

DYNAMIC FEATURES IN A 3D CITY MODEL AS AN ENERGY SYSTEM

W. Wen, E. Kjems, L. Bodum, J. Kolar

Centre for 3D GeoInformation, Department of Development and Planning, Aalborg University, 9220 Aalborg, Denmark - (wanw, kjems, lbo)@3dgi.dk, kolda@grifinor.net

Commission IV, WG IV/8

KEY WORDS: City, Modelling, Three-dimensional, Visualization, Dynamic, Representation, GIS

ABSTRACT:

A dynamic feature refers to an individual feature that can involve the change of population counts, identities, thematic attributes, spatial, or temporal characteristics etc. A 3D visualization of individual dynamic features provides an intuitive and vivid way of conveying information, and it is also an important milestone for achieving a complete spatiotemporal city model, therefore a city model that supports the 3D representation and visualization of dynamic features is of general interest. The solution for 3D interactive dynamic features presented in this paper was developed within the project EnergyCity. This project is situated in the city of Frederikshavn in the north part of Denmark. The goal for the project was to make a 3D city model act as an awareness tool allowing politicians as well as citizens to visualize and understand the change of energy consumption and energy sources in an urban environment over a period of time. By interacting with those dynamic features, users can visually and immediately realize how the change of energy consumptions matters, which helps them to realize a more sustainable while feasible energy scenario. The method called managed object is applied to represent 3D dynamic features in EnergyCity. A managed object refers to a pure object-oriented and platform-independent binary representation that carries both the executable behaviours and attribute data regarding an object.

1. INTRODUCTION

Following the ideas and thoughts of Digital Earth (Gore, 1998), many different solutions for Digital Globes have emerged within the last 10 years (Butler, 2006). They have brought a wide range of geo-related applications for mass-market through the Internet (Goodchild, 2008). In the same period there has been a rising of awareness when it comes to issues such as global warming and climate change. From a geovisualization perspective several applications in relation to the growing awareness have been developed showing results of e.g. the models for warming up the globe and the sea level rising (Buki, 2007; Google, 2008). These visualizations of global consequences plays an important role in understanding of the eco-systems and how the behaviour of people in general interfere with climate and the effects on the environment. Following this trend, this paper takes one step further in the development of three-dimensional (3D) city models as energy system.

This paper exhibits a project called EnergyCity developed by the Centre for 3D GeoInformation at Aalborg University. In this project, a 3D city model is enhanced into an energy system that allows politicians as well as citizens to visualize and understand the change of energy consumption and energy sources in an urban environment over a period of time. This enhancement of 3D city model is based on a combination of a purely numerical based model named EnergyPLAN (Lund and Münster, 2003, Lund and Østergaard, 2008) and an object-oriented (OO) geovisualization platform called GRIFIN (Bodum et al., 2005; Kjems et al., 2009). Though solutions proposed by energy planners are well elaborated by EnergyPLAN based on numeric calculations, it is difficult for non-experts to understand. However, the success of achieving a self-sufficient energy city is very much dependent on the cooperation of the industry, politicians and not least the citizens, who should act with the knowledge presented by energy planners and change their life

styles in terms of energy consumptions. Therefore, an enhanced 3D city model is suitable for better conveying information from energy planners to the public.

EnergyCity is supposed to help users to aware a more sustainable and feasible energy scenario by allowing the interaction between users and dynamic features that stand for various energy consumption or energy sources. For this purpose, the dynamic features in our project play a significant role for bridging numerical calculations and 3D visualizations (Figure 1). For example, when the users increase the population of wind turbines, more wind turbines are immediately shown up in the 3D city model. Meanwhile, the changed parameters are submitted to the underlying EnergyPLAN and the results are presented as graphs after few seconds.

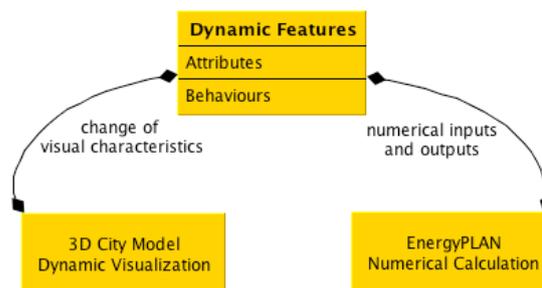


Figure 1. The role of dynamic features: The numbers held by attributes or behaviours of dynamic features are some of the inputs or outputs of EnergyPLAN, which executes numerical calculations. In the 3D city model, the visual effects of those dynamic features are also dynamic according to the change of their attributes.

The approach called Managed Object (MO) is firstly applied and implemented to represent 3D city features. A managed

object refers to a pure object-oriented and platform-independent binary representation that carries both the executable behaviours and attribute data regarding an object. The idea of representing features as managed objects has been researched and mentioned before (Kjems et al., 2009). The platform used in the project is called Geographic Reference Interface For Internet Networks (GRIFIN), which is a digital earth platform developed in Java language with strong focuses on interoperability and visualization.

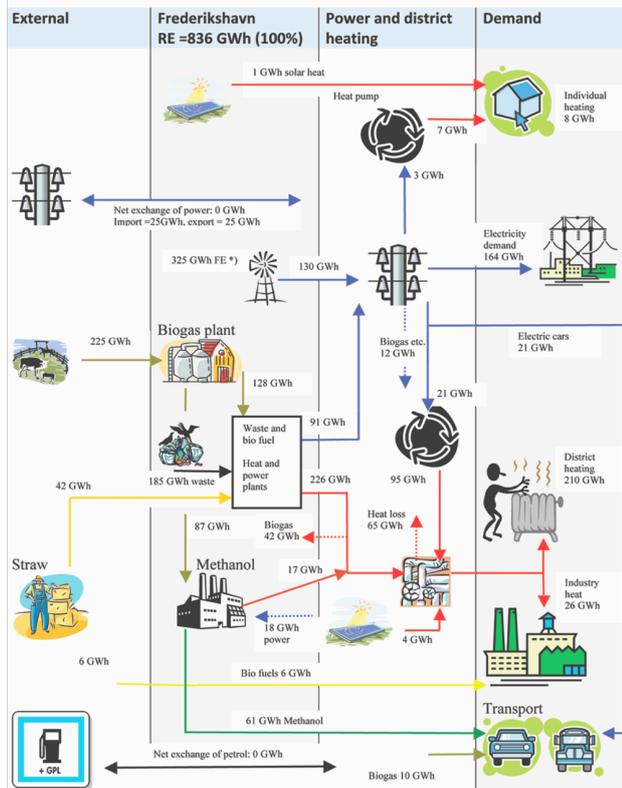


Figure 2. Proposals from energy planners: This diagram proposed by energy planners shows how the optimized energy system could look for a self-sufficient energy city (The selection site is the city of Frederikshavn in Denmark). It is important to understand that the suggested solutions are also depends on the efforts from both the politicians and citizens. From (Lund and Østergaard, 2008)

Two main parts are elaborated regarding the representation and visualization of changeable features. The first part (Section 2) explains our attempts to define the dynamic features as managed objects. This includes firstly a general-purpose pattern for representing geospatial features, especially for those with complex visualization requirements. Further, the mechanism of implementing dynamic visualization and interaction of geospatial features in GRIFIN is also included. The second part (Section 3) exhibits our experience on geo-communications in the EnergyCity project.

2. DYNAMIC FEATURES IN 3D CITY MODELS

2.1 Representing Dynamic Geospatial Features as Managed Objects

Representation has a profound impact on defining and limiting what can be represented in an information system and how the information is represented (M. Yuan et al. 2004). An efficient representation of features is essential for naturally representing

the ontology applied in both the geospatial and application (e.g. energy, transport, etc.) dimensions, and therefore it is vital for enhancing the usage of 3D city models from static visualization to dynamic and semantic-enhanced applications. Conceptually, the location, time, geometry and appearance, semantics, and individual behaviours are all important aspects of geospatial 3D features. However, in most conventional 3D city models, those aspects are often separately handled in practice, especially regarding the behaviours, which are often implemented as software functions than part of the feature representation. This section aims to introduce a MO based method, which aims to fulfil a comprehensive representation of geographic features by allowing an integration of different characteristics, 3D visual effects, and functions regarding individual features.

A MO is created by pre-translating (compiling) a piece of source code, which holds the description of a certain feature in terms of both its attributes and executable behaviours to an intermediate representation known as byte-code. Since the byte-code is designed to be executed by a virtual machine (Gough, 2005) rather than by any dedicated software, operating system or hardware, all the managed objects are like interoperable black boxes in the environment of a certain virtual machine. Consequently, contents inside those black boxes are freely customizable; geographic coordinates, geometries, appearance, thematic attributes, links to extra data sources, functions are all allowed to be included.

The MO method actually enables a 3D city model to apply the OO concept for content representation and data management. The usage of OO concepts has been long timely discussed in the field of GISc (Peuquet, 2001; Lohfink et al., 2007; Worboys et al., 1990; Li B. and Cai G., 2002; Borges et al., 2001). However, the actual implementation of OO concepts in the 3D geovisualization has not yet been well explored. Based on the encapsulation and inheritance mechanisms (Meyer B, 1988), the OO concept provides a flexible, extensible, and natural way for representing of individuals and their relationships. To utilize the merits of OO concept sufficiently, a proper design of data model is crucial. A data model deals with both a foundation for database implementation and a conceptual modelling of geographic reality, upon which different cognitive perspectives and applications exist. Based on general geographic information applications, Borges et al. (2001) points out ten requirements for a geographic data model, covering considerations on conceptual level, presentation level and implementation level. Based on his ten requirements, this paper extends several requirements for designing data models with a focus on the 3D applications:

- The data model should allow a general representation of geospatial features. This includes the considerations covering two aspects. Firstly, a satisfactory data model should provide the flexibility for users to model their own interesting features rather than to choose from a limited set of feature types. Secondly, the way to represent a geospatial feature differs, according to what a user expects to gain from the representation. Therefore, the ideal way to depict features should allow users interests rather than only certain conventions.

- A data model is based on the conceptual model, which embeds certain ideology of modelling the reality. To facilitate the embodiment of such ideology on a presentation and implementation level, important aspects of representation are suggested to be highlighted by the data model.

- The extensibility is important for 3D geographic information systems, because the generation and maintenance of 3D city

models is expensive. A data model is required to allow multiple and new semantics, geometry, and functions. The authors correspondingly suggest that a data model should only enforce the most essential and general properties that are common for all features rather than any ordinary properties.

Type	Description
Context information	For a world-range geographic information model that deals with different spatial extents, ontologies, and time intervals, a mechanism for sorting data and fast retrieving data of interests is significant. The idea of adopting the context information for MOs is mainly aimed to facilitate the data management and retrievals. Every MO uses context information to provide a basic reference or province of itself. In our project, (1) a reference point standing for the approximate location on earth and (2) the time interval standing for the life span of a feature are selected as the context information, which basically holds the extent of a feature in the context of space and time. Such spatiotemporal context is considered as the most essential attributes for every geospatial feature; therefore, it is the only thing enforced on all valid contents on GRIFIN, by declaring it in the abstract class, which is inherited as a template by all geospatial MOs.
Visual properties	The 3D models provide an intuitive and vivid way to present the visual properties of features, which are actually important aspects for describing characteristics of features. Practically, the visual properties mainly refer to the geometry and appearance (including texture) data that are needed for rendering and visualizing the feature. Besides visualization purposes, the geometry of a feature also records the spatial occupancy or boundary information. Depending on practical conditions, the geometry and appearance can be either stored inside MOs or linked to other data sources. For example in our project the texture image of terrain, which is of large size, is only maintained as a URL link in the visual properties of the terrain MO.
Thematic, spatial, and temporal properties	The thematic properties refer to the properties in a thematic or application dimension, such as transportation, immigration, etc. If the spatial or temporal properties in context information are not complete, more spatial or temporal information can also be maintained here.
Behaviours including also relations	The behaviours of features are also included in this list of properties, in the sense that behaviours are also regarded as an important aspect to depict the characteristics of features. Especially for the changeable features, allowing individual features to manage their own behaviours or functions makes the feature representation more complete and natural. The relationships between features can also be maintained by behaviours, besides the inheritance mechanism.

Table 1. Pattern for representing geographic features as MOs.

- Data model should also facilitate the practical implementation. In terms of 3D city models, this mainly refers to issues regarding 3D visualization, which is the most attractive, important while onerous parts of 3D projects. The data sources for 3D city models are often large and complex, involving problems like rendering speed, multi-formats, visual effects, etc.

Corresponding to those items above, Table 1 is suggested as a pattern for representing dynamic features as MOs in 3D city model. This pattern is designed with a strong focus on structuring information for depicting the characteristics of features for general purposes, rather than giving a detailed OO template for setting attributes or functions, nor only for the purpose of dynamic and visual presentation. The (1) context information, (2) spatial, temporal and thematic properties, (3) visual properties, and (4) behaviours including also relationships are highlighted as four important aspects for representing features in a 3D city model, for the sake of reducing the gaps among application requirements, conceptual design, and implementation issues. For general cases, one geographic feature is represented as one MO, which consists of data regarding one or several aspects of its corresponding feature. It is important to mention that for representing a single feature it is not necessary to complete all those four aspects. For example, our project only enforces the context information (i.e. spatial and temporal context in GRIFIN) for facilitating spatiotemporal indexing, based on the premise that every geographic feature exists with its spatiotemporal extents. Since visualization holds special significance for 3D city models, visual properties, mainly about the geometry and appearance, are separated from other properties, i.e. spatial, temporal and thematic properties. Based on the concept of MO, the spatial, temporal, thematic, and visual properties are never defined as static. Rather, via behaviours they can be interrelated and interact on each other.

2.2 Mechanism of implementing dynamic geospatial features in GRIFIN

This subsection describes the method for implementing dynamic geospatial features in GRIFIN based on our experience in the EnergyCity project. In GRIFIN, the mechanism of implementing dynamic geospatial features starts from the inheritance of the GO class, which is the super-class that passes common attributes and methods down to every MO that stands for valid geospatial content. The GO class specifies three main types of context information, which are automatically tagged by every geospatial MO. The first type is a spatial index interpreted from a three-dimensional spatial reference in a geocentric coordinate system. Since a spatial reference is a point that approximately specifies the geographic location of its associated feature, the spatial index derived from the spatial reference benefits a fast spatial query. The second type is temporal information that records the start and end points of a feature's life. The main role of temporal context information is to index objects in the temporal dimension, so the temporal query is facilitated. More detailed spatial or temporal attributes regarding a feature's life is normally held outside the context information part but by other attributes and behaviours of the MO. When the temporal dimension is not significant, the temporal context can be ignored or set as everlasting in order to simplify the representation. The third type is an object ID primarily used for direct referencing or retrieving MOs in databases or over network. Besides those three types of context information, the GO class also specifies a common function called *goMain()*, which is the key access that allows the platform to interact with MOs. Since the *goMain()* method is embedded in all the MOs, it can be directly utilized by any client (viewer, navigator) in order to access application-specific behaviours of the object. Two parameters, a selection event and a mediator for a hosting client application, are required by *goMain()* (shown in Figure 3). A selection event refers to an event object initialized by a client against a particular geospatial MO for selection purposes. For example, a click on a MO is a

simple selection event. A mediator for a hosting client application helps the client to recognize that a selection has been initiated. The mediators are designed to provide more sophisticated means of user interaction as well as hard coded functionalities that are not part of feature definitions; it is

closely bound with particular clients rather than the MOs. Smart implementations of *goMain()* can enable advanced interactions based on the type of selection event and the type of hosting application.

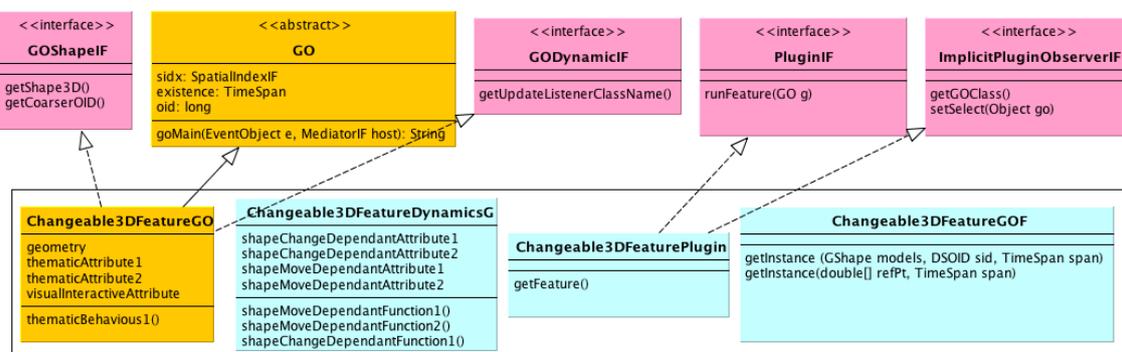


Figure 3. The design diagram: geospatial features supporting dynamic visualization, interaction and application dimensions.

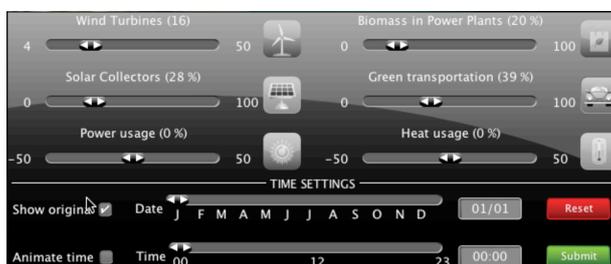


Figure 4. The graphical user interface for adjusting energy parameters: six parameters are shown above the Timeline Settings.

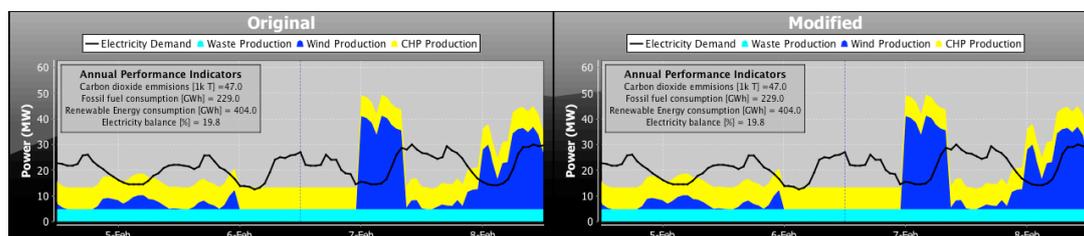


Figure 5. The graph showing energy consumptions and the balance.

Besides the inherited properties and method from *GO*, a definition of a MO class can also include thematic properties, extra spatial or temporal properties, visual properties like geometry, and other behaviours. The procedure to define a geospatial MO class is called a feature definition in this paper, which allows a feature-based customization for representing a particular type of geospatial features. In Figure 3, *Changeable3DFeatureGO* is an example of feature definition generally standing for an arbitrary dynamic 3D feature.

In order to better support the dynamic visualization and advanced interaction regarding those dynamic features, several optional components are designed in this framework in addition to the feature definitions. In this paper, we present only two components. One optional component is aimed at updating, morphing, moving or just changing features from the client. This is solved through listeners that can change the interpretation of a feature on the client side. The listeners in GRIFIN are designed to be both feature and client specific, which means that one feature definition can have multiple listeners for different clients, and one listener can also serve multiple feature types. As shown in Figure 3, a specific listener called *Changeable3DFeatureDynamicsG* is designed for a specific feature type called *Changeable3DFeatureGO*. When a

listener is specifically created for a specific feature type, many of feature's attributes and methods largely associated with the visual dynamics can be included in the definition of the listener, rather than its feature definition. Applying this component to certain feature type is via an interface called *GODynamicIF* (Figure 3). Another optional component aims at manipulations regarding the geometries of features. In 3D applications, the geometry of a feature is usually an essential while complicated property. Therefore a particular interface called *GOShapeIF* is designed to allow shape-specified features like 3D buildings to access and manipulate their geometry-associated properties. Non-shape-specified features like addresses, sounds etc., can also be visualized somehow, but their shapes for visualization are not their pre-specified essential properties. A MO class can extend several optional interfaces, which allows specific optional components to work together with the feature definition. It is suggested to keep only the essential properties and behaviours of a feature type in the feature definition part, and leave other intermediate parameters or specific functions for visualization or interaction purposes to those specific components.

When specific user interfaces and common analytical functionalities are required for all the instances of a particular

feature type, a particular plug-in can be applied. In Figure 3, a particular plug-in called *Changeable3DFeaturePlugin* is designed for feature type *Changeable3DFeatureGO*. Any particular plug-in is a concrete implementation of the *PluginIF* interface. A single feature type can apply multiple plug-ins, and a single plug-in can also suit multiple feature types. Depending on application requirements, which plug-in is used can be determined by the returned value of *goMain()* in the feature definition. If no specific plug-in is selected, a default plug-in is applied. At last, in order to facilitate the generation of feature instances, the factory class can also be designed (shown as *Changeable3DFeatureGOF*).

3. DYNAMIC FEATURES FOR COMMUNICATIONS

This section discusses our experience on geo-communications in the EnergyCity project. As what has been introduced in the first section, the main motivation of the EnergyCity project is to use 3D visualization to convey information given numerically by energy planning experts to the public and the politicians of the municipality in an intuitive way. Therefore, this project faced the challenges of presenting changes due to the users' interaction and changes of the overall energy consumption of the city of Frederikshavn. Not having numbers on the level of each individual building but only for the entire city as one unit actually increased this challenge rather than simplifying it, because it is difficult to make people understand the changes happening for a whole city by presenting individual 3D buildings, which emphasize their individual realistic 3D visual effects rather than an overall effect.

In order to make the system easy to understand, many resources are put into the development of the graphic user interface (GUI), which indicates six different energy parameters controlled by the users (Figure 4). These six parameters allow the users to set inputs for numerical calculations in EnergyPLAN, which is the numerical model designed by energy planning experts. Other parameters have been pre-determined because the increasing of interactive parameters would decrease the degree of usability. Numerical outputs from EnergyPLAN are then visualized by the 3D city model, which intuitively makes the users understand the consequences of their input settings.

To visualize the consequences corresponding to users' input settings, three different visualization ways were developed. These three ways are classified only based on their geo-communication effects, rather than their actual implementation approaches. The first way aims to make users understand the change of popularities by adding or deleting corresponding features in the 3D city model. The change in numbers of wind turbines in the model were presented by showing a number of similar wind turbines appear and disappear in the city model according to the position of the slider on GUI (Figure 6). The increase or decrease in the decomposition of biomass was presented quite similarly by showing a number of green silos appearing and disappearing according to the percentage of possible biomass use. The second way was aiming at the whole city model but still only addressing parts of it indicating individual choices on the household level. In this way solar collectors were appearing on specific roof sites. As a matter of limited resources and a fast approaching deadline the selection of potential roofs suitable for solar collectors were done rather pragmatic than correctly. The result showed a limited amount of the smaller buildings getting solar collectors attached according to the percentage settled for with the GUI slider. The solar collectors were shown by changing the colour of the roof site

pointing primarily towards south as the most plausible choice of roof site. Actually the colour change was achieved by adding a new black face right above the original roof. It is interesting to mention our experience of choosing the geometry and appearance for rendering solar collector. At the very beginning we chose 2m*2m squares with silver colours, which is close to real solar collectors, standing for individual solar collector models. However, in this way users could not gain any visual effect indicating the situation of the whole city, because those realistic squares were invisible until zooming into individual buildings. At the end, we changed the geometry as big as the whole roof, and marked it with noticeable dark colour, so that users can gain an overview of solar collectors when zooming out (Figure 7). From a visual perspective, the green transportation was presented similar though here all cars in the model were set to change colour going from white to green, indicating the shift to electric propulsion of vehicles in the city. The colour change was achieved by holding the relationship between the green car MOs and the white car MOs. Finally in the third way, the energy consumption regarding electric power and heat, which in most cities in Denmark is provided with central heating, is presented as an all-over colour change of the roof on buildings (Figure 8). For the electric power, the roofs changed from green to dark red indicating a decrease of power by 50%; while the heat usage was shown with colour changes of the walls of all buildings going from deep blue to dark red indicating a decrease and increase ranging from -50% to +50% compared with the actual use in 2007.



Figure 6. Change of wind turbine population

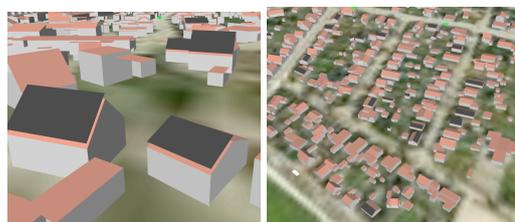


Figure 7. Solar Collectors: amount of solar collects appearing or disappearing on individual house roofs indicates the percentage of utilizing solar energy on a whole city level

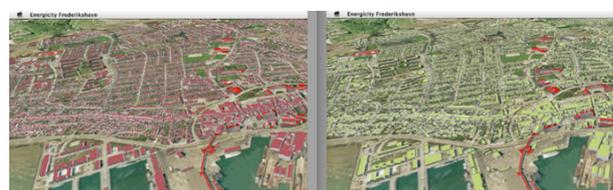


Figure 8. Changing the colour of all the roofs: an all-over colour change of the roof on all the buildings indicates the usage of electric power

The way users or audiences perceived the implemented interaction and the results shown by the city-model and the connecting graphs vary quite a lot. The system has been presented at the Bright Green Exhibition in Copenhagen in conjunction with the COP15 Climate Conference December 2009. The audience was spreading from school children to

professionals within energy production, monitoring systems and consultancy. It was obvious and not unexpected that professionals were focusing on the graphs shown in Figure 5 rather than the 3D city model. Non-professionals were playing with the sliders but not really paying attention to the serious background of the system and the underlying energy calculations. But the ones who were just a bit interested in understanding the overall energy model could easily understand the connection between the model and the graph. From model to graph, the audience could notice even small differences like the green transportation is related to an increasing or decreasing amount of CO₂ in the overall energy calculation. The audience normally still needs some explanations about the meaning of dynamic visualization. For example, a turbine running faster or slower does not really indicate whether it produces enough wind power to balance the energy consumption to the green side; but a curve lying beneath or beyond a line (Figure 5) is clearly showing the balance.

The original idea within the project was to be able to identify every household in the model and further show the exact energy consumption of each building. The inhabitants of the city could then learn about whether their individual consumption lies within an expected measure or exceeds too much. However, due to legal regulations this idea was not developed. But an assumption can still be made that the 3d city model would have had a much larger impact to the public users if the energy consumptions were shown on household level. If one should draw an objective conclusion of the effect of EnergyCity for the purpose of communicating the energy consumption of a whole city, the right word would probably be “overkill” though a nice one.

4. CONCLUSIONS

This paper elaborates our solution and considerations of representing and visualizing dynamic features in a 3D city model, which is enhanced to act as an energy system that allows the public to understand the energy consumptions and sources of the city. The key methodology for implementing dynamic features is the Managed Objects (MO). Based on the concept of MO, an object representation that supports the context information, visual properties, thematic properties, spatiotemporal properties and behaviours of a feature is presented. Besides, a mechanism of utilizing MOs for dynamic visualization and interaction in our system is also elaborated. The presented solution should be considered as an open paradigm for representing complex geospatial 3D features associated with interactions and dynamic visualizations, rather than a concrete solution that specifies any template for assigning attributes or functions. Based on this paradigm, more advanced dynamic or interactive components can be added for better results. Future work can apply and refine the suggested method across different possible applications, such as transportation, forests, etc. There should also be a systematic evaluation of the suggested method by examining the possible changeable mechanisms held by various dynamic features. From a geo-communication perspective, our experience on making the 3D city model as an information-conveying tool is presented and analyzed as the result. Though we believe that the effect would be better if the visualization is on a household level rather than an overall city level, we still gained certain experience on developing 3D city models for showing large-scale information. Compared to the 2D visualization, the 3D visualization is often more realistic-oriented by natural, which to some degree limits the visualization approaches that represent

thematic information; however, based on the case of developing solar collectors, we found out that a realistic depiction of changeable features does not always work effectively, and sometimes even fails to convey correct information when the range is not suitable. The setting of colours in an overall city range is also tricky; the colour indicating a meaning can disturb the appearance of the model and make other colour indicators less usable.

REFERENCES

- Bodum, L., Kjems, E., Kolar, J., Ilsøe, P.M. and Overby, J., 2005. GRIFINOR: Integrated Object-Oriented Solution for Navigating Real-time 3D Virtual Environments, In: P.v. Oosterom, S. Zlatanova and E.M. Fendel (Editors), *Geo-information for Disaster Management*, pp. 937-949.
- Borges K, Davis C, Laender, 2001. A. OMT-G: An object-oriented data model for geographic applications. *GeoInformatica*. 2001;5(3), pp. 221-260.
- Buki, Z., 2007. Google Earth Community: Rising Sea Level Animation v2.0
- Butler, D., 2006. Virtual globes: The web-wide world. *Nature*, 439(7078), pp. 776-778.
- Craglia, M. et al., 2008. Next-Generation Digital Earth. *International Journal of Spatial Data Infrastructure Research*, 3, pp. 146-167.
- Google, 2008. Earth Outreach KML: Climate Change In Our World, Google Earth Outreach - Showcase.
- Gough, J., 2005. Virtual Machines, Managed Code and Component Technology. *Software Engineering Conference, 2005. Proceedings*. 2005 Australian, pp. 5-12.
- Li B, Cai G, 2002, A General Object-Oriented Spatial Temporal Data Model. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(4). pp. 100-105.
- Lohfink A, Carnduff T, Thomas N, Ware M, 2007. An object-oriented approach to the representation of spatiotemporal geographic features. *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems - GIS '07*. 2007:1.
- Lund, H. and Münster, E., 2003. Modelling of energy systems with a high percentage of CHP and wind power. *Renewable Energy*, 28(14): 2179-2193.
- Lund, H. and Østergaard, P.A., 2008. EnergiBy Frederikshavn - Scenarieplanen, *Energy City Frederikshavn secretariate, Frederikshavn*.
- M. Yuan, D. Mark, M. Egenhofer, and D. Peuquet. 2004. Extensions to Geographic Representation. *Research Challenges in Geographic Information Science*, pp. 129–156.
- Meyer B, 1988. Object-Oriented Software Construction. London: Prentice-Hall International; 1988:534.
- Peuquet D, 2001. Making space for time: Issues in space-time data representation. *GeoInformatica*. 2001; 5(1), pp. 11-32.
- Worboys, M. F. and H. M. Hearnshaw, 1990. Object-oriented data modeling for spatial databases. *International Journal of Geographical Information System*, 4(4), pp. 369-383.