A PRODUCT MODEL OF A ROAD

Danijel Rebolj
Faculty of Civil Engineering, University of Maribor, Slovenia
rebolj@uni-mb.si

1. INTRODUCTON

The product model approach to CAD/CAE has been recognized as the probably most efficient way to complex integration of processes in the life cycle of a product in many different engineering fields. Implementations of product models have been most successful in mechanical engineering where first standards for model descriptions have been introduced and then further developed to become international standards (see 1 and 2). Because of the uniqueness of products the development of product models in civil engineering has not been as fast. Assembly constructions were among first types of building structures for which product models could be defined efficiently and were the existing standards could be adopted in a simple way. Superstructures were the following candidate, since their components could be described quite clearly (see for example 3). Some less complex attempts to standardize the description of structures in CAD have been made as well (see 4). However, very few attempts have been made in the field of roads. We noticed a model, developed in Nederlands by TNO Building and construction research, which has very clear structure and can be used as a perfect studying object. On the other hand, some software producers integrated many phases in the life cycle of the road within their own software environments. There are also noticeable attempts to define a standard of a road product model in the industry (e.g. in IAI - Industrie Allianz für Interoperabilität, Arbeitskreis Verkehrsplanung). In general, the development of product models can be separated in two streams:

- Defining a product model from scratch is a typical approach in academic institutions and research groups, where there are no great problems in breaking off with the past concepts, models and software. In this way, new methods, techniques and tools can be tested without having troubles with "outdated" software that is mostly seen as a ballast. The often recognized drawback in this case is the incompatibility with existing "standard" procedures and software.
- Synthesizing a product model from existing in-house software modules is typical for commercial software developers. It has the advantage that the model is supported by software already in use. However, many compromises are necessary to preserve the fundamental functionality of integrated components. The most common problem of such development is the lack of flexibility and compatibility with other environments.

In our research projects, we are trying to follow the very narrow path between the new standards, methods and tools in the information technology (IT) and the engineering practice with its standard software. In the following paragraphs we will describe our approach to integrate existing "islands of automation" in the engineering practice in an evolutionary way. However, let us first define the observed object more precisely.

2. THE LIFE CYCLE OF A ROAD

A road may have a long life, comparing to other building objects. Ways, were people are traveling or transporting goods, are not moving in years or even hundreds of years. (This fact every road designer has to keep in mind, as well as those, who are designing software support for them.) On the other hand, many aspects are relevant in the road life cycle, whereby the main focus is changing from one to another - from political to engineering and financial, from engineering to managerial, etc. The following main phases are significant for the engineering aspect:

- design, which includes: analysis of traffic and, if necessary, conception of the new road, determination of possible corridors, collection of necessary data about corridors, selection of corridors with respect to collected data and defined criteria, conceptual project of the road in numerous variants, variance analysis based on different preliminarily determined aspects and selection of the most suitable variant,
- preparatory activities, such as: obtaining sites, selecting the suitable technology, determining activities and setting the time schedule,
- construction, during which execution of activities is being supervised, recorded and adjusted to the plan (or the plan is adjusted respectively),
- maintenance, which means controlling the state of the road and performing and recording maintenance activities.

For a life cycle of a civil structure, especially road, it is characteristic that individual tasks are performed in different environments, at different places. This fact greatly adds to the known difficulties occurring when individual phases are interconnected into a uniform information flow:

- partly automated processes only (isles of automation),
- data transfer between contractors in the "paper" form, which induces errors and delays,

- digital data transfer on a low abstract level (text, graphical primitives), which induces errors in interpretation and documentation,
- variability of communication modes (post, fax, telephone, computer networks), which greatly hinders the application of modern methods for integrating individual program modules into a uniform information system,
- simultaneous execution of activities, which hinders data integrity.

Due to the mentioned reasons, the extremely distributed execution of activities should be carefully considered if we are to create an information system for the control of several phases in the life cycle of a road.

3. THE ROAD PRODUCT MODEL

Since the beginning of 90's we have intensively tried to find an effective solution for the integration of processes in road design. First attempts were based on a special method for integrating different types (generations) of software modules into an object oriented environment - the object shell method (see 5). The object shell method, however, didn't prove to be effective in distributed and inconsistent computer environments. On the other hand, many tools had been (and would have to be) designed and implemented to support this method (see 6 and 7). It is probably a better (and more productive) idea to use promising new methods and tools which tend to become (or already are) IT industry standards (like Active X for software components and STEP for product modeling). On the other side the organization of processing partial tasks in the road life cycle in different environments should be considered very carefully.

Introducing a brand new information system with a totally new database concept and revolutionary new user interface without any link to any software already in use will surely cause negative reactions in every user community. Therefore "perfect models" as well as "perfect software", which is in most cases a result of development "from scratch", will hardly come to life in the real life cycle chain.

Since we gain some experiences in the already mentioned "perfect model" approach, we have begun to try the "evolutionary approach". The basis is still an object-oriented structure, whereby we followed two basic guidelines:

- links with the existing relevant "standard" software have to be supported in a very direct way (otherwise no one would be able or motivated to build complex interfaces of any kind) and

- efficient "industry" standard methods and tools (e.g. SQL and relational DBS) should be used for the implementation of the model instead of introducing new methods based on primary research

3.1 MODEL ARCHITECTURE

To preserve the most possible compatibility with the standard software in use, the structure of the model is based on the *road geometry*, derived from the conventional road design. The horizontal and vertical axis are described with a sequence of basic elements in a parametric form. Crossections are described in form of lists of points (crossection polylines) and lists of crossection elements with pointers to points in the lists. The list of element types is intended for conversion of elements in different road design systems (RDS). The main parts of the model structure is shown in Figure 1 - the details, however, can be recognized in the external representation of the model (see Figure 2).

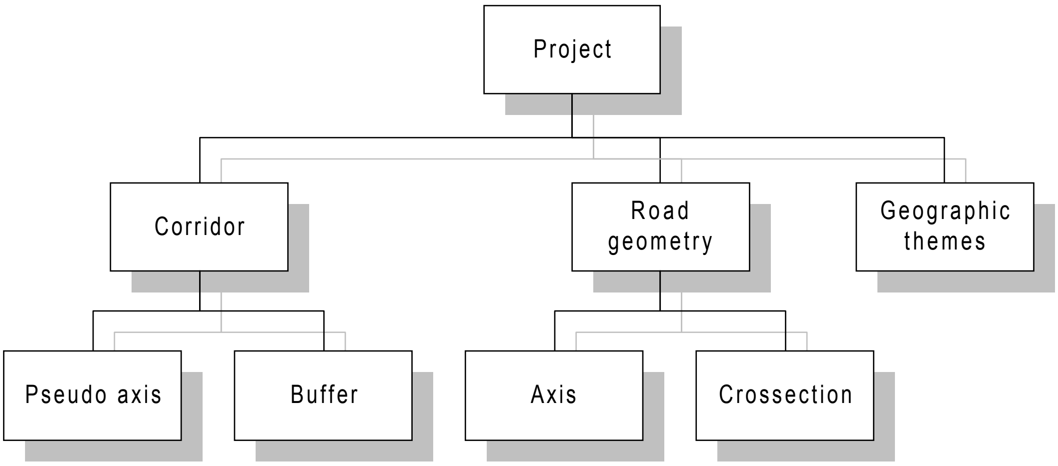

Figure 1. Main class structure in the road product model.

The *corridor* is a simple structure that defines the possible area where the road might be located on the earth's surface. It is important in the first stage of the road life cycle, where the most suitable road corridor and the first approximation of the road axis are defined. The corridor is then used in the stage of defining the road geometry (road design) since it restricts the design area. After the road geometry is defined in detail it only represents the outer limits of the "area of interest".

The road is part of the geography, therefore the corridor and the road geometry can be seen as a part of a geographic information system (GIS) or as its thematic components. On the other side, other spatial data in form of geographic themes (or layers) are needed in many tasks in the life cycle of a road. For this reason, a link to *geographic themes* is provided in the model.

The binding element in the structure of the model is the *project*, which includes general information about the road project as well as the key attribute to other parts of the structure. The model is defined in a way which permits adding and/or changing of sections without affecting other parts of the structure.

3.2 EXTERNAL REPRESENTATION

For the integration of existing programs involved in the life cycle of a road it is necessary to define an external representation of the model. The first step in our approach was a simple metafile, called MCT (*metadatoteka cestnega telesa* or road body metafile). We are still implementing interfaces to read and/or write MCT from and to program packages of all sorts, used in many different bureaus and companies involved in the road life cycle. It might seem to be more productive to describe the model using STEP, it would on the other hand take much more effort to equip all those used programs with a STEP interface.

MCT is a text file with a hierarchical structure, whereby components are described in form of sections, subsections, etc. Different sections always begin with a section name, enclosed in square brackets, and ends with a new section or with the end of file (see Figure 2). The metafile may exist with all or with only few sections - depending on the state of the project and on special wishes of the user.

[]	beginning of the section
↱	repetition is allowed
X,Y,Z	world co-ordinates
X', Y'	local (crossection) co-ordinates
italic	variables (words in parenthesis denote variable types)

```
     [PROJECT]
     'comments
         [NAME]
             ProjectName(text)
         [DESCRIPTION]
             ProjectDescription(text)
         [SCHEDULE]
             Start_date(date)  Actual_date(date)
         [WINDOW]
             Y_min  Y_max  X_min  X_max
     [THEMES]
↱        Name  Path  Type
     [CORRIDOR]
         [PSEUDOAXIS]
↱            X  Y
         [BUFFER]
↱            X  Y
     [AXIS]
         [HORIZONTAL]
↱            X_start  Y_start  X_end  Y_end Radius  A  Length
         [VERTICAL]
             [TANGENTS]
↱                L_start  Z_start  L_end  Z_end
             [ROUNDINGS]
↱                L_start  Z_start  L_end  Z_end  Radius  Length
     [CROSSECTIONS]
↱        [POSITION]
             Number  L  X  Y  Z
↱        [TERRAIN]
   ↱         X'  Y'
↱        [ELEMENTS]
             [POINTS]
   ↱             Id  X'  Y'
             [CONNECTIONS]
   ↱             Id_startpoint  Id_endpoint  Element_Type(text)
     [CROSSECTION ELEMENT TYPES]
↱        Element_Type(text)  Class(text)  Description(text)
     (eof)
```

Figure 2. Definition of the road metafile MCT.

Since the model is GIS supported, any geographic theme that covers the relevant area might be seen as a part of the external representation as well. The distribution of explicitly or implicitly linked geographic themes is, however, left to more or less standard spatial databases and their network distribution or import/export facilities.

4. THE ROAD LIFE CYCLE ENVIRONMENT

RO (ρ), as we named our integrated software environment for the road life cycle support, has passed many development phases and was implemented with many different methods and tools (see 5, 6 and 7). In its present evolutionary state, it would be best described as an component-oriented client-server information system with a flexible and modular structure. The concept of the system is shown on Figure 3. RO workstations should use the project database, wherever possible. In other cases, the external representation of the model in a form of MCT metafile can be used for data transfer. MCT is also used to integrate workstation with specific software equipped with a MCT interface.

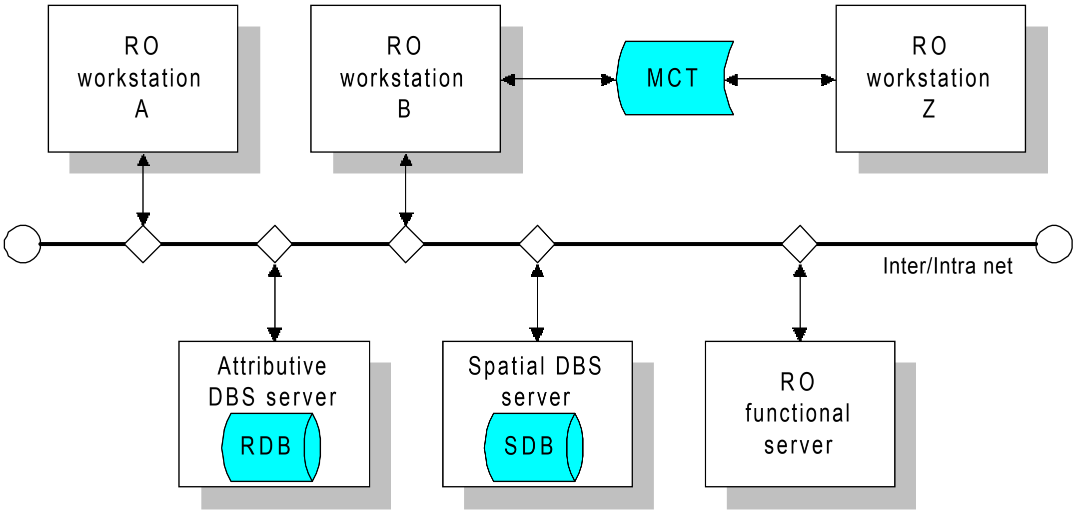

Figure 3. Operational scheme.

In designing and implementation we followed the same concepts of using "industry" standard IT methods and tools as for the model design.

4.1 DESIGN AND ARCHITECTURE

The system's architecture is based on the road model. The main objects and methods are listed in the following table:

Object	Method	Description
Project	Define	define a new project
	Open	make an existing project current
	Edit	change unbound project's data
	List all	list all projects in the DB
	MCT out	write the desired parts of the current project into a metafile
	MCT in	read desired sections from a metafile into the current project
Corridor	Define	define a new corridor (boundary and/or pseudo axis)
	Edit	change the geometry of the corridor
Road geometry	Design	link to the chosen RDS program
	Analyze (generic)	links to procedures to analyze the road geometry from different aspects
Theme	Add	add a relevant geographic theme to the GIS layers list
	Remove	remove a theme from the list
	List all	show all themes in a project

Objects and methods are distributed in three different parts of the system:

- the main objects definitions reside in the *RO kernel* with links to the systems database and to external methods, executed by functional servers,
- the attributive and spatial datastructures are defined in the *system database* which consists of two main parts: the attributive part (a relational database system) and the geographical part (a spatial database system),
- most of the methods are separated from the system into independent modules - *functional servers*, which enables a high degree of flexibility and simple upgrading of the system functionality.

The interface to the external representation of the model (MCT) is a part of the RO kernel. Functional servers have a direct connection to the RO database (Figure 4).

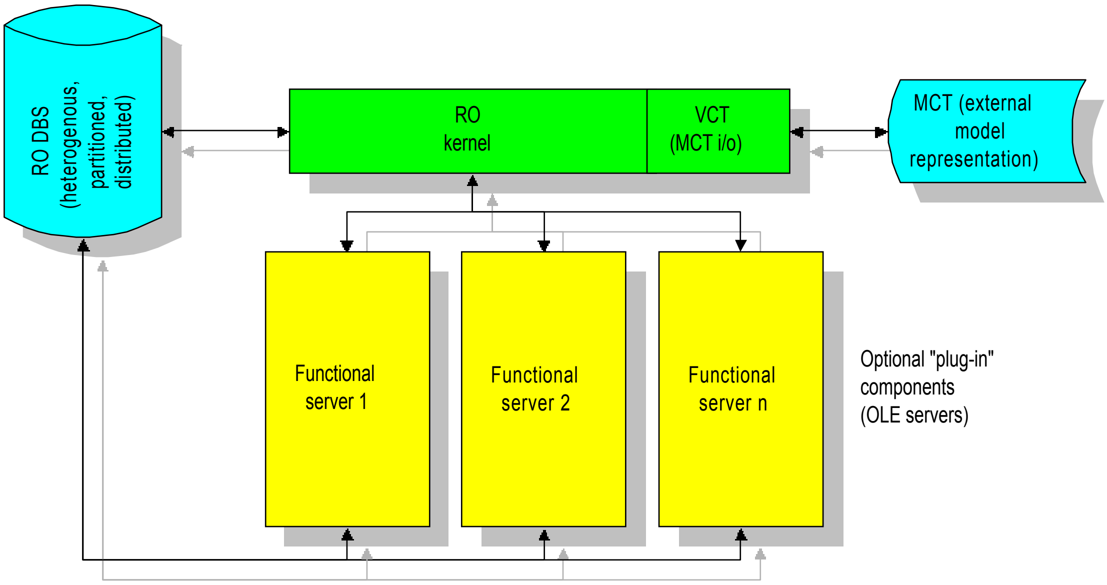

Figure 4. RO architecture.

4.2 COMPONENTS

The open component-oriented architecture of the RO environment enables adding and/or changing of methods without affecting other parts of the system. In this way we believe to assure covering of tasks in the life cycle of the road with new functional servers not only by our group but also by others. At present, we built some fundamental methods as well as some special ones that were needed by specific users:

Corridor definition is usually invoked in the earliest phase of the road life cycle. The basis for this method are geographic themes which may be chosen from the spatial database. Depending on the applied criteria we might choose themes like landuse, wood, biothops, road network, isolines, lot cadastre, water, etc. Following the given criteria we then define a pseudo axis between places A and B that the new road should connect. In the last step we define the corridor boundary - either manually (as a polygon that avoids any critical object in the environment) or automatically (as a buffer that surrounds the pseudo axis at a specified distance). A more sophisticated method of corridor definition will be developed in the future.

3D fast visualization is usable to judge the road geometry after or even between steps in the process of road geometry design. The method uses the crossection descriptions of the road model to build a three dimensional graphical model of the road (for details see 8). It's most important advantage is that it doesn't need any additional data or manual procedures to visualize the road and can therefore be used at any stage of the road design, assuming that the needed data is imported from a road design system (RDS) into the RO database. Since some RDS already include 3D visualization, our visualization method seems redundant in cases when such programs are integrated with the RO environment. There is, however, another application of the visualization module. On RO workstations where other tasks in the road life cycle (after the road design) are being processed and where there is no need for any RDS software, an interactive visualization of the road can give a better impression of the road layout.

Emission calculation module was designed and implemented as a prototype of an analytical functional server. Since very detailed geometry of the road is available in the RO model, we had the opportunity to apply a precise algorithm of emission calculation dynamically - which means instant use of data, changing in the road design process (for more information about Dynamic Emission Model see 9). Results are shown as geographical layers and may be combined with other geographical themes.

Land acquisition module is a result of a research project requested by the Ministry of transportation of the Republic of Slovenia. From the very beginning it was anticipated as a part of RO environment, since this is the only way to truly enhance the process of land acquisition - one of the most time-consuming task in the pre-construction phase of the road life cycle. The method is based on overlaying the digital parcel of land cadastre with the road body. Affected parcels of land are then processed according to the regulation whereby changes in any procedure are saved into the database. The relevant attributes can be used for rendering the (geo)graphical image of the cadastre map.

4.3 IMPLEMENTATION

The current version of the RO system is intended to run under Windows 95 and Windows NT (except servers, which may run under other OS as well). RO was developed using a relational database system and RAD technology with a strong support of available software components. The attributive part of the model was implemented using MS Access. This

RDB system is sufficient for the requirements of the most environs where the RO system (or parts of it) are being used. However, any other RDBS might be used instead.

The RO kernel was implemented using MS Visual basic. VB supports MS Access format directly, it supports development of OLE servers which may be used in local or remote mode, and the use of many commercially available software components (OCX, ActiveX or whatever they are called at the moment). Some components were implemented using Visual C++ because of a much higher speed of executive code.

For all GIS related functions, including management of the spatial database, a boundle of ActiveX controls was applied. In the current version we used MapObjects, developed by ESRI (Environmental System Research Institute, Inc). Through these controls the system gains direct access to all ESRI formats of geographical themes as well as a direct support of ESRI's SDE (Spatial Database Engine).

Fast 3D visualization was implemented using another sophisticated component - 3d Graphic Tools by Micro System Options. The road body is generated from the available crossection description using mesh elements. The visualization is then conducted interactively using cameras, viewpoints, lights and other available elements. Except the mentioned components, many other more or less standard ones were used.

5. CONCLUSION

We believe that our approach to the integration and support of different parts of the life cycle of the road will be successful. At present there are only a few related companies that are willing to cooperate and of course the institutions which build and manage roads. However, these few users of the RO environment have been convinced about synergetic effects of the process integration. High abstraction of the data and simplification of data transfer, reduced need for data transformation, error reduction, process acceleration and the increasing number of simply upgradable functional modules are only some of them. We expect that with the growth of the RO user community the positive effects will grow even more. The only problem seems to be, how to convince the critical mass of potential users. On the other hand, we will try to further develop the model according to STEP - to be compatible with new generations of commercial software that slowly shows up.

REFERENCES

1. *STEP, Standard for exchange of product data*, ISO CD 10303 -41, -42, -43, 1993.
2. *Product data representation and exchange*, ISO TC184/SC4/WG3 N434 (T12), 1995.
3. Bakkeren, W.J.C., "Integrating Structural Design and Analysis through Product Modeling", *Proceedings of CIB W78 Workshop on Computer Integrated Construction,* Helsinki, Finland, 1994.
4. *Technical product documentation - Organization and naming of layers for CAD*, ISO/DIS 13567, 1996.
5. Rebolj, D., *Computerunterstützter integrierter Straßenentwurf in einer objekt-orientierten Umgebung*, Verlag für die Technische Universität Graz, Graz, 1993.
6. Rebolj, D., "Integrated road design and evaluation environment", *Proceedings of Computing in Civil and Building Engineering.* Berlin, 1995, pp. 1001-1006.
7. Rebolj, D. and Tibaut, A., "Integrated information system supporting distributed tasks in the life cycle of a road", *CIB Proceedings of Construction on the information highway,* Bled, 1996, pp. 423-434.
8. Tibaut, A. and Rebolj, D., "Visualisierung des Straßenkörpermodells", *Proceedings of Internationales Kolloquium über Anwendungen der Informatik und der Mathematik in Architektur und Bauwesen,* Weimar, 1994, pp. 151-156.
9. Rebolj, D., "Simulation of harmful emissions caused by road traffic with the use of dynamic emission model", *Proceedings of Power Engineering,* Maribor, 1995, pp. C/277-284.