Virtual Construction Using Map-Based Approach

V. Semenov, E. Alekseeva, O.Tarlapan, Institute for System Programming of the Russian Academy of Sciences (ISP RAS), Moscow, Russia, (step@ispras.ru)

Summary

The paper presents a general map-based approach to prototyping of products in virtual reality environments. Virtual prototyping of products is considered as a consistent simulation and visualization process mapping the source product model into its target visual representations. The approach enables to interrelate formally the product and visual information models with each other by defining mapping rules, to specify a prototyping scenario as a composition of map instances, and then to explore particular product models in virtual reality environments by interpreting the composed scenario. Having been realized, the proposed approach provides for the strongly formalized method and the common software framework to build virtual prototyping applications. As a result, the applications gain in expressiveness, reusability and reliability, as well as take on additional runtime flexibility.

Being oriented on STEP family standards for the computer-interpretable representation and exchange of product data throughout the whole life cycle, both the presented general approach and the developed software framework allow a lot of potential applications for different industry branches. These solutions are illustrated by a virtual construction application developed to study IFC-driven building models in 4D virtual reality environment. The application enables its users to visualize multi-disciplinary architecture and engineering information as well as to plan and to control construction processes at the building site.

The work is supported by RFBR (grant 04-01-00527) and Russian Foundation for Science Support (http://www.science-support.ru).

1 What is virtual prototyping?

What is virtual prototyping? The answer to this question is rather not trivial. There are a lot of particular examples of “right” applications like virtual mannequins in textile industry, virtual wind tunnels in airbus industry, virtual buildings in architecture, engineering and construction. But there is no yet systematic understanding of what constitutes virtual prototyping would be meaningful for the product life cycle, sound for its matter and features, and attractive to natural models, human perception and mental vision. Nevertheless, transforming or mapping product data into some perceptual representations allowing clear visual interpretation can be considered as the underlying principle for virtual prototyping.

Indeed, the traditional conveyor (see the figure 1) can be employed to represent typical virtual prototyping phases such as requirement analysis and statement of the problems to be studied, simulation of product features, visualizaiton of results, modeling of visual scenes and their rendering (Felger et al. 1994). At the first phase the source product data and technical requirements are preprocessed to form a computational model for the investigated problems. Then the mathematically stated problems have to be solved to simulate the product phenomena. At the postprocessing phase the simulation results are improved to exclude redundancy and to enhance their content. The visualization phase maps the derived results from one symbolic representation to another having explicit visual meaning. At the next phase the obtained visual representation may be converted to virtual reality modeling scenes supported by popular Internet browsers and graphic software systems. At the final phase the modeled scenes are rendered and displayed as digital images that are targets of the whole conveyor. The generated images can be immediately observed and analyzed to change the conditions upon which virtual prototyping is accomplished or modify the source product data not satisfying the imposed technical requirements.

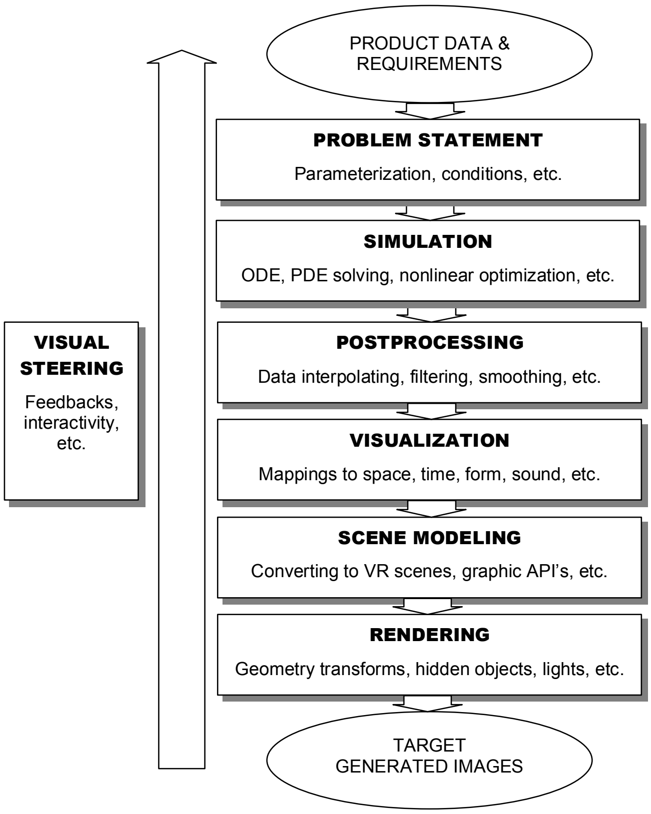

Figure 1. Map-based representation of the virtual prototyping conveyor

It is essential that the whole conveyor and all of its phases can be represented as compositions of maps performing partial transformations of classified data each into other. Each map realizes a basic transform rule according to that the classified data units driven by one information model have to be mapped into the corresponding units belonging to the same or another model. In the considered context maps realize simulation and visualization transforms constituting the whole virtual prototyping conveyor. The circumstances above force to look at virtual prototyping through the map-based paradigm.

Because of high scales and complexity of the recent product information models like STEP application protocols (ISO 1994) and Industry Foundation Classes (IFC) by IAI (IAI 1999), an implementation of virtual prototyping applications may be a difficult task as the significant part of the comprehensive multi-disciplinary information has to be processed and interactively managed. Often several alternative methods and scenarios have to be applied to visually represent the product information in ways adequate to the studied phenomena by reproducing some visual metaphor. Therefore, such applications may be sophisticated and need to be developed and evolved using some formal approach and common software framework.

The need to create such framework is also motivated by numerous EU and national research projects (DIVERSITY, OSMOS, e-Construct, IST-forCE, e-COGNOS, ViSICADE, CREATE, TOURBOT, ART NOVEA) aimed at particular virtual technologies and facilities for acoustic and thermal simulation, construction and space planning, urban visualization.

In the section 2 we outdraw a formal approach to defining syntax and semantics of maps. Here we follow object-oriented modeling of the product and visual information. In the section 3 we present a general-purpose framework to realize the formalized map-based approach. The section 4 gives an example of virtual construction application illustrating the approach. The application is described with emphasis on practical benefits of the approach proposed. Its advantages are shortly summarized in the conclusions.

2 Formal approach

We introduce a formal approach to virtual prototyping by first looking at concepts representing a meta-model suitable for specification of product and visual information. The data modeling language EXPRESS is considered here to illustrate the concepts introduced and to prove their feasibility with respect to the industry-specific models regulated by the STEP family standards. Then we proceed with formalization of mappings interrelating product and visual models as well as impose rules according to which visual scenes can be unambiguously interrelated with product data. The formalization conducted is similar to the way exploited in semantic analysis of languages (Breu et al. 1997).

2.1 Object-Oriented Meta-Model

So, we define an object-oriented meta-model as a structure $M = (C, \prec, Attr_C, Rule_C)$ with the following meaning:

— C is a set of classifiers that introduce distinct entity types. The subtype hierarchy induced on classifiers by generalization/specialization is utilized to employ a kind of inclusion polymorphism;

— $\prec$ is a partial order on C reflecting the generalization hierarchy of classifiers and entity subtyping;

— $Attr_C$ is a set of attributes or, more correctly, a set of operation signatures $a_c : c \mapsto s$, $a_c : c \times s \mapsto c$ for functions accessing the attribute values. Each attribute of an object of type $c \in C$ represents a property given by a value of some sort s. We connect a sort s having the same name with each classifier, admitting the attributes can represent association properties too;

— $Rule_C$ is a set of constrains in terms of which the behavioral characteristics of entities are defined. They may limit the number, kind and organization of attributes as well as impose relationships among their values. Formally, the rules are a set of operation signatures $r_c : c \mapsto boolean$.

The attributes assume to be of basic sorts S_D mapped into basic semantic domain D, complex sorts $S_{\overline{D}}$ mapped into multi-valued semantic domain $\overline{D}$, and class sorts C. In a way quite similar to algebraic specification, we provide a signature $\Sigma_M = (S_D \cup S_{\overline{D}} \cup C, \prec, \Omega_D \cup \Omega_{\overline{D}} \cup Attr_C \cup Rule_C)$, where Ω_D, $\Omega_{\overline{D}}$ are sets of operations defined on basic and complex sorts. Here a partial order $\prec$ on sorts extends the order of classifiers and generalizes subtyping relations. The signature formed by such way describes all of the sorts and the operation symbols belonging to the information model M as well as contains the initial set of syntactic elements upon which the expressions $Expr(\{var_s\}_{s \in S_D \cup S_{\overline{D}} \cup C})$ with variables indexed by the sorts can be defined.

The set of objects with attribute values and associations established among them, constitutes the state of a model σ_M. In conformity with our discussion, the state σ_M is a set of objects composing a product or visual model depending on semantics of the domain model M.

So, EXPRESS introduces basic sorts {*Real, Integer, Number, Boolean, Logical, String, Binary*} $\subseteq S_D$ corresponding to simple data types of usual semantics. Complex sorts {*Generic*

Aggregate, Bag, Set, List, Array, Enumeration, Select, Defined} $\subseteq S_{\overline{D}}$ correspond to different types of aggregates, enumerated types, selections, and redefined types that permit definition of derived nested data structures that cannot be avoided in non-trivial product and visual models. Possible additional categorization of sorts does not prevent the formalization conducted above.

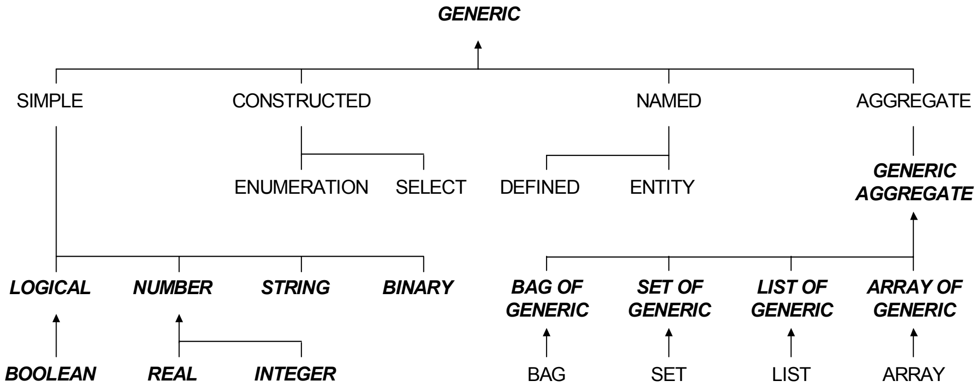

Figure 2. A classification of the underlying types available in the EXPRESS language

The figure 2 gives a general classification of the types available in the EXPRESS language. The sorts marked by italic bold font correspond to its underlying types. Arrows indicate subtyping relations between them. The virtual sorts marked by normal font are used only to consistently classify the language constructions and to represent user-defined types. For briefness we omit deeper formalization, addressing to (Breu et al. 1997).

2.2 Defining Map Semantics

On conditions that information models for product and visual domains have been specified, a mapping scheme can be defined as a system $R_{M_S M_T} = (M_S, M_T, Map, \prec, Pr_{Map}, Ps_{Map}, Link_{Map}, Attr_{Map}, Qr_{Map}, Qs_{Map})$, where:

— M_S, M_T are models of source and target domains with meanings defined above. Here we admit that they may differ, intersect or coincide. If the models are the same, then the scheme $R_{M_S M_T}$ corresponds to some transform mapping the model M_S into itself;

— Map is a set of bi-directional map sorts $m(l'_1,\ldots,l'_k,l''_1,\ldots,l''_l,l'''_1,\ldots,l'''_n) \in Map$ with input, mixed and output links $l'_1,\ldots,l'_k,l''_1,\ldots,l''_l,l'''_1,\ldots,l'''_n \in Link_{Map}$ belonging to corresponding type sorts of the source and/or target models $s'_1,\ldots,s'_k,s''_1,\ldots,s''_l,s'''_1,\ldots,s'''_n \in S_{D_S} \cup S_{\overline{D}_S} \cup C_S \cup S_{D_T} \cup S_{\overline{D}_T} \cup C_T$. Each map *m* is represented by pairs of signatures $m_f : m \times s'_1 \times \ldots \times s'_k \times s''_1 \times \ldots \times s''_l \mapsto s''_1,\ldots,s''_l,s'''_1,\ldots,s'''_n$ for forward function mapping the source objects into target ones and $m_b : m \times s''_1 \times \ldots \times s''_l \times s'''_1 \times \ldots \times s'''_n \mapsto s'_1,\ldots,s'_k,s''_1,\ldots,s''_l$ for a backward function. The links play role of formal parameters for the mapping functions as well as for other operations suggested by the map concept. Each map specifies what parameter sort combinations, including basic, complex and class sorts, could result in generation of target representations or in reconstruction of source representations and how this should be done, basing on concrete preconditions Pr_{Map}, postconditions Ps_{Map} and attribute mapping operations $Attr_{Map}$. Here we implicitly suggest that a target scene can be generated from the comprehensive source information, and the source data can be consistently managed in varying the target scene;

— $\prec$ is a partial order on *Map* reflecting the generalization/specialization relations between maps;

— Pr_{Map}, Ps_{Map} are sets of preconditions $pr_m : m \times s_1' \times \ldots \times s_k' \times s_1'' \times \ldots \times s_l'' \mapsto boolean$ and postconditions $ps_m : m \times s_1'' \times \ldots \times s_l'' \times s_1''' \times \ldots \times s_n''' \mapsto boolean$ to reveal the potential cases when forward and backward maps should be actually executed. They can also be applied to control correctness and consistency of the results;

— $Attr_{Map}$ is a set of attribute operations $a_m : m \mapsto s$, $a_m : m \times s \mapsto m$ for functions accessing the attribute values. Each attribute of a map of the type $m \in Map$ represents an association property given by the value of this sort *m*. Thus, the maps can be interconnected with each other;

— Qr_{Map}, Qs_{Map} are sets of quality rules $qr_m : m \times s_1' \times \ldots \times s_k' \times s_1'' \times \ldots \times s'' \mapsto number$, $qs_m : m \times s_1'' \times \ldots \times s_l'' \times s_1''' \times \ldots \times s''' \mapsto number$ for prior and posterior estimating the results derived from execution of maps. Here we assume that a degree of quality and usefulness of the results can be expressed by means of the numeric sort *Number*.

It is important that the order of type sorts of the source and target models induces a partial order on maps. Maps $m(l_1', \ldots, l_k', l_1'', \ldots, l_l'', l_1''', \ldots, l_n''') \in Map$, $\overline{m}(\overline{l_1'}, \ldots, \overline{l_{\overline{k}}'}, \overline{l_1''}, \ldots, \overline{l_{\overline{l}}''}, \overline{l_1'''}, \ldots, \overline{l_{\overline{n}}'''}) \in Map$ can be connected by a relation $\overline{m} \prec m$, if $k \le \overline{k}$, $l \le \overline{l}$, $n \le \overline{n}$, $\overline{s_i'} \prec s_i'$ for $\forall i = 1, \ldots, k$, $\overline{s_i''} \prec s_i''$ for $\forall i = 1, \ldots, l$, and $\overline{s_i'''} \prec s_i'''$ for $\forall i = 1, \ldots, n$. The partial order on *Map* implies a subset condition on data identifiers to be mapped potentially. If the data identifiers satisfy to the map $\overline{m}$ and $\overline{m} \prec m$, then they satisfy to the map *m*. It means that the same data set or object population can be processed in evaluation of allied maps. Natural choice consists in applying the map most suited for the particular data set by the mapping function signature. For this purpose data sorts and map link sorts can be compared in a way similar to lexicographical comparison of strings to produce a sound quality estimate and to take a final decision based on the valued estimates. Certainly, there may be the other application-specific ways to define quality estimate functions for the maps.

The mapping scheme $R_{M_S M_T}$ allows meaningful implementation $R_{M_S M_T} : \sigma_{M_S} \to \sigma_{M_T}$ converting the state of the model M_S in the state of the model M_T or, in other words, transforming object populations relating to the model M_S into object populations belonging to the model M_T. If the model M_S semantically means some product model and M_T do some visual model, then the implementation will correspond to the virtual prototyping process as such. Besides, being a bi-directional mapping scheme, $R_{M_S M_T}$ may permit the implementation $R_{M_S M_T} : \sigma_{M_T} \to \sigma_{M_S}$ reconstructing the source product data in accordance with the varying visual scene. Although applications often provide for interactive capabilities based on manipulations with objects of a visual scene and their transforms to a source domain-meaningful representation, we do not require mandatory existence of backward maps in the general case. In the following discussion we will consider the approach by looking only at forward maps, admitting that backward transforms could be applied similarly.

Let us consider the underlying principles and mechanisms essential for the introduced concept of map. These are of significant value for building a wide range of applications using the map-based programming approach.

Principle A. Map inheritance. Allowing a classification, maps can be represented by object types within the object-oriented meta-model discussed above. The partial order $\prec$ on *Map* reflects a generalization hierarchy of map classifiers and map subtyping. We define the map inheritance as a mechanism regulating succession rules for attributes, links, map execution operation, pre-, post-conditions, and quality estimates in super- and sub-maps absolutely similar to inheritance routine commonly applied within object-oriented approach. Attributes and links are interpreted as data encapsulated by the maps, whereas evaluation operation, conditions and quality estimates are considered as their methods.

Principle B. Structural specialization and behavioral polymorphism. The principle assumes that the structural properties of maps like attributes and links can be defined as optional ones, in that way admitting their values may be missing without any losses for consistency and transformation semantics of the maps. These properties can be also redefined in the inherited maps by specialization of their types. Behavioral properties like map execution operation, pre-, post-conditions, and quality estimates can be polymorphicly redefined, thereby allowing more meaningful and efficient implementations at the level of more specialized inherited submaps.

Principle C. Aggregative maps. Sometimes in applications there is a need to simultaneously apply transformations coincident or very close by function signatures and to union the final results in correspondence with some routine. For that, the allied maps can be grouped using so-called aggregative maps. Each aggregative map instance represents a container of nested base map instances that share common links. Execution of the aggregative map consists in subsequent execution of the aggregated base map instances in the order predetermined by the specified aggregate sort and its admissible dimension range. Thus, the aggregative map combines particular behavioral properties of the separate aggregated maps.

Principle D. Selective maps. Selective map is a construction similar to map aggregates except for only one base map instance has to be applied to generate output results. The selection criteria may differ depending on the application purposes. They can be based on random choice, prior and posterior estimates to select the best result among the delivered ones. Typical implementations are provided for these criteria, if only another implementation has not been provided specially.

Default implementation of prior estimates consists in calculation of maximum difference in specialization levels for formal and factual input parameters of the map. In such criteria the base map most near by mapping function signature to factual parameters has to be selected. For posterior estimates the other techniques can be used that calculate specialization degree for output results of the map and their correspondence to the target schema. A base map is taken as the best one if all the generated output results belong to the target schema and occupy the lowest positions in the hierarchy of types corresponding to the most specialized basic, complex, and class sorts. These general and constructive techniques provide the choice meaningful for a lot of use cases.

Principle E. Dependent maps. Dependent map is a sort of usual maps some links of which have map types. Dependent maps can be interrelated with structural and behavioral properties of the linked base maps. The reason to support such constructs is a need to parameterize derived map realizations by the other maps. Essentially that various implementations of the dependent map can be derived as a result of assigning different base map instances. It can be easily done without any changes in the specifications and implementations for these maps.

Principle F. Composed maps. Structural composition of more complicated maps from already defined ones is one of the underlying principles of the approach presented. Composed maps are represented as ordered sets of map instances interconnected with each other via links. The maps are interconnected by assigning the output results of one map to input of another map. Composed maps are evaluated by sequential invocating the mapping functions of separate maps in the order they have been enumerated. Unlike usual maps, there is no need to supply map execution, condition, estimate functions for the composed maps since the comprehensive structural representation is available for their evaluation. Moreover, being structured, composed maps can be applied for both forward and backward data mappings.

Principle G. Map-based scenario. Finally, we introduce scenario as a concept very similar to the composed maps. But we assume that a mapping schema can contain only one scenario corresponding to the main map from which evaluation of the whole mapping schema has to be started. The scenario includes built-in input and output links of the predefined generic type via which data are imported and exported in some meta-formats. More details can be referred to implementation issues, discussed in the next section.

3 General-purpose application framework

The map-based approach allows both declarative and imperative capabilities for the realization. Here we very shortly present a general-purpose framework intended to build virtual prototyping applications employing this approach.

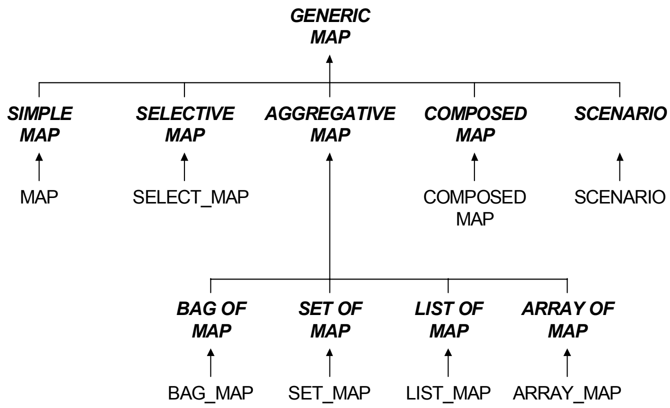

Figure 3. A hierarchy of map classes of the general-purpose framework

The discussed framework is a consistent system of C++ classes to represent arbitrary model-driven data and maps of the introduced above categories, to introspect both data and map instances using metadata dictionaries, to check their consistency, as well as to evaluate the specified maps and the composed scenarios. Essentially that all of these actions on data and map management are accomplished by the framework at runtime, so there are provided capabilities to define new virtual prototyping scenarios, to change and to execute them as the application session is running. A hierarchy of the map types supported by the framework is presented at the figure 3. To significant degree the map hierarchy reflects the principles suggested by the map-based approach.

The framework is compliant with STEP SDAI and provides for all the STEP declared capabilities to export/import product data and to access them via software interfaces. Due to compliance with the STEP family standards and the EXPRESS data modeling language, uniform and harmonic manipulations with both product data and map-based transforms are becoming possible and making development of sophisticated software applications more transparent and easy. In that sense, the framework design can be considered as an evolution of STEP SDAI interfaces to manage both product data and computations defined over them.

To simplify development of the software components typical for product visualization applications, the framework provides map libraries for geometry modeling problems. In particular, the offered libraries implement visualization transforms with canonical geometry primitives and set-theoretic operations (subtraction, intersection, union) over arbitrary polyhedrons.

Leveraging both object-oriented and map-based programming paradigms, the framework allows rapid prototyping and building of software applications for different industry branches and purposes. The built applications gain in expressiveness, reusability and flexibility.

4 Virtual construction application

The approach and the framework developed have been used to build a software application intended for virtual construction. The application is oriented on the recent needs of architecture, engineering, construction, and facility management (AEC/FM) industry to be possessed of effective tools for mocking-up building projects, modeling and controlling construction processes in virtual reality environments. These tools would allow to improve understanding between different stakeholders participating in a joint project like architects, engineers, economists, construction managers, to eliminate potential drawbacks and errors at earlier phases of the project, and to accelerate its performing.

4.1 General functionality

Using the approach, we need define source and target models properly describing product and visual domains. Certainly, by adapting some available standards already recognized by wide industry communities, the application would gain in interoperability and practical value. Therefore, the application built employs the international information standard IFC as a source model and Virtual Reality Modeling Language (VRML/X3D) designed and implemented by Web3D consortium (ISO, 1997) as a target model. Since IFC has been permanently evolved, the versions IFC 2.0 and IFC 2x has been chosen for implementation purposes. These releases have been found significant acceptance in a lot of AEC/FM industry domains as well as have been recently supported by numerous software applications. The VRML97 has been taken for the similar reasons as it has obtained recognition of the Internet community as a standard language for describing interactive 3D worlds within the Web.

Because of both VRML97 and X3D are specified in informal way, EXPRESS schemas for this standard have been developed to harmonize the source and target information models as well as to uniformly manage them within the application framework. The detailed specifications of the developed EXPRESS schemas for the VRML97 and X3D information models can be found at the web site http://www.ispras.ru/~step.

Obviously that visualization of such complex multi-disciplinary information as IFC-driven product data in VRML-accompanied virtual reality environments, probably, using variety of methods is a non-trivial task, and the presented map-based approach and framework seem to be promising to be employed for this purpose. The additional motivation is that the application built has to allow adjusting the visualization facilities to study various aspects by different user categories. The attractive advantage of these solutions is that they provides for capabilities to compose and to execute working scenarios directly at runtime. It is the tempting feature making the applications more customizable and flexible.

4.2 Working scenarios

Let us consider typical working scenarios illustrating the underlying map-based approach and the advantages of the developed virtual construction application. The scenarios allow resolving some problems related to visualization and analysis of source IFC-driven data and composing target VRML scenes.

To realistically visualize building product data, materials of appropriate visual properties can be assigned to correspondent types of building, furnishing, electrical, transport, equipment, flow distribution elements assumed by the IFC information model. Certainly, as the collection of elements may count hundreds of types, users are becoming not able to distinct the assigned materials and to identify the elements adequately. A possible solution consists in assigning the materials to wider categories of elements in accordance with user preferences. But for this case we need develop a scenario according to that the concrete material has to be selected among the already assigned ones to suit better for each particular element type.

In the scenario presented at the figure 4 the prior estimate for the map `StandardMaterialMap` is connected with calculation of difference in type specialization levels for the factual element passed through the input link to the map and for the element type to which the material was assigned. Let us remind that the map `UserMaterialMap` will act as a selector of the included `StandardMaterialMap` instances with criteria based on computing the estimates for all the potentially applicable map instances included to it. In fact, for each particular element it selects the visual property most appropriate by its type. The selective map `UserMaterialMap` participates in the external selective map `MaterialMap`. This map combines the described above selection routine and the map `CADMaterialMap` that assigns the materials to those elements for which some of the visual properties have been already defined in some CAD application. Thus, the nested selective maps allow covering practice-meaningful use cases. It is important that the working scenario is remained valid as the application session is running and new properties are being dynamically assigned to element types by the user at runtime.

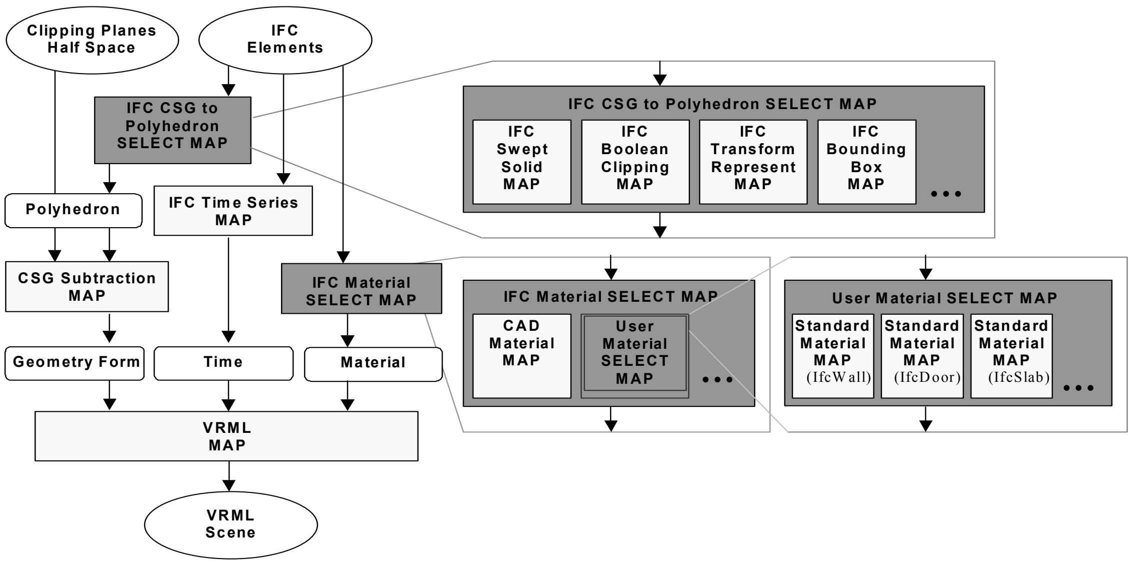

Figure 4. A virtual construction scenario for IFC-driven product data

Another problem is connected with the alternative geometry models that can be used to represent the product elements. In particular, geometry of the elements can be delivered both as constructive solids and as boundary representation. Sometimes several geometry representations can be supplied simultaneously. It means that some strategy in transforming and selecting the models should be applied. As in the previous case, often such strategy has to be defined by the user and has to be changeable at runtime.

For example, the IFC2x elements are geometrically represented as bounding boxes, as swept solids, as clipped solids, as boundary representations or as one of the enumerated models with additionally imposed geometric transformations. Several alternative representations may exist simultaneously to allow the applications and the users to employ the model which is most appropriate for particular purposes. In the presented above scenario the selective map `IfcCSGtoPolyhedronMap` performs such functionality by evaluating one of the base maps `IfcSweptSolidMap`, `IfcBooleanClippingMap`, `IfcTransformedRepresentationMap`, `IfcBoundingBoxMap` that correspond to different ways and detailization levels representing the appropriate geometry models. In such realization the user can manage the necessary level of detailization in the scenario outputs.

So, the map `IfcSweptSolidMap` extracts a swept solid model from IFC data, reconstructs extrusion and profile curves represented by canonical geometry forms and generates a target polyhedral representation. Intent of the map `IfcBooleanClippingMap` is in subtraction of a half-space solid from the IFC solid represented by one of the alternative forms considered above. It is important to note that the dependent maps `IfcCSGtoPolyhedronMap` and `IfcBooleanClippingMap` use each other to transform the recursively defined IFC geometry model to a canonical polyhedral form allowing explicit visualization. By similar way the map `IfcTransformedRepresentationMap` is realized. The only distinction is that the map applies standard 3D geometry transforms over IFC geometry given by one of the considered alternative forms. Thus, the processing maps can refer each other to proceed with sophisticated recursively defined geometry representations.

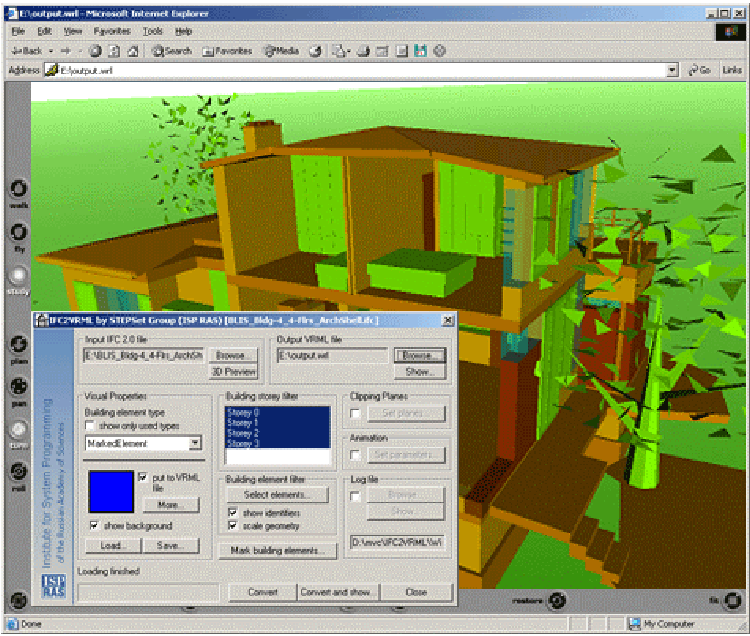

Figure 5. An example of the generated image using a virtual construction application and a scenario

Let us return to the virtual construction scenario presented at the figure 4. The scenario contains also the map `CSGSubtractionMap` to cut off the visualized buildings to provide better views on separate details of the project usually hidden by nested architectural forms and elements. In the scenario clipping planes and half-spaces are optionally defined by the user. The figure 5 presents an example of usage of this technique. The scenario is capable to animate the virtual construction process by using proper IFC project data. For this purpose the map `IfcTimeSeriesMap` analyses the sequence of actions assumed by the building data and assigns time intervals during that the corresponding elements have to be constructed. The map `VRMLMap` assembles geometry forms, materials, and time intervals delivered by the scenario maps and generates output results in the representation of static or dynamic VRML scenes.

The scenario presented at the figure 6 is intended to control construction processes at building site using the preliminary prepared IFC-driven project data and comparing them at runtime with factually conducted construction works. The progress analysis on building sites can be performed by means of optical surveying with usage of photo-grammetry or laser scanning technologies. We suggest that video shooting equipment can be suited for such purposes.

The map `SelectingMap` extracts from the IFC data those elements occurence of which is expected in the building site at the given time interval, and the elements have been already constructed and detected at the previous time intervals. The scene formed from the extracted elements is then rendered by the map `RenderingMap`. The result of the map execution is a generated 2D image and a scene derived by projection of the 3D scene objects into camera view plane. The observed image frame is preprocessed by the map `ImageProcessingMap` to recongize and to remove temporal objects, to eleminate or to reduce camera distortions and lighting artifacts, and to extract features. At this step the geometry of the surveyed building can be partially reconstructed too. The prepared 2D image and attendant scene primitives are compared with the rendered image and the derived primitives corresponding to the planned construction tasks. The detected distictions are identified by the map `ComparingMap` and a list of the detected violations is generated.

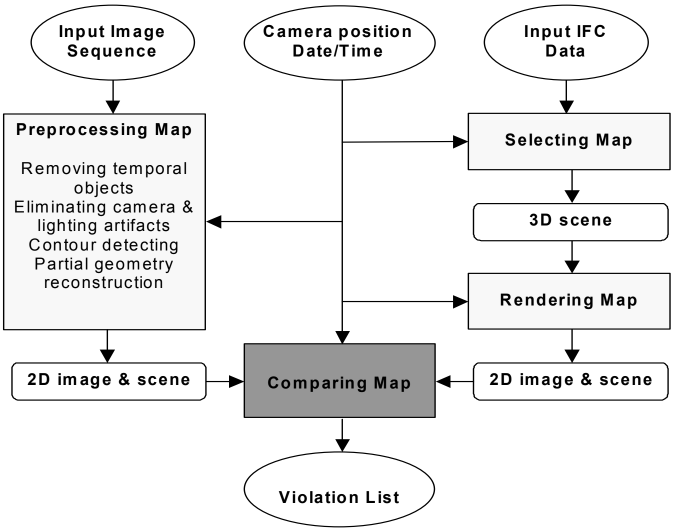

Figure 6. A scenario to control construction processes at the building site

The figure 7 shows movie frame series demonstrating a construction process at the building site and the violations detected using the presented scenario and the application.

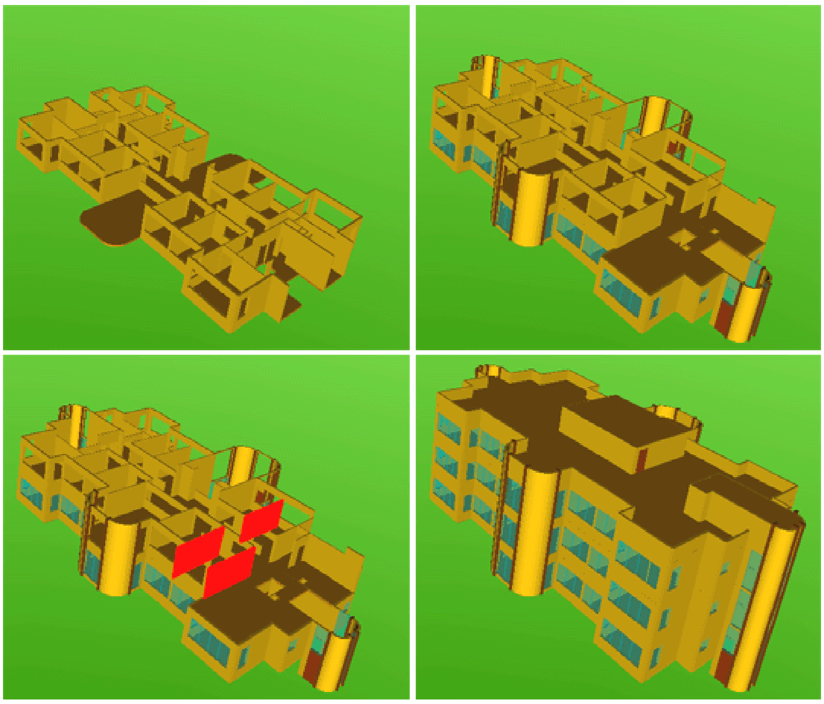

Figure 7. An example of movie frame series demonstrating virtual construction process

Due to automatization these solutions enable to increase effectiveness of construction management and to improve quality of performed construction works. The capability to output the building site scenes in VRML97 representation allows modeling construction processes and their analysing in virtual reality environments, probably, from remote places. In this case the building project stakeholders can effectively collaborate, studying the detected drawbacks and obstacles occurred during realization of the construction project.

Thus, the map-based scenarios and the application can be effectively used to solve various virtual construction problems.

5 Conclusions

The research conducted has shown that the map-based approach can be successfully used to build virtual prototyping applications upon which strong requirements on flexibility, reliability and reusability are imposed. In particular, its usage can be fruitful in conformity to applications proceeding with complex multi-disciplinary information and employing sophisticated user-defined working scenarios to be adapted at runtime.

The approach allows effective realization using the developed general-purpose framework. Support for the emerging STEP and VRML/X3D standards makes this solution a natural choice for the software applications targeted on prototyping of industrial products in virtual reality modeling environments and at the Web.

The presented virtual construction application and accompanied scenarios proof the approach consistency and feasibility as well as make for its promoting into AEC/FM industry practice. The application has a prototype system status and its demo version can be free downloaded from the web site http://www.ispras.ru/~step.

6 References

Breu, R., Hinkel, U., Hofman, C., Klein, C., Paech, B., Rumpe, B., Thurner, V. (1997). Towards a Formalization of the Unified Modeling Language. In ECOOP'97 — 11th European Conference on Object-Oriented Programming, Proceedings, eds. M.Aksit and S.Matsuoka, LNCS 1241, Springer-Verlag, 1997, pp.344-366.

Felger, W., Fruhauf, M., Gobel, M., Gnatz, R., Hofmann, G.R. (1994). Towards a Reference Model for Scientific Visualization Systems. In Proceedings of Eurographics Workshop on Visualization in Scientific Computing, eds. M. Grave, Y. Le Lous, W.T. Hewitt, Springer-Verlag, Wien, NewYork, 1994, pp. 63-74.

IAI, (1999). Industry Foundation Classes, Release 2, The International Alliance for Interoperability, 1999. <http://www.iaiweb.lbl.gov>

ISO, (1994). ISO 10303: 1994, Industrial automation systems and integration — Product data representation and exchange.

ISO, (1997). The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997. <http://www.web3d.org/x3d/specifications/vrml/ISO_IEC_14772-All/index.html>