

PROTOTYPING A SENSOR ENABLED 3D CITYMODEL ON GEOSPATIAL MANAGED OBJECTS

E. Kjems^{a,*}, J. Kolář^b

^a Aalborg University, Centre for 3D GeoInformation, 9000 Aalborg, Denmark - kjems@3dgi.dk

^b Copenhagen, Denmark - jan.kolar@grifinor.net

KEY WORDS: Geospatial Managed Objects, InfraWorld, Interface, Sensor, Virtual Machine, Data Fusion, Geoinformation

ABSTRACT:

One of the major development efforts within the GI Science domain are pointing at sensor based information and the usage of real time information coming from geographic referenced features in general. At the same time 3D City models are mostly justified as being objects for visualization purposes rather than constituting the foundation of a geographic data representation of the world. The combination of 3D city models and real time information based systems though can provide a whole new setup for data fusion within an urban environment and provide time critical information preserving our limited resources in the most sustainable way. Using 3D models with consistent object definitions give us the possibility to avoid troublesome abstractions of reality, and design even complex urban systems fusing information from various sources of data. These systems are difficult to design with the traditional software development approach based on major software packages and traditional data exchange. The data stream is varying from urban domain to urban domain and from system to system why it is almost impossible to design a complete system taking care of all thinkable instances now and in the future within one constraint software design complex. On several occasions we have been advocating for a new end advanced formulation of real world features using the concept of Geospatial Managed Objects (GMO). This paper presents the outcome of the InfraWorld project, a 4 million Euro project financed primarily by the Norwegian Research Council where the concept of GMO's have been applied in various situations on various running platforms of an urban system. The paper will be focusing on user experiences and interfaces rather than core technical and developmental issues. The project was primarily focusing on prototyping rather than realistic implementations although the results concerning applicability are quite clear.

1. INTRODUCTION

1.1 Background

Since the inauguration of the Centre for 3D GeoInformation (3DGI) at Aalborg University in 2001 one of the main goals in the centre have been to develop and present a sustainable and feature rich platform to represent a 3D model of the world. After initial years and several cycles of research prototyping, a novel concept of handling object-oriented representations of geographic features on the server-side in the exactly same form as on the client has emerged. The key reasoning behind this concept based on the first prototype was addressed in (Kolář, 2006). This meant a paradigm shift in the core of the design and development. While traditional software packages have predefined input and output data structures, the new design was based on contents compiled to byte-code and executed directly by a virtual machine (VM). This means every geographic feature may, if convenient, run as a little program. Initially to run graphically on a computer screen but equally importantly also running things in the background. These objects can be enabled for instance to change position, or change form, communicate with other objects, or with real objects connecting the virtual model to the real world. This approach has advantages compared to traditional GIS software and is very suitable for 3D environments and the sensor city concept ("Senseable citylab - MIT," n.d.). The content objects were eventually called Geospatial Managed Objects (GMO). Running code in a VM as a middle layer, is not new and was already introduced in 1966 (Richards, 1969; Rohl & Feldman, 1967). Today the VM concept is known through Java Virtual Machine,

Microsoft's .Net, or Adobe Flash. Java Virtual Machine is used in our development due to its availability on most system platforms. (Bodum et al., 2005; Kjems, Bodum, & Kolar, 2009; Kolar, 2006).

1.2 InfraWorld

Getting funding for a new GI system, build more or less from scratch and on new ideas, is not easy. Some claim our approach has been tried before without any success, and others don't believe in features build as objects using byte-code. Anyway, together with the companies Iver from Spain, Vianova Systems Finland Oy, Norkart and Vianova Systems from Norway we applied and got the grant from the Norwegian Research Council. About 4.0M Euro for a new virtual world approach aimed at infrastructure data hence the project title InfraWorld. Prior to the InfraWorld project the GMO concept was used on a couple of smaller projects, which clearly showed potential, but it was not possible to go any further with the concept at that state due to the lack of a working platform. (Kjems & Bodum, 2009).

The main motivation for the commercial partners was the ability to get more out of the data produced during the design and construction phases of an infrastructure project. Steps in this direction can also be observed in the building and construction area where for instance the IFC standard (Industrial Foundation Classes) has moved the whole sector from using simple spatial geometry to a rich object oriented representation of the building. Creating virtual city models (VCM) is quite costly although prices have dropped tremendously over the past decade, and

* Corresponding author.

even though the list of different applications using these city models is remarkable (Batty et al., 2000; Kolbe & Gröger, 2003) the real use seen as share of produced city models is rather modest. The lack of usability and costs of city models can quickly make the originally enthusiasm fade away. Probably therefore the growths rate of virtual city models is slowly but steadily declining (Morton, Horne, Dalton, & Thompson, 2012).

The InfraWorld project took its origin in a planning → design → construction → phase organized situation. The idea was to be able to not only reuse data produced during the phases, but also to let the data be part of future designs, for maintenance, and other purposes. In that case the phase diagram gets a product life circle shape shown in Figure 1.

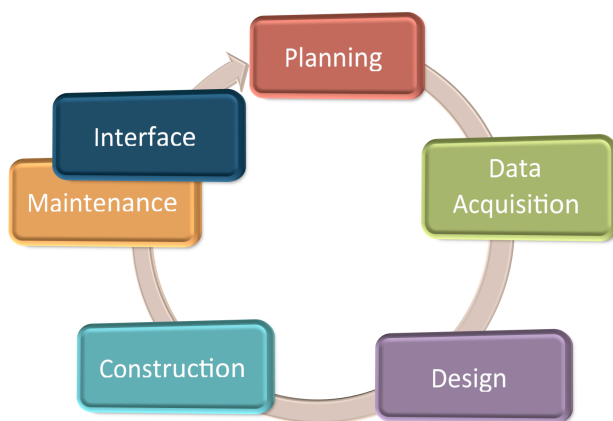


Figure 1. Feature data life cycle

Apart from the perhaps obvious circle shaped diagram, which indicates the reuse and maintenance of data in general, this figure also shows a new phase that lies close to the maintenance and is called interface. Interface implies interaction which has not yet been part of such construction data but is possible, when these data are transformed to GMO's. As a GMO feature data can be included in a monitoring situation connecting the virtual model to a real construction site whether it's a building, a bridge, a road or any other construction. Sensor data are easily monitored this way, and in principle only one model is necessary to cover large areas. Figure 1 might not be entirely correct anymore since the data acquisition phase in a computer environment is more or less included in all phases, although the main data still are acquired in the beginning of a project.

GMO's can be used widely for many different purposes, when it comes to representing real features in a virtual model. In the InfraWorld project we had to decide what kind of functionality of GMO's we wanted to show within the time frame given. During the preliminary discussions of the project, which had a duration of almost the whole first project year, it got clear that a GMO approach would mean a completely new paradigm and data model not used within geospatial data anywhere else, but it also gave us the opportunity to create and test a platform that can handle the sensor city concept in a cross disciplinary and a sustainable way. Therefore the test scenarios chosen in the project had a broader perspective towards sensor city and were aiming at a platform development rather than an application limited for infrastructure data. The test scenarios are presented here in short:

1. Cross platform interaction and platform independency

One of the strongest arguments for using GMO's is not only the independency of using different platforms, but also the possibility of fusing different technological solutions together into one coherent platform. In InfraWorld the development comprised an interaction directed from three different existing clients. One based on gvSIG (<http://www.gvsig.org/>) a 2D GIS system, which ran on Linux provided by Iver, another based on Virtual Map (<http://www.vianovasystems.com/>) a 3D model viewer running on Microsoft Windows provided by Vianova in Finland and finally Virtual Globe (<http://www.virtual-globe.info/>) a 3D globe viewer which also is running using Java Virtual Machine, in this case on Apple OSX and provided by Norkart in Norway.

One thing is that these three different clients were loading the same model, but they did also access the same feature database the model was based on. That way an alteration of the model in one client triggered the same alteration in the other clients. A GMO added in one client almost immediately showed up in the other clients if they had the same geographic area of the model loaded. By the way, no exchange standards, or conversions were needed for that.

2. Features handled in space and time

To show how GMO's can be handled when it comes to space and time, it was decided to monitor moving features allowing their object representation to move around in the virtual model according to their counterpart in real life. Such objects could be trains, ships, busses, aeroplanes etc. But we chose taxis because there were a lot of taxis in the test area, and the data were freely available. With regard to possible sensor activities these mobile objects easily could contain important for instance environmental sensors which then would be part of the GMO definition representing the features. In our case there are no such additional data.

3. Collision Detection

When several domains are working on the same project with each their focus on either water, lightning, sewer etc. using different kinds of data and specialized software, the only thing in common is the geography as long as the same coordinate system is used, which unfortunately sometimes is not the case. Many systems can handle geometric collision detection but since we are using GMO's we extended the collision detection to be rule based. This way features could collide when for instance a minimum distance rule was not obeyed.

4. Energy monitoring system

Even though this part of the project was a bit off topic when it comes to infrastructure data, the content is very much up to date and proofing the concept in general using sensors in the model connected to reality. The idea here was to monitor the buildings in the model with regards to their energy consumption. They got coloured according to their consumption compared with the size of the building and use of sustainable energy. Finally a possibility was developed to add solar collectors and panels on to the building showing up in the model and improving the overall energy consumption. This kind of sensor activity is very relevant and very much on the energy provider's wish list, because a tight monitoring of the energy consumption enables the companies to plan the energy flow from production to customer in a better and more beneficial way.

2. THE CORE PLATFORM R&D

2.1 Virtual Machine Selection

The GMO concept, as described in (Kolář, 2013), has been implemented as a software library named Geospatial Reference Interface For Internet Networks (GRIFIN). In order for the GRIFIN implementation to be applicable and robust few already existing technologies have been applied. The most crucial in this regard was the selection of the VM technology. Note that the use of VM in our approach is not a mere software engineering convenience for implementation or porting to different operating systems. The role of the VM's byte-code relatively to GMOs is similar to the importance of HTML relatively to Web pages. The consequence of a future change to a different VM would have an analogy in changing HTML to, let's say, PDF format - making all the previous content obsolete and non-functional. Hence requirements on the VM technology are relatively strict and include: non-proprietary solution, strong focus on backward compatibility, production quality with commercial leadership, and widely established availability. Given these priorities the HotSpot VM, which is the original VM used within the Java ecosystem, stands out as a nearly unchallenged choice, despite its currently marginal availability on mobile platforms.

The GMO concept has been coded in form of an abstract class, which provides access to the common representation of geographic space with time and scale to an indexing mechanism associated with the space, and to custom functionality needed by each individual GMO definition. Hence it is guaranteed that all implementing subclasses and their GMO instances have these three properties. Concrete examples of GMO definitions, which include references to the source code, are available at: <http://grifinor.net/examples>.



Figure 2. Identical GMO's displayed in three different clients

2.2 The main features of the platform

In order to facilitate prototyping and the re-use of GMO models most of the examples utilize the Scala language and the GRIFIN Shell (GShell), which allows for an interactive use. GShell can be used to create, manage and consume the geospatial content on the GRIFIN platform. It has all management, server, and remote access features available from a uniform environment and provides a way to exchange and execute code on GRIFINs distributed network. This follows the original vision of a space for collaboration on model development, and not just a one-way publishing medium for static, predefined, and hard-to-change

types of geospatial information. Identical GMOs displayed in three different clients.

Figure 2 depicts the Strømsø city model and daily energy consumption per building modelled as GMOs. While the model is relatively complex the software clients only implement the API for handling GMOs, which accounts circa twenty methods. The city model itself has several times more methods. Three main practical properties, which might be hard to address using non-GMO solutions, are:

1. The model brings all its functionality to the different clients,
2. The model specification and definition may undertake big changes while the API for GMO could be implemented independently in parallel,
3. Once the GMO API is done the software client is ready for all future currently non-existent models using the GMO method.

GRIFIN is an experimental library that additionally implements several practical features not addressed in this article including mechanisms for storage and retrieval of GMOs, exchange of GMOs over network, automatic 2d interpretation of 3d geometries, support for visualization, and API for client applications to use GMOs. The experiment is maintained at <http://grifinor.net>, and is provided under the GNU General Public License.

It is not possible to compare this solution to any other because each GMO is a little executable held together in the GRIFIN framework. No one else has chosen this approach why for instance the data structure, data exchange and interaction is very different to other systems which may in general look like a similar solution but in reality is far away.

3. INFRAWORLD EXAMPLES

3.1 Platform Independency and Cross Platform Interaction

When one designs a platform, where collaboration and correlation of information is connected to several different domains, in this case engineering domains, one certainly must design its solution as open as possible enabling interdisciplinary working environments. This was one of the main reasons why we chose the Java Virtual Machine (JVM) as executable layer over for instance a .Net solution, which would have constrained us to the Microsoft platform. JVM has working solutions for many different platforms and appurtenant operating systems among them Microsoft's most common Windows operating systems, Apples OSX and several GNU/Linux distributions. All three platforms were used and tested under different conditions in our test environment, see Figure 2.

Even though the whole idea is based on JVM, and that choice by itself should secure a platform independency it is not a trivial setup. The consortium of the InfraWorld project consisted of not only different companies with different applications but also with different work and development environments, which made it natural rather than intentional to test the GMO concept cross platform. Iver from Spain were developing on a 2D GIS platform using the open source environment gvSIG programming in Java. Iver used different platforms but was here running on a GNU/Linux platform. Vianova from Finland were using Novapoint Virtual Map, their own development, and a quite advanced viewer for 3D infrastructure models developed

in C++, which also was their preferred programming environment. Vianova only deploy their developments for Microsoft Windows. Since GMO's are build using Java a special technique encapsulating GMO's within the C++ environment was used by the Finns. Finally Norwegian Norkart involved themselves using a client for visualising virtual worlds handling terrain and aerial images together in a need way. Norkart was developing in Java also using Java Virtual Machine. This platform also was used for the main development running on an Apple/OSX based machine. These three very different environments were able to run JVM proofing the versatility of the concept of using JVM. Even though the environments were very different not only from a machine/OS point of view but also from an application point of view. GMO's created and executed in each of these combinations gave expected results, and even though for instance gvSIG was a 2D environment, the results were usable. The scenarios presented in the following paragraphs were all using this setup and carried out during a life presentation with about 100 attendees. A video footage of the event can be watched following this reference (Hoel, 2012).

3.2 Features handled in space and time

A GMO is compared to for instance a traditional GIS not dependent on fixed coordinates in the data model. This means that features are allowed to move around due to behaviour build-in in the GMO. Also the aspect of time can be handled very flexible and be part of the GMO. To show this kind of GMO behaviour in InfraWorld, the choice fell on modelling taxis letting them move around in the model according to their real position. With an online link to a taxi company in the Oslo area, it is possible to retrieve data from each of the cars and their momentary position. In the model the taxis are shown as long poles to make them clearly visible in the example, see Figure 3. Clicking on one of these poles gives in our case an id and the possibility to follow the car and activate an animation of the latest movement of the car. A lot of other information could be relevant, but the ones chosen here are just indicating a possible interaction and remote information. The clustering of poles in the figure is caused by the railway station, which happens to be at that place.

3.3 Collision Detection

This example goes beyond the classical collision detection, which is controlling whether geometry is colliding unintentionally due to the boundary representations of the modelled features intersecting. Especially within CAD software used for construction, and since the implementation of the IFC standard, this kind of controls are rather mainstream. Also our Finish partner in the InfraWorld project had geometric collision detection with regards to infrastructure features implemented in their software package. The collision detection here is rule based. The idea is to let semantically defined rules be part of the collision detection. This means rules can be described in the GMO and obeyed. In our simple case we incorporated a distance rule between respectively a light pole and a manhole thereby coming from two different domains. They can easily conflict placed on the same spot during the design phase. When GMO's are loaded into the model and executed they check on this rule an act accordingly. In our case conflicting GMO's change colour to orange and enable the possibility to resolve the problem, which in this case means moving apart obeying the rule of distance to neighbouring objects. This situation is shown in the following illustrations.



Figure 3. Taxis showing as poles.

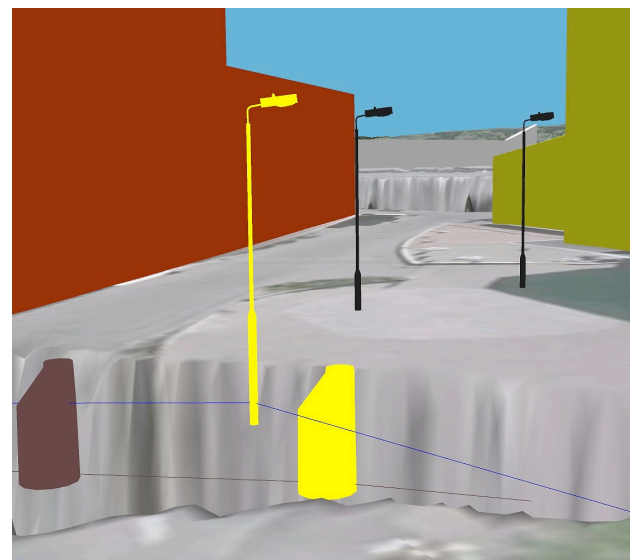


Figure 4. Two selected GMO's - light pole and manhole

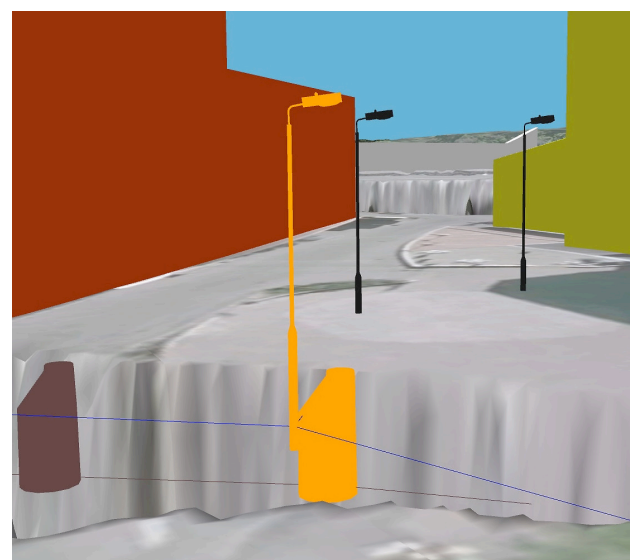


Figure 5. Light pole moves too close to manhole – the colour changes to orange

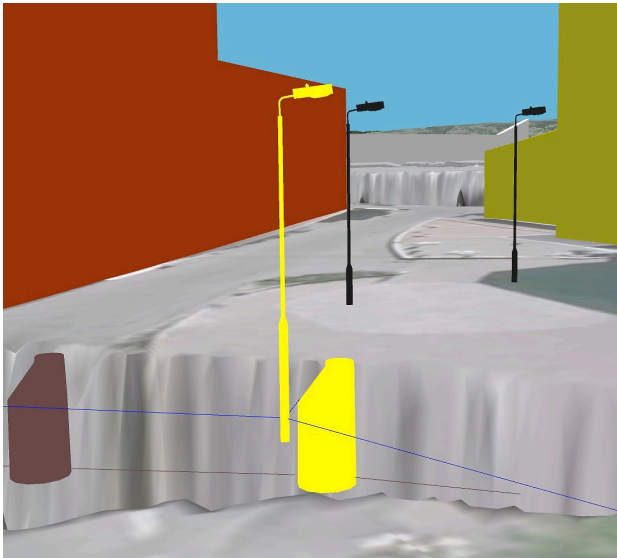


Figure 6. Light pole moves to minimum required distance

Figure 4 shows the light pole and the manhole in question. Both are selected (yellow) and in this case the GMO's are activated. The light pole is now moved to a different position but collides with the distance rule of a manhole. Both the light pole and manhole turns orange, see Figure 5. A resolve button for this special issue has been developed and can be activated. The light pole moves a bit back obeying the distance rule of the manhole, see Figure 6. One can argue that it not necessarily should be the light pole moving but the manhole, and certainly there are a lot of questions arising from this kind of functionality. Since we did not have this kind of autonomous behaviour available in objects within construction before, the numbers of possibilities accompanied with questions and challenges is overwhelming. How do one handle for instance semantic rules, and how do one assure the right decisions found among the objects. Can parts of the collaboration, and agreements found among engineers coming from different domains, be replaced by autonomous objects? How do one organize behaviour as part of objects? I.e. who is enhancing them from dumb geometry entities to autonomous agents, and how can it be assured they actually can communicate and understand each other. The idea is of course to have a system, which can help avoiding design errors, and saving money, being ahead of the situation, and not creating additional problems. A Ph.D. project elaborating on some of these issues mentioned here has been carried out as part of the InfraWorld project. Since GMO's have agent like behaviour the Ph.D. project looked at the possibility of negotiations between agents and also self-learning functionality. (J. D. Faus & Grimaldo, 2012; J. Faus, Grimaldo, & Barber, 2012).

3.4 Energy monitoring system

This example was mainly developed to show the potential of the GMO concept used for monitoring features, which can react due to sensors attached or can change due to other external influences. In this case selected buildings in the model are monitored with regards to their energy consumption. With comparison to the online taxi scenario, it unfortunately was not possible to retrieve the energy consumption data with an online setup due to legal concerns. The energy consumption was represented by a dataset of sampled data during one year. In the model one can click on a building as one can do with all GMO's in the model to evoke a window, if such is relevant to the object, which allows one to interact with the GMO, see Figure

7. Buildings in Figure 7 appear with different colours indicating the energy consumption level going from dark green indicating a relatively low consumption to dark red indicating a relatively large consumption. In the window one can observe the energy consumption due to a specific day of year. Since we have sampled data for the year of 2010, these dates are available in the model. Another feature build in, is the possibility to apply solar panels and solar collectors respectively producing electricity and heating water. Choosing a certain number of collectors or panels will respectively imply an amount of panels and collectors placed on the roof of the building, and accordingly carry out a recalculation of the overall energy consumption, see Figure 8. Here 40 solar panels have been applied and a yearly saving of 10.400 NOK is indicated (blue text). Although the saving is evident the building is not changing colour yet since the saving still is not significant to provoke any changes. The solar panels are now fixed to the building, and other clients retrieving the model from the database will likewise see the panels showing up as well, as being able to interact with the model changing the amount further. Of course this kind of possible client access needs a strict organization around the design.

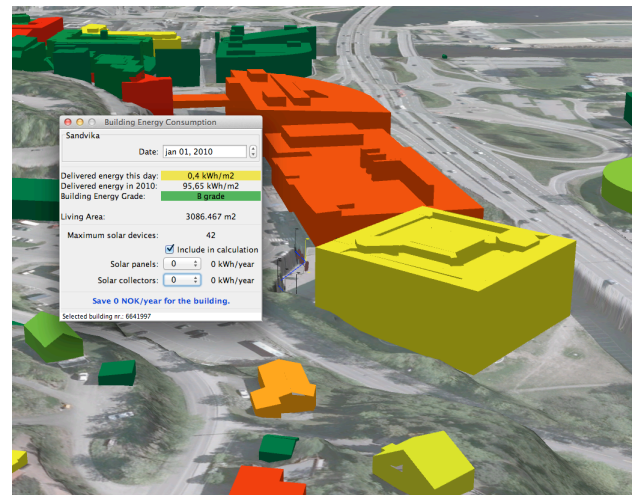


Figure 7. Energy consumption for selected building

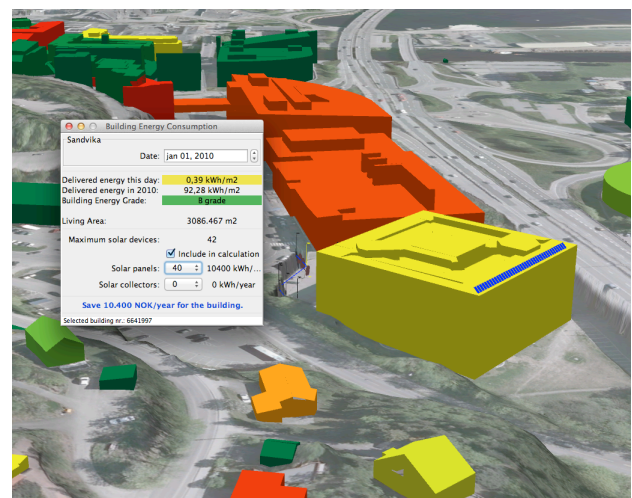


Figure 8. Selected building (yellow) with 40 solar panels placed on the roof (appears here as a blue line)

During the mentioned presentation (Hoel, 2012), this was done by all three clients in turn thereby showing the connection to a life model, which could be altered freely by the connected clients. The alteration in the model happened automatically without interference by the passive users. As a curiosity it should be mentioned, that the model server was running in a different country in a home environment. It should be clear by now that a similar effect could be obtained moving around with manholes and light poles. Although this example perhaps makes it even more clear what effect such a system could have in a working environment during the design phase, since all involved partners/clients immediately can see changes carried out by other partners in a project.

4. CONCLUDING REMARKS AND FUTURE WORK

The InfraWorld research project had its origin in the challenge of increasing the usability of design data produced during a design and construction phase within the cross disciplinary fields of an infrastructure project. The solution, if one chooses to call it that, turned out to be much more and something else than an application, another standard or database. It turned out to be a completely new approach of handling a virtual world, and its features. A platform that can face future challenges within the sensing city, where information from many places, coming from many different technological solutions, working within unsynchronised timeframes, being able to not only handling features in time, and space, but connecting the information they produce in real time, and being able to let them co-exist on one single platform. On one hand one gets a user interface, which intuitively lets one interact with the city. On the other hand and perhaps most importantly the platform allows one to fuse the data or information produced within the city. There is really no limitation on what kind of data that can be handled. A lot of data today are collected separately within each domain, usually based on technological islands, and not prepared for possible data exchange with open systems. But in the near future it is crucial that the information flow is transparent and connected, and organized in a way that decision support systems have access to a broad range of data within the city eco system, and can help to optimize our spare resources in fast growing environments. A human being will not be able to react in time or will easily make mistakes, due to the huge amount of data influencing the decisions. The development carried out in the InfraWorld project has clearly shown a way to face these challenges by using the GMO technology.

5. REFERENCES

- Batty, M., Chapman, D., Evans, S., Haklay, M., Kueppers, S., Shiode, N., Smith, A., et al. (2000). Visualizing the city: communicating urban design to planners and decision-makers.
- Bodum, L., Kjems, E., Kolar, J., Ilsøe, P. M., Overby, J., Oosterom, P. van, Zlatanova, S., et al. (2005). GRIFINOR: Integrated Object-Oriented Solution for Navigating Real-time 3D Virtual Environments. Berlin: Springer Verlag.
- Faus, J. D., & Grimaldo, F. (2012). Infraworld, a multi-agent based framework to assist in civil infrastructure collaborative design. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3* (pp. 1499–1500).
- Faus, J., Grimaldo, F., & Barber, F. (2012). Multiagent System for Detecting and Solving Design-Time Conflicts in Civil Infrastructure. In J. M. C. Rodríguez, J. B. Pérez, P. Golinska, S. Giroux, & R. Corchuelo (Eds.), *Trends in Practical Applications of Agents and Multiagent Systems SE - 7* (Vol. 157, pp. 57–64). Springer Berlin Heidelberg. doi:10.1007/978-3-642-28795-4_7
- Hoel, J. E. (2012). InfraWorld – Avslutningsseminar. *Vianova Systems, InfraWorld project*. Retrieved from <http://www.vianovasystems.no/Brukermoe/ter-2012/InfraWorld#.UbcqTPbfxc>.
- Kjems, E., & Bodum, L. (2009). Object Oriented Visualization of Urban Energy Consumption. In Yeh Anthony G. O. ; Feng Zhang (Ed.), *11th International Conference on Computers in Urban Planning and Urban Management (CUPUM)*.
- Kjems, E., Bodum, L., & Kolar, J. (2009). Managed Objects for Infrastructure Data. In J. Lee & S. Zlatanova (Eds.), *3D Geo-Information Sciences* (pp. 97–107). Springer Verlag.
- Kolar, J. (2006). On The Road To 3D Geographic Systems: Important Aspects of Global Model-Mapping Technology. *Innovations in 3D Geo Information Systems* (pp. 207–223). Springer.
- Kolář, J. (2006). Interoperability and 3D Content in GRIFINOR. *3d Workshop at 25th Urban Data Management Symposium*. Aalborg.
- Kolář, J. (2013). Bytecode Unification of Geospatial Computable Models. In *FOSS4G 2013 Proceedings. Vol. 11*. Nottingham. Retrieved from <http://grifinor.net/g/publications/pdf/DRAFT1/viewer.html>
- Kolbe, T. H., & Gröger, G. (2003). Towards unified 3D city models. *Challenges in Geospatial Analysis, Integration and Visualization II. Proc. of Joint ISPRS Workshop, Stuttgart*.
- Morton, P. J., Horne, M., Dalton, R., & Thompson, E. M. (2012). Virtual City Models: Avoidance of Obsolescence. *Education and research in Computer Aided Architectural Design in Europe-eCAADe, Prague, Czech Republic*.
- Richards, M. (1969). BCPL: a tool for compiler writing and system programming. *Proceedings of the May 14-16, 1969, spring joint computer conference* (pp. 557–566). New York, NY, USA: ACM. doi:10.1145/1476793.1476880
- Rohl, J. S., & Feldman, J. A. (1967). A formal semantics for computer languages and its application in a compiler-compiler CACM Vol. 9 No 1 pp3-9 January 1966 9 N WIRTH and H WEBBER. *Proceedings of... National Conference* (Vol. 3, p. 385).
- Senseable citylab - MIT. (n.d.). Retrieved from <http://senseable.mit.edu/>