extended in civil engineering and construction, where many different views have to be considered through a product 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 [16] with the development of the common software platform (IAI IFC 2x [17]) rooted in the Industrial Foundation Classes (IFC, [18]), which can be seen as applicable building blocks tailored to cover the software interoperability within the AEC/FM industry. However, the building blocks are still not resolving the problem of views (as evident from [33]). IAI recently made a submission to ISO for accreditation of the IAI IFC 2x Model as an ISO "Publicly Available Specification" under new "harvesting of externally developed specifications rules". STEP technology has succeded as an ISO 10303 standard [19] .Much broader in scope than IFC and with the ambition to seamlessly integrate processes in manufacturing engineering, the ISO 10303 standard bundle contains 86 standards published between 1994 and 2002. Latest additions and updates being on bindings to different programming languages (Java, C, and C++), new application protocols (electrotechnical design, automotive mechanical design, composite and metallic structural analysis) and the encoding of the exchange structure. The concept of application protocols provides standardized views for products. But will they prevail over IFC?

A conflict between the concept of a single 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" [22], a vital part of every company.

Such and similar problems have already been recognized by some authors, who have expressed their hesitation about applicability of complex product models, either between lines (e.g. [14] and [3]), or directly (as in [11] and [30]). Appending authors' own experiences, the main deficiencies of product models could be summed up into the following essential points:

- 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.

[11] and [30] also 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 tried, 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,
- programs should not limit but extend the engineers capabilities (virtual reality, telepresence, multimedia, etc.)
- pure information exchange should be upgraded with communication software for collaboration support.

We can conlude that the aplicability of "giant" product 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 product model concepts that feature some degree of self-initiative and adaptibility will be widely accepted. Any other concept risks the high migration cost to be paid by its users.

Domain knowledge on top of product models

Concept

The presumption is that the automation of product modelling is the mainstream concept. Furthermore, it is evident that engineers gather around product modelling technologies with the broadest user base and best-practice examples. Today's most popular technologies with industrial automation systems are STEP and IFC.

Based on the positive and negative experiences in modelling of building products a concept of the Virtual Product Model (VPM) is proposed, which could preserve positive, and avoid some negative characteristics of product models.

The virtual product model is represented by a network of loosely coupled particle models, interconnected by relatively simple but strong rules (like gravity in the macro-cosmos). The neighbourhood of a particle model is in a logical sense defined through a process model, which also determines relations between particles. So the main point is in the communication network that interconnects the particles.

First consider the concept, which solved the complexity problems of computer networks. The problems of how to get together many different communication technologies seemed unsolvable, until a cut has been made in decomposing the network into clearly defined functional layers – a solution nowadays known as the 7-layer ISO model [10].

Decomposition has also become a magic word in software development. Huge monolithic systems tend to evolve into open and flexible structures of software components [15].

The virtual product model can be explained as a decomposed product model, consisting of the three main levels (Figure 2):

- 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).

Our integration model can also be viewed as a multilevel integration model ([7], [28]) 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 Figure 2) 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 this 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 adaptibility towards foreign product models.

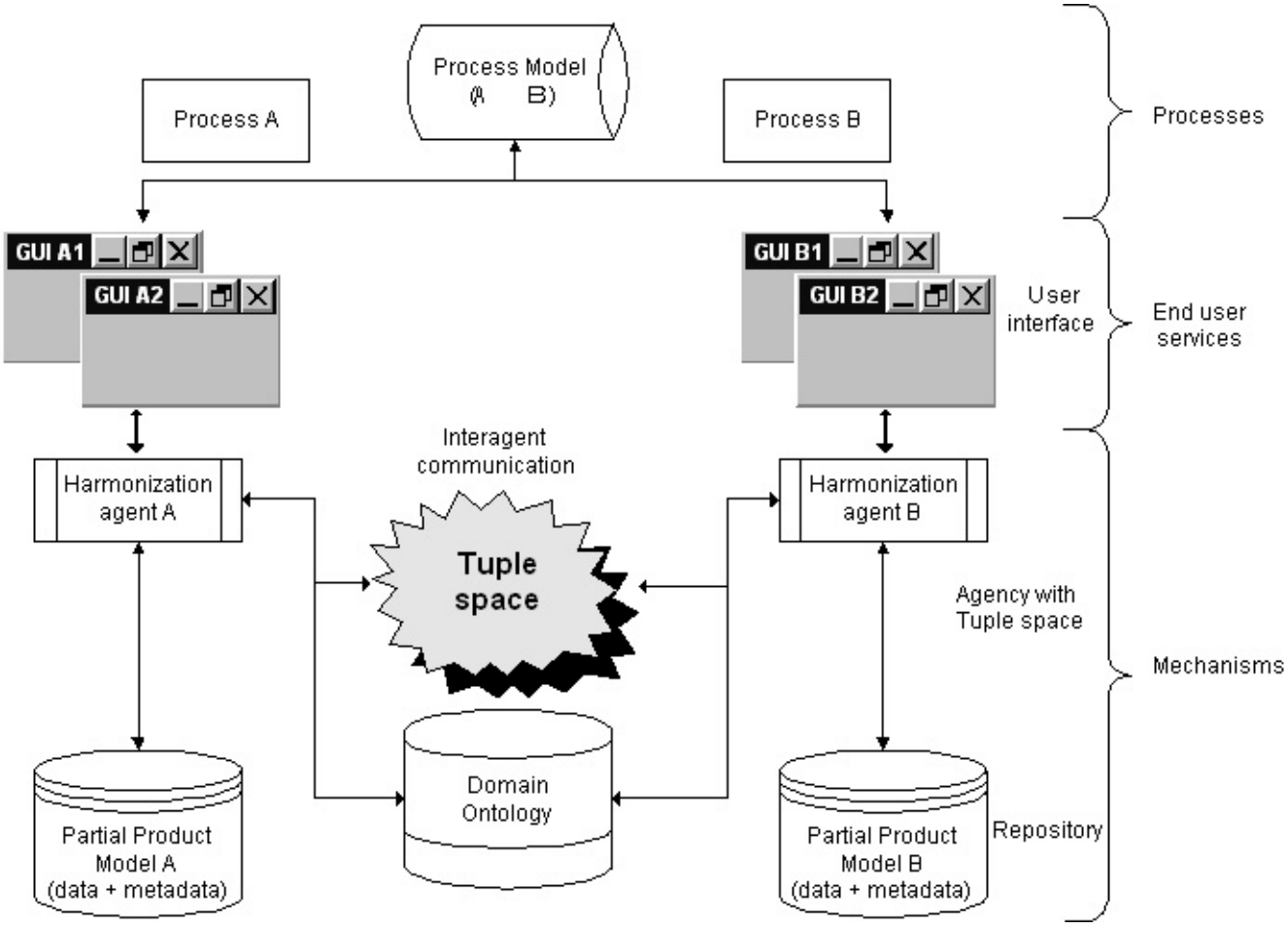

Figure 2. Integration model for Virtual Product Model

System and data heterogeneity

In order to implement communication mechanisms for the Virtual product model we have investigated the integration issues that arise in a federation of heterogeneous data sources, possibly storing related information. In an integrated heterogeneous information system, system and data heterogeneities among partial systems are the major obstacles toward information sharing. While system heterogeneity problems are usually resolved by using a common communication infrastructure (e.g. TCP/IP, RPC, CORBA ORB, Java RMI, etc.) the differences in information and data models are more difficult to deal with. It is not possible to force all users to use the same data or information model for different applications since one data model, which is suitable for one application domain, may not be suitable for another. Besides people working in different application areas may have their own preferences on which models to use.

The solution to this problem is either to do pair-wise translations of modelling constructs or to use a common or neutral model, to and from which all data models used by the users are translated. The data model heterogeneity problem needs to be resolved before any successful system integration can be achieved.

Problems with data heterogeneity have been thoroughly investigated and can be generally categorized in two types: schematic and semantic heterogeneity. Schematic heterogeneity is due to different ways of naming and structuring data, whereas semantic heterogeneity is due to different representations of data values. Because of these problems, queries issued between information systems with partial product models cannot be processed directly and data returned from them cannot be readily used. Query and data conversions are needed to bridge the naming, structural and semantic gaps.

Two basic approaches have been taken to deal with the problems of data heterogeneity. The traditional approach is to establish a global schema, which reconciles the naming and structural conflicts and unifies the semantic representations of heterogeneous component systems [9]. This integrated global schema is then shared and referenced by the users to exchange queries. Thus, all conflicts and differences are resolved at the time of schema design and integration. The major drawback of this approach is that the shared integrated schema forces users to view data in the same way instead of the ways that are familiar to them.

Mediation versus integration

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.

Tuple space

Our model supports a distributed enviroment of network–aware applications where structural and semantical disharmony is resolved through a multi-agent system. Each component in the model is assigned an agent that intercepts queries for product model data sent to another component. The query is stored in a shared data space. Information in the space is organized in tuples and retrieved in an associative way through a pattern-matching mechanism by an agent assigned to another component. The concept of shared data space accessed in an associative way originates from the coordination model Linda [1]. Several recent proposals for interactive agent applications use Linda-like coordination model: Java-Spaces [20], J-Spaces [21] and MARS [8]. In contrast with client-server coordination where communicating components must explicitly name their communication partners and be directly connected during the communication (spatial coupling), use of tuple spaces allows components to communicate without synchronizing their activities (spatial uncoupling). Use of tuple-space ideally suits our Virtual product model concept.

Architecture

The emergence of data modelling languages has introduced the need for higher level of abstraction at which we can create and compose heterogeneous data models. The problem can be examined by providing a formal semantics to the composition of different data models within a virtual team, i.e. the problem of composing a product model from heterogeneous data sources. On top of this we can build a multi-agent system where intelligent agents take care about query processing that come from multiple sources.

Relations between components are of most importance. From the aspect of the model as a whole the proper relations should assure the integrity of the model. Therefore special attention has been given to the harmonization of the content of component models, which are representing parts of the virtual product model. The mechanism is based on harmonization agents, which are leaving the components their individuality but also bind them to the whole.

Harmonization agents do not require uniform semantics of component models, but only common basic primitives. It is therefore possible to allow different structuring and representation techniques and standards for component models. While communicating harmonization agents use their own knowledge about structures, which they gathered and saved in common dictionaries, whereby in insolvable situations agents establish contact with humans. Actually, agents will in the first stage act as assistants, then as advisors and at the final stage as autonomous agents.

The common dictionary is a repository of basic element (term) descriptions in a semantic domain. There is a domain dependent starting set of terms with relations between them, which are necessary for communication start-up between agents. It is supposed that agents will soon come into situations where basic terms and relations won't be enough to exchange views in a new situation. In such cases, agents will have to ask an expert - either a human or another, more "experienced" software agent. For the second case, agent "chat rooms" will be the place, where "inexperienced" agents will have the opportunity to learn, and then improve their "native" dictionary. Building a dictionary automatically from very simple starting terms should avoid the known trap of defining complex view mapping schemes. A related work is LexiCon [31] which is a tool that classifies construction terms in an object-oriented way as instances of *Built Object*s. A *Built Object* is a top level class, their instances have real-world *Names* like airport or bolt. The LexiCon is a result of an ISO effort to link product models with Framework for classification of information (ISO/TR 14177:1994).

To become a part of the VPM the particles (applications + data structures) have to fulfil certain conditions regarding data representation:

- all exchange data has to be available in an external representation in a text form (which also implies database systems able to communicate in text form),
- a data description (metadata) has to be available,
- data has to be structured in an object-oriented way, using Express, XML, but also non-standard description languages.

Having in mind that it is not necessary to adapt the semantic and the data structure of a particle to integrate it with the product model, even old computer programs can be upgraded to suit the conditions. It is, however, a good idea to redesign the data structures to give components a better ability to interact with others.

On the other hand, the VPM concept, which supports the flexibility and autonomy of applications, is a good accelerator of application's (particle's, component's) self-intelligence and adaptability.

Example

Figure 3 shows the application of the VPM concept in the road life cycle (for the last few years road as a building object has been the focus of our product modelling research; see [26]). Top part of the figure shows early phases in the road life cycle process. Corridor definition preceeds phase of geometrical design. RO/Corridor is an example tool for corridor definition, Plateia is a road design software. Results (data) of the two phases are stored in the road product model called MCT. Road geometry is a prerequisite for project management analysis and cost estimation. At some point in the process emission analysis needs to be performed (this is supported by a software called Dynem). The software needs 3D data about road axis for emmision calculation. If heterogeneity is assumed in its broader sense then there is no need for Dynem to know in advance that the Plateia owns the requested data. In a traditional software environment data interfaces between heterogeneous software applications must be defined in advance.

When the user specifies the project name he wants to work on, Dynem tries to read relevant data.

The agent that "works" for the Dynem creates a new tuple in the Tuple space. The tuple contains 3D data request. The Tuple space acts as the meeting point for participating agents. The read request is intercepted in an asynchronous way by the Dynem's harmonization agent through the tuple 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) in the tuple space. Location of the data source is not determined in the process model since it only accesses (reads, writes) the tuple space. The Plateia-agent, which is responsible for the geometrical design, detects the read request (tuple1) in the tuple 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 VPM 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 principle of the VPM will be especially effective in civil engineering, where processes and partners, as well as applications used, are changing from project to project. However, same component applications are used more often, which makes it possible for their agents to improve knowledge repositories.

Conclusion

The concept of the product model is a result of human's mental activity and desire of mastering the whole to the smallest possible detail. Especially the development of product models shows that the human has, as so often before, ignored the natural laws as well as himself. He only relied on his own mental constructs and has equated them with the objective reality. His models work only under special circumstances, but they are not generally applicable. This does, however, not mean that the concept of product models is useless. It is the opposite, inevitably needed. The remaining question is, what still makes it a "Holy Grail"?

Through the concept of the virtual product model it is believed that it is possible to preserve the independence and flexibility of particles - existing island models and applications, and the simplicity of mastering them, but also to preserve the positive integration effects of complex product models. The reason for this belief lies in the simplicity of the principles used and in their closer relation to natural mechanisms (basic laws), which also includes the ability of implicit evolution. The evolution and improvement is supported by harmonization agents, which not only communicate, but through the communication also gain new knowledge and develop adaptability. This is actually a new perspective that bridges the gap between the worlds of knowledge management and product data technology. We believe that our approach would enable more transparent information exchange in construction process regardless of semantic and structural heterogeneity of data. In other words, we have tried to find a mechanism to preserve the advantages of product models while avoiding the shortcomings caused by their complexity, having in mind the words of the German philosopher Oswald Spengler "Everything complex is of short lifetime".

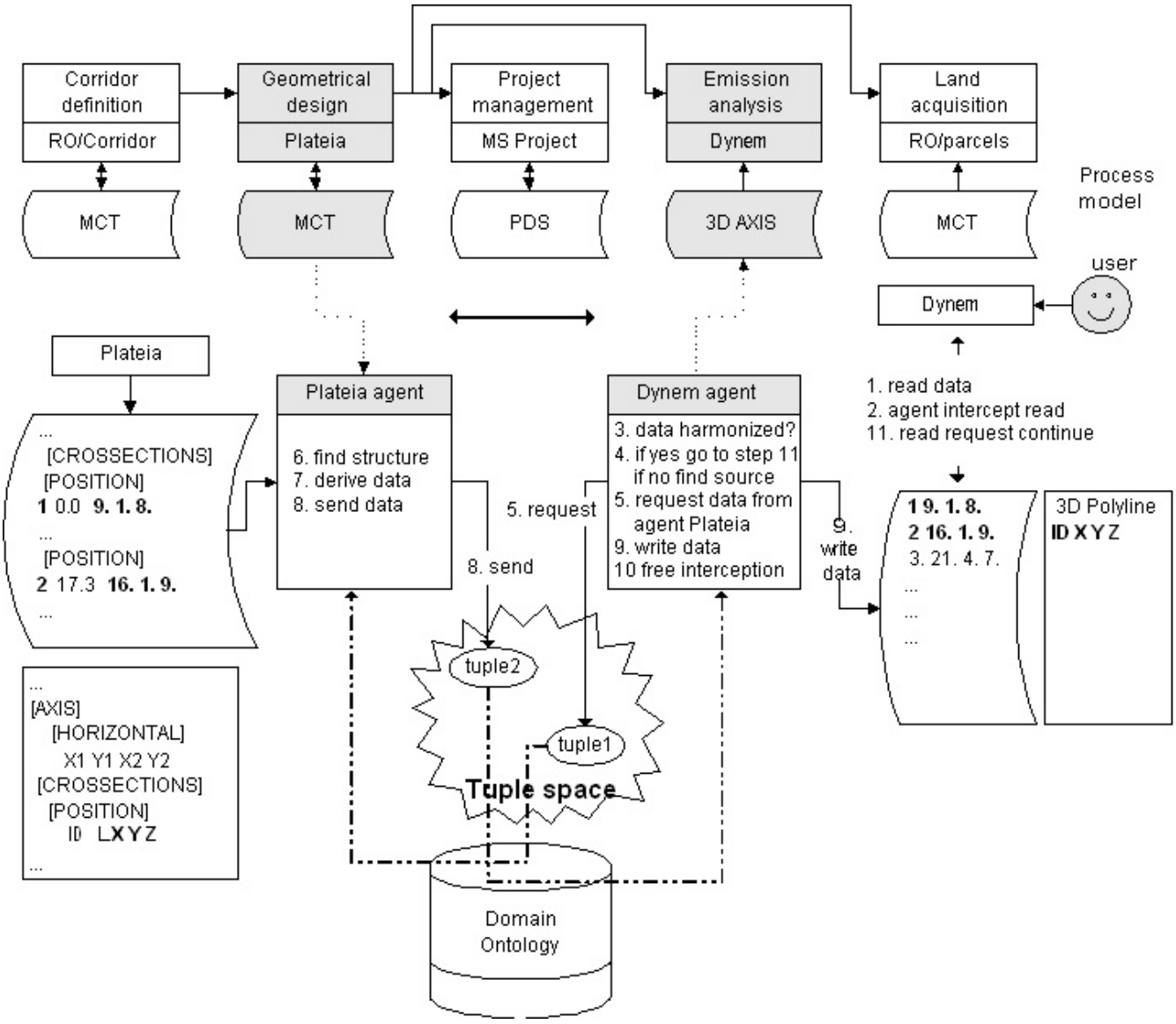

Figure 3. Example of VPM mechanism operations

REFERENCES

[1] Ahuja S., Carriero N., Gelernter D. 1986. *Linda and friends*, Computer, vol. 19, no. 8, Aug. 1986, pp. 26-34

[2] Ammerman E., Junge R., Katranuschkov P. & Scherer R.J. 1994. *Concept of an object-oriented product model for building design*. Dresden: Technische Universität.

[3] Amor R. 1998. A UK survey of integrated project databases. *Proceedings of the CIB W78 conference The life-cycle of construction IT innovations*. Stockholm: The Royal Institute of Technology.

[4] Augenbroe G. 1993. *COMBINE, Final Report*. Delft: Delft University.

[5] ATLAS 1992. *Architecture, methodology and tools for computer integrated large scale engineering – ESPRIT project 7280*. Technical Annex Part 1, General Project Overview. http://www.newcastle.research.ec.org/esp-syn/text/7280.html (15.3.2002).

[6] Björk B.C. 1989. Basic structure of a proposed building product model. *Computer Aided Design* 21(2): 71-78.

[7] Brown et al. 1994. Principles of CASE Tool Integration. *Oxford University Press, Oxford*.

[8] Cabri G., Leonardi L., Zambonelli F. 2000. *MARS: A Programmable Coordination Architecture for Mobile Agents*, Internet Computing, vol. 4, no. 4, July 2000, pp 26-35

[9] Chen, L. & Arbee, P. 1988. *Schema integration to support multiple database access*. Bellcore technical report TM-STS-012948

[10] ISO 8648:1988. Open Systems Interconnection – Internal organization of the network layer

[11] 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.

[12] Faraj I., Alshawi M., Aouad G., Child T. & Underwood J. 1999. Distributed Object Environment: Using International Standards for Data Exchange in the Construction Industry, *Computer-Aided Civil and Infrastructure Engineering* 14(6): 395-405.

[13] Gallicon 2001. Integrated Inormation exchange Towards an Industry-Wide Process Improvement with Particular Emphasis on Water and Housing Industry Projects . *Supporting Partners: Industrial report.* University of Salford.

[14] Graves G. 1998. Industry requirements for data standards harmonization. *Proceedings of the Global Business Solutions for the new millenium*. CD ROM.

[15] IEEE 1997. Engineering meets the internet: how will the new technology affect engineering practice? *IEEE Internet Computing* 1(1). Vol. 1. no. 1. pp. 30-38. 1997. IEEE Computer Society.

[16] IAI 2002, Industrial Alliance for Interoperability, http://cig.bre.co.uk/iai_uk/ (15.3.2002).

[17] IFC2 2002, IFC 2x Final – latest IFC platform, Industrial Foundaton Classes, http://cig.bre.co.uk/iai_uk/Press_release2.htm (15.3.2002)

[18] IFC3 2002, Latest IFC Release 3 extension model, http://cic.vtt.fi/niai/technical/ifc_3/ (15.3.2002)

[19] ISO 10303 2002. Collection of ISO STEP 10303 standards, http://www.iso.ch

[20] JavaSpaces 2001. http://java.sun.com/products/javaspaces/ (20.7.2001)

[21] J-Spaces 2001. http://www.j-spaces.com (18.7.2001)

[22] Larson M. 1998. AF integrated digital environment. *Proceedings of the Global Business Solutions for the new millenium.* CD ROM.

[23] Pfennigschmidt S., Kolbe P. & Pahl P.J. 1997. Integration von Datenmodellen. *Proceedings of the IKM conference*. CD-ROM. Weimar.

[24] Rebolj D. 1990. Graphic Modelling of Superstructures. *Automatika* 31(1-2): 147-156.

[25] Rebolj D. 1993. *Computerunterstützter integrierter Straßenentwurf in einer objekt-orientierten Umgebung*. Graz: Verlag für die Technische Universität.

[26] Rebolj D. 1999. Integration of computer supported processes in road life cycle. *Journal of transportation engineering* 125(1): 39-45.

[27] Spooner D.L. & Hardwick M. 1997. Using views for product data exchange. *IEEE Computer Graphics and Applications* 17(5): 58-65.

[28] Stavridou V. 1999. Integration in siftware intensive systems. *The Journal of Systems and Sofware* 48 (1999): 91-104.

[29] Syal M.G., Parfitt M.K. & Willenbrock J.H. 1991. *Computer integrated design/drafting, cost estimating, and construction scheduling. Housing Research Center Series Report No. 11*. The Pennsylvania State University, Dept. of Civil Eng.

[30] 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.

[31] Woestenenk K. 1999. LexiCon – A common vocabulary for the construction industry. Proceedings of the 2nd international conference on concurrent engineering in construction (CEC99), Espoo, Finland, 25-27 August 1999,

[32] Yau N.J., Melin J.W., Garrett J.H. & Kim S. 1991. An environment for integrating building design, construction scheduling, and cost estimating. *ASCE Seventh Conference on Computing in Civil Engineering and Symposium on Databases*. Washington, D.C.: ASCE.

[33] Yu K., Froese T. & Grobler F. 2000. A development framework for data models for computer-integrated facilities management. *Automation in Construction* 9(2): 145-167.