

AN INTEGRATED WEB-BASED 3D MODELING AND VISUALIZATION PLATFORM TO SUPPORT SUSTAINABLE CITIES

Sam Amirebrahimi*, Abbas Rajabifard

Center for SDIs and Land Administration (CSDILA), Department of Infrastructure Engineering
The University of Melbourne, Victoria 3010, Australia – sami@pgrad.unimelb.edu.au and abbas.r@unimelb.edu.au

Commission IV, WG IV/5

KEYWORDS: Sustainable Development, 3D Platforms, Visualization, Model, Web based, Decision Support

ABSTRACT:

Sustainable Development is found as the key solution to preserve the sustainability of cities in oppose to ongoing population growth and its negative impacts. This is complex and requires a holistic and multidisciplinary decision making. Variety of stakeholders with different backgrounds also needs to be considered and involved. Numerous web-based modeling and visualization tools have been designed and developed to support this process. There have been some success stories; however, majority failed to bring a comprehensive platform to support different aspects of sustainable development. In this work, in the context of SDI and Land Administration, CSDILA Platform – a 3D visualization and modeling platform – was proposed which can be used to model and visualize different dimensions to facilitate the achievement of sustainability, in particular, in urban context. The methodology involved the design of a generic framework for development of an analytical and visualization tool over the web. CSDILA Platform was then implemented via number of technologies based on the guidelines provided by the framework. The platform has a modular structure and uses Service-Oriented Architecture (SOA). It is capable of managing spatial objects in a 4D data store and can flexibly incorporate a variety of developed models using the platform's API. Development scenarios can be modeled and tested using the analysis and modeling component in the platform and the results are visualized in seamless 3D environment. The platform was further tested using number of scenarios and showed promising results and potentials to serve a wider need. In this paper, the design process of the generic framework, the implementation of CSDILA Platform and technologies used, and also findings and future research directions will be presented and discussed.

1. Introduction

Over half of the world population is now living in cities (UNFPA. 2007) and the urban population growth is predicted to continue for the next hundred years. Sustainable development has been identified as a solution to environmental and social problems caused by this growth (Brundtland & Khalid 1987). Achieving this in urban development however is complex and requires holistic and multidisciplinary decisions across social, economic, and environmental domains (Giddings et al. 2002), by range of stakeholders with variety of backgrounds and interests. This becomes challenging especially in cases of cities which are vertically extended both above and below ground level. To support sustainable development in this environment, having this in mind that sustainability can be measured in urban development context (Isaacs et al. 2011), a large number of web-based platforms using various technologies has been developed by both practitioners and academics to facilitate better decision making and improve public private partnerships in the context of sustainable development. However, a majority of these tools/platforms lack uptake, and are too generic, or too specific around a particular field (Isaacs, et al. 2011). They are classified into three general groups. The first group includes tools such as Giovanni 3 by NASA (Berrick 2009) with extensible and powerful modeling capacity but inadequate visualization power. In contrast, there exist platforms such as K2Vi (AAM 2012), virtual globes such as Google Earth and CityGML-based visualization platforms like the work by Elwannas (2011) and Altmaeier and Kolbe (2003) which concentrate on improvement of visualization techniques to convey 3D information about the world. Despite the flexibility of this group in manipulation of objects in the 3D environment and presentation of the scene to the user, these tools generally lack underlying modeling capabilities to complete their

comprehensive visualization. Lastly, there are platforms such as S-City VT (Isaacs, et al. 2011) and those tools developed to model and visualize earthquakes and flood mapping with both effective modeling and visualization capabilities. These platforms however, are mostly expert-oriented applications with a focus to specific applications and limited target users and as a result, there may be difficulties in data access, extensibility in integrating additional models to the system, and an insufficient level of trust towards the result since the used modeling algorithm may not be familiar and sufficient to the user (black-box effect) (Isaacs et al. 2008).

To address the limitations of existing tools and approaches, this paper aims to introduce a generic framework for development of web-based modeling and 3D visualization platform to support urban sustainable development. This comprehensive framework can be used for further developments in the research community. This paper describes the development of a seamless and integrated web-based 3D prototype in the Centre for SDIs and Land Administration in the School of Engineering in The University of Melbourne. “*CSDILA Platform*” supports multi-disciplinary modeling of sustainability; and allows the integration of user-defined models to the system with effective 3D visualization in a user-friendly environment. In this way, sustainability of a development can be assessed in an integrated modeling environment and provides a tool to facilitate the decision making process.

The remainder of this paper is structured as follow: Section 2 presents the development of the platform. This section first describes the investigation and extraction of technical and non-technical requirements as well as the process of development of the generic conceptual framework. Later, the implementation of the prototype system (CSDILA platform), its architecture, and

challenges faced in its development and possible alternative solutions are discussed in details. Lastly, section 3 provides conclusions and future directions.

2. Platform Design and Implementation

Prior to development of the CSDILA Platform, a generic conceptual framework for design and development of an integrated web-based 3D visualization and modeling platform was created based on the needs highlighted by academic and industry work. The process of designing the framework is discussed in details in the following sections.

2.1 Conceptual Framework

The generic framework and its components for realization of an integrated 3D visualization and modeling platform for supporting sustainable development were designed and put together to serve the requirements extracted by the extensive review of the literature and investigation of (a) methods and essential requirements of integrated modeling tools for sustainability analysis, (b) important visualization factors and existing technologies to illustrate spatial and non-spatial aspects of results, and (c) identification and study of previously developed web-based tools/platforms and evaluation of their technical and non-technical requirements, architecture, the technologies utilized in them, and finally their limitations.

In their comprehensive study, Hilderbrandt and Dollner (2010) identified and listed 19 general requirements from the literature for web-based analytical and visualization platforms. In this work, their list was extended by the addition of number of requirements to address the limitations raised earlier in this paper. This extended list classifies requirements into five general categories: *User Interface*, *Visualization*, *Modeling and functional*, *Data management*, and *General requirements*. The requirements in each category are described here.

2.1.1 User Interface Requirements

The user interface must be simple, intuitive, lightweight, and allows a high level of interactivity. It also must be sufficiently flexible to enable customization according to the preference of the user as well as allowing a *coordinated* single/multi-view (split-screen) structure to support multi scene visualization.

2.1.2 Visualization Requirements

Visualization requirements on the other hand include provision of real-time, effective, and high quality visual representation of urban scene and objects (e.g. buildings, traffic network, utilities, city furniture, etc.) in 3D; collaborative visualization; support for styling visual representations; visualization of massive amount of geo-data; visualization in different levels of details and support for generalization; show and hide capabilities at object or layer level; photorealistic views; temporal visualization; and visualization of results of modeling using proper visualization techniques.

2.1.3 Modeling and Functional Requirements

The requirements for modeling and other functions include navigation through the 3D environment with support for zooming, panning, tilting and smooth seamless transitions between different locations; effective user and access level management to support security and privacy of data; support for integration and real-time data acquisition from a variety of heterogeneous sources in raster and vector formats; support for

interoperability; provision of data definition (spatial/non-spatial) and manipulation in 2D and 3D to the user; support for computational analysis, modeling and simulation functions on the whole or subset of the city; multi-design evaluation by public to support Public Private Partnership (PPP); tracking changes in the structure of the city based on time-stamped data; report generation based on preference of the user; and support for the workflow engine for flexible definition and execution of modeling functions.

2.1.4 Data Management Requirements

This set of requirements include: effective data storage and administration; support for modern data acquisition models and importing/integration of commonly used formats; definition of layers and 4D objects, their relationships and spatial and non-spatial attributes; support for storage of objects at different levels of details; support of dynamic geo-data; data management at object level; support for export of data to commonly exchanged formats; allowance for necessary conversion between formats within the platform; storage of library of 3D models; support for event-driven data management for data quality and consistency.

2.1.5 General Requirements

General requirements for the platform are: to be web-based, have modular structure for supporting extensibility and flexibility; support a high degree of interactivity; allow for reducing and controlling complexity; be independent from hardware and software; to be economic (e.g. low cost of investment, hardware, maintenance and extension); and supports distribution of the system across multiple computers and locations.

Collective analysis of these requirements led us to the design of a generic framework for the development of a platform which is able to satisfy the majority of these needs. Here, the framework and its components are explained in details. The framework is composed of nine generic components in 4 conceptual layers: *data storage*, *data management*, *business*, and *presentation* layers which are illustrated in figure 1. Each component may be furthered broken down into number of sub-components.

At the most fundamental level of the framework, the data layer must cover two components to provide the necessary data for the platform to work. These include a **4D Data Store** which is the repository of nearly all major required data for the platform; and an **External Data Interface** which includes interfaces for communication with external data services such as Web Mapping Service (WMS) or Web Feature Service (WFS). Real-time integration of these services with the system may create added value by consolidation of up-to-date data from different data/service providers and avoiding redundancy in data storage. Utilization of suitable acquisition methods and quality and consistency check for integration becomes critical here. This falls under the authority of the data integration and management layer at the higher level. Data integration and interoperability components in this layer such as **data acquisition (import) and export components**, **integrity and consistency checking component**, and **converter components** are responsible for integrating the data from a variety of sources, the necessary data conversion, possible exports to other platforms via download or web service, as well as their quality assurance to comply with possible defined rules by the administrator in the platform. This layer is also responsible for establishing and managing the database through the **data store interface API** component. This extensible module allows for definition of different syntax and

objects which are used for establishing and working with different kinds of databases. It also makes it flexible for users to choose the database based on their purpose and preferences.

The Data integration and management layer, in general, maintains constant connection between the core business layer of the platform and the quality data from variety of sources.

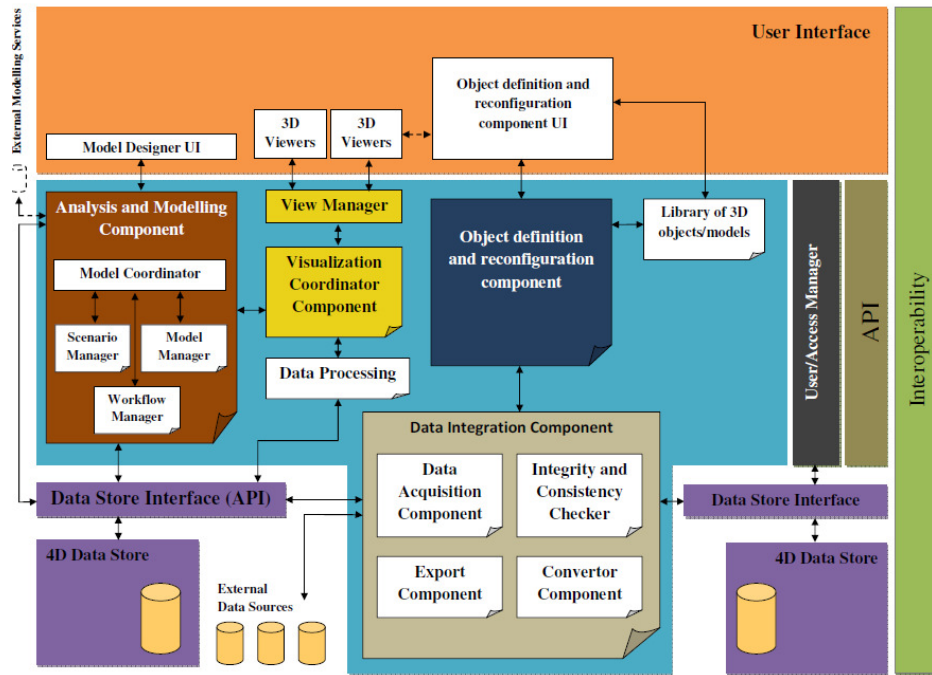


Figure 1: Conceptual framework for an integrated web-based 3D visualization and modeling platform

The business layer is the most important layer which contains the core components that take care of the main functionalities of the platform. The *User access manager* component is responsible for managing users, their access level to data and other functions of the system. The *3D library manager* on the other hand is designed for managing the 3D models library in the system. Models in this library (e.g. COLLADA or 3D Max models) can be incorporated along with geometry of the urban objects during their definition to allow photorealistic view of the city. The most important component in this layer is the *Analysis and Modeling component* which is further broken into a number of integrated sub-components. Modeling functions are defined using the API of the platform and managed in the *model manager* sub-component. In addition to these, external modeling services (external individual functions or a complete model) can be integrated to the model manager and used in modeling a particular scenario. It is the responsibility of the *model coordinator* to manage inputs and outputs of each model and coordinate the order of execution. *Object definition and reconfiguration* is another component in business layer which is integrated with its counterpart in the presentation layer to define, modify or remove objects in the data store. The *data processing component* in this layer processes raw data from the database and external sources and converts them into the data model of the platform for analysis and visualization. These processed data then are sent to the visualization *coordinator*. The coordinator module must communicate with the modeling and analysis component to be able to integrate results of modeling and those from data processing unit and pass them to *view manager* to determine how the visualization will be performed and prepare suitable data for that purpose.

Due to different nature of components in the framework and importance of effective communication between them,

interoperability must be assured throughout the platform between its modules as a requirement. A comprehensive data model is then needed to be designed for this purpose. Each component should know how to communicate with the others. Hence, the data model needs to be sufficiently rich to support syntactic and semantic interoperability for effective data exchange between external sources and internal components of the platform. The *API* of the platform on the other hand is required to be designed so that its further extension becomes possible.

Lastly, the presentation layer is responsible for visualization of the result and satisfies the requirements introduced in visualization as well as the user interface groups (section 2.1.1 and 2.1.2). The *3D viewer(s)* of the platform are the most important components in this layer and are defined here. In addition, interfaces for data definition, importing/exporting and management of other parts of the platform should be developed in this layer. The *Model Designer* is another important component and is responsible for visually managing the definition and integration of models within one or more scenarios. It is connected to the workflow engine of the platform in the business layer and uses the metadata of defined models to visually create a workflow.

2.2 Implementation Architecture

According to the framework introduced in the previous section, CSDILA Platform was developed. The platform is a prototype system which allows effective 3D visualization and flexible modeling of various types of analysis such as sustainability modeling, land administration (e.g. housing permit and property market analysis) and risk management (e.g. bushfire analysis) related models to support decision making for sustainable development. Additional functionalities of the platform include

3D measurement tools, presentation of attributes of a selected individual or group of objects, show/hide layers and objects, add/remove object layers and their grouping into folders, querying the objects within layers based on particular or group of attributes, defining a new layer and real-time creation of objects in the form of point, line, polygon (3D solid objects will be added in the future), visualization of Different LODs, etc. In this section, the architecture of the system and the technologies used for the development are explained.

The prototype system was developed using Service Oriented Architecture (SOA) in four conceptual layers – i.e. *data layer*, *data access layer*, *engine*, and *presentation layer* – which are illustrated in figure 2. Object Oriented Programming (OOP) was utilized to develop modules in all these layers. They are however, loosely coupled with each other and use web services and Extensible Markup Language (XML) to communicate. In this implementation, the database server was located remotely from the data access layer and the engine of the platform and visualization can be done on any client with a geospatial browser (Isikdag & Zlatanova 2010).

In the data layer at the lowest level, Oracle spatial 11.2g was utilized as the information backend to allow the storage of time-stamped spatial and non-spatial data. It also contains user data, system logs, 3D models library, and metadata of internally or externally available modeling modules and the data services. Oracle Spatial was utilized in implementation of the CSDILA Platform for its power as well as its license being made available to the University of Melbourne. Experience in this work shows that despite its power, Oracle Spatial is costly (hardware and the human aspects) to maintain. Despite the existing experience in database management systems (DBMS) in the CSDILA and the availability of support documents on the Oracle website, it was found that expert's knowledge is required for the management and maintenance of Oracle Spatial. Hence, it is highly recommended to consider simpler and open source database management systems (e.g. PostgreSQL) in developments and migrate to Oracle Spatial only if it is necessary.

The data access layer is implemented as part of the middle layer of the architecture. Oracle Data Provider for .NET (ODP.NET) and C# programming language along with .NET Framework 4.0 was used to create an API to communicate between the Oracle database and the Unified Data Model (UDM) developed for the engine of the platform. This API understands the objects in Oracle spatial and is able to execute commands for storage, retrieval and manipulation of data in the database using direct method or stored procedures in the DBMS. As mentioned earlier, this API can be further extended to work with SQL Server, PostgreSQL or any other databases that support storage, retrieval and different operations on spatial data.

Similar to the data access layer, ASP.NET with C# and .NET Framework libraries for their strong OO capabilities were utilized to develop the modular structure of the engine of the platform. The engine is composed of various modules explained in the framework for data management, user management, modeling workflow engine, object definition and reconfiguration and data processing unit integrated with the visualization coordinator and manager.

For assuring the quality of the communication between the above components, as mentioned earlier, a UDM was designed. It defines the data structure and possible contents which each component in the platform may be required to process or

communicate in that specific form. Considering services and XML as the communication medium among components in the platform, XML schemas were utilized by the integrity and consistency checker for validation and data transformation in the CSDILA Platform.

A variety of modeling modules can be developed using C# and integrated with the system using the developed API of the platform. These analysis models are then imported as a Dynamic Link Library (DLL) file. During the import, they are checked in the system, their metadata are automatically extracted based on the defined attributes in the library and saved in flexible groups of models in the platform. Later, whenever required, they can be loaded into the memory and used individually or along with other models within the workflow engine component to model a particular scenario.

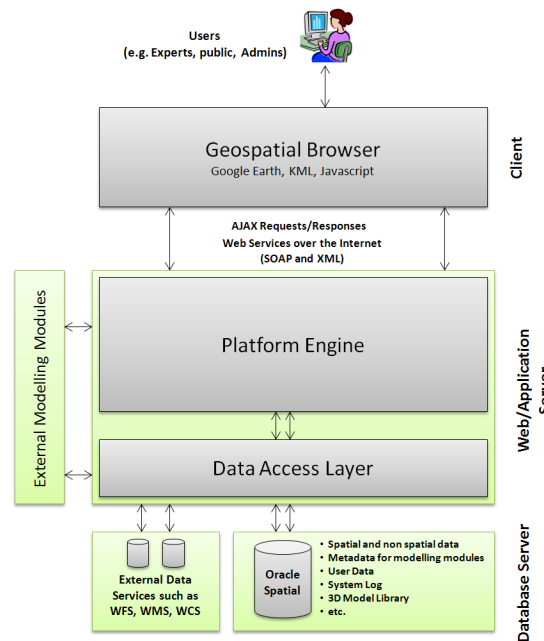


Figure 2: Architecture of the CSDILA Platform

In this layer, after processing of the data and incorporating the end result of modeling in the workflow engine modeler component, the outcome is sent to the visualization coordinator and based on the communication with the client, the type of visualization technique and technologies to be used at the client-side are determined and suitable data for processing and visualization is sent to the view manager. In the CSDILA Platform, the view manager was implemented at the client side using custom-developed Javascript classes and Google Earth API. This component is responsible for communication between the client and the server to manipulate and visualize the 3D environment in one or more viewers. It will be discussed shortly.

At the highest level of the architecture (presentation layer), on the client-side, "Geospatial web browsers" (Isikdag & Zlatanova 2010) and Google Earth virtual globe viewer, its API and Keyhole Markup Language (KML) along with JavaScript language, was utilized for visualization.

Data is exchanged between the visualization coordinator (server-side) and the view manager of the platform (client-side). In the first attempt in implementation of the platform, XML was utilized for communication between these two modules.

However, the performance tests in CSDILA and previous research (Elwannas 2011; Wu et al. 2010) showed that XML has too much overhead in communicating large amount of data. For this reason other technologies were tested to find an alternative data exchange format. In this process, JSON was researched and found as an efficient alternative to XML. It is text-based (similar to XML) and much more compact and can be flexibly transformed to XML and vice versa. However, choosing JSON required us to develop additional classes at both server and client sides to handle the transmitted JSON data. This data is converted to KML inside the view manager module developed using JavaScript classes in the client. Google Earth's API was incorporated for visualization in the viewer. The view manager is constantly reacting to user actions by querying the server and subsequently processes the responses.

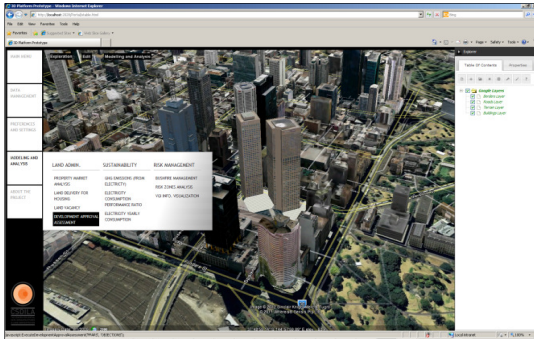


Figure 3: CSDILA Platform User Interface

Google Earth (GE) was utilized in this prototype because of its seamless visualization of the Earth along with highly developed API which allows manipulation and visualization at object level and access to their geometry and non-spatial attributes using Document Object Model (DOM). However, a number of limitations in using this product were identified. (a) The first is that GE does not support underground visualization. Many of the urban features such as utilities (e.g. gas or water pipes, tunnels, etc.) are located below the ground and there is a need for their inclusion in visualization and modeling; (b) the issues related to its performance since GE is very efficient for small number of objects; but as the number of objects in the viewer increases, it lowers down the performance of the computer since all objects remain in main memory and sometimes it results in the viewer plug-in to crash. Performance tests in this research showed that more than a gigabyte of main memory (RAM) was used for visualization of only nine thousand objects. This should be considered in further development of the platform since sometimes hundreds of thousands of objects (even though at low level of details) throughout the city are required to be visualized; (c) It also takes long time to load the objects in the GE viewer as the number of objects increases. To solve this problem however, Wu et al (2010) suggest the progressive transmission of the data which will increase the effectiveness of the platform in this case.

Considering the advantages and disadvantages of Google Earth, a number of alternative technologies such as WebGL, 3D web rendering services and visualization pipeline (Hildebrandt & Dollner 2010; Wood et al. 2008), and CityGML and VRML viewers (Dollner et al. 2006; Elwannas 2011) can be considered to be utilized as the viewer for the platform.

The User Interface (UI) of the platform was developed using Dynamic XHTML 4.01 and Ajax. A number of complementary JavaScript libraries such as *JQuery* and *Prototype* were also

used to complement the developed classes for data visualization and handling client interactions with the UI. Figure 3 illustrates the user interface of the CSDILA platform. The main challenge faced in this phase of development of the platform was the cross-browser programming since some browsers may or may not support particular JavaScript functions or library.

To test the feasibility of the platform, a number of models were developed to apply to the city of Melbourne. These models included *property vacancy*, *market analysis*, and *housing*. Property market model analyzes the properties in Melbourne's property market and after processing the data, it visualizes the market in 3D using color (the transaction date) and height (representing the value of transaction) to provide a clear snapshot of the trends in the market to the decision maker. On the other hand, housing models aim to visualize the complex process of issuing approval for house development in Melbourne. 3D visualization is proposed here to facilitate the visual presentation of the correlation between factors which are thought to cause delay in issuance of the development permits. Other models – carbon footprint as well as energy consumption of buildings in the University of Melbourne to visualize energy efficiency level of buildings to identify the trends and hotspots throughout the university. The respective algorithms for these models were provided by the experts in the field in the University of Melbourne and models were developed in the Centre for SDIs and Land Administration. Figure 4 shows the result of 3D property market modeling and visualization as an example of above models.

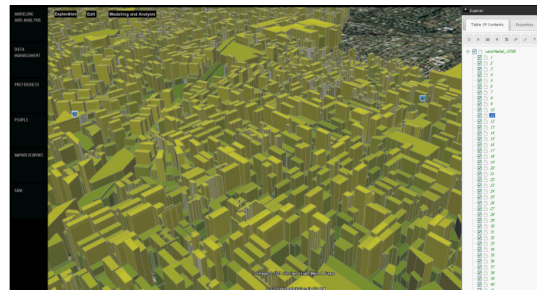


Figure 4: Property market modeling in CSDILA Platform

The CSDILA Platform and the visualized results in the 3D environment were then presented to a number of academic and industry experts in planning area during separate interview sessions for evaluation purposes. The received feedbacks were positive and the results of the modeling and visualization were found promising. They indicated that such a tool can increase the effectiveness of the decision making process for a variety of aspects of sustainable development by providing number of benefits such as immediate detection of different positive and negative aspects of the current/proposed plans in 3D environment and their comparison in multiple viewers; and fast and easy implementation – or modification – of the models and incorporating them into a scenario to reflect views of different stakeholders.

3. Conclusions and Future Research

To support sustainable development, there are different functions and needs to be satisfied. This paper explains a 3D visualization and modeling platform – CSDILA Platform – to support sustainable development in particular in urban context. The development of the platform was facilitated using a generic framework which was designed in the Centre for SDIs and Land

Administration in the University of Melbourne and based on an analysis of a variety of factors and requirements as well as a study of the literature to address the limitations of previous works. The CSDILA Platform was implemented using a number of technologies in a four-tier SOA architecture. This platform allows for flexible and powerful modeling and effective visualization while extensibility for its architecture and development and integration of more models are supported.

Some of technologies used in this platform are promising and bring strong capabilities, flexibility and extensibility to the architecture of the platform. In contrast, some of the others need reconsideration, and alternative technologies or approaches for their technical improvement should be utilized. Further, the modeling and visualization of the platform was tested for demonstration of the carbon footprint analysis of the University of Melbourne's buildings, land market analysis and public housing analysis in two suburbs of Melbourne, Australia. They showed promising results within the current activities of CSDILA research center for sustainable development.

Future works include research into recently emerged technologies such as WebGL and evaluation of their use for the developed platform. On the other hand, some of the aspects introduced in the generic framework were not implemented in the first version of the CSDILA Platform. The first aspect is that currently the platform only supports 2D and 2.5D (extrusion as well as multi-surface geometry) visualization. Another aspect is that in the current version, for integrating the models with each other to create a scenario, workflows has been hardcoded in the developed modules within the workflow engine and it is not currently possible to flexibly design the workflow by non-expert users. However, this is not a critical aspect of the platform since the model designer component is just a facilitator and does not affect the overall implementation of the platform. Models can still be imported, integrated with others, and run using the Model coordinator component of the platform. In the next version however, above features will be fully implemented and capabilities of the CSDILA Platform will be extended to full 3D modeling and visualization; and the workflow engine UI will be fully developed to provide flexible design of workflows and automatic integration of models and their respective parameters.

References

- AAM, 2012. K2Vi. <http://aamgroup.com/products/k2vi.cfm> (5 January 2012)
- Altaier, A., & Kolbe, T. H., 2003. Applications and Solutions of development proposals: reactions from planning and related professionals. *Computer, Environment and Urban Systems*, 29, 321-339.
- Berrick, S. W., 2009. Giovanni: A Web Service Workflow-based Data Visualization and Analysis System. *IEEE Transactions on Geoscience and Remote Sensing*, 47, 106-113.
- Brundtland, G. H., & Khalid, M., 1987. *Our common future*. Oxford, England: Oxford University Press.
- Dollner, J., Kolbe, T. H., Sgouros, T., & Teichmann, K., 2006. The Virtual 3D City Model of Berlin-Managing, Integrating, and Communicating Complex Urban Information. In: *Proceedings of the 25th Urban Data Management Symposium UDMS*, Aalborg, DK.
- Elwannas, R., 2011. 3D GIS: It's a Brave New World. In: *FIG Working Week 2011 - Bridging the gap between cultures*, Marrakech, Morocco.
- Giddings, B., Hopwood, B., & O'Brien, G., 2002. Environment, Economy and Society: Fitting Them Together Into Sustainable Development. *Sustainable Development*, 10, 187-196.
- Hildebrandt, D., & Dollner, J., 2010. Service-Oriented, Standards-based 3D geovisualization: Potential and Challenges. *Computers, Environment and Urban Systems*, 34, 484-495.
- Isaacs, J., Falconer, R., & Blackwood, D., 2008. A Unique Approach to Visualizing Sustainability in the Built Environment. In: *International Visualization Conference*, London, UK.
- Isaacs, J., Falconer, R., Blackwood, D., & Gilmour, D., 2011. Immersive and Non-Immersive 3D virtual city: decision support tool for urban sustainability. *Journal of Information Technology in Construction*, 16, 149-159.
- Isikdag, U., & Zlatanova, S., 2010. Interactive modelling of buildings in Google Earth: A 3D tool for Urban Planning. *Developments in 3D Geo-Information Sciences*, 1, 52-70.
- UNFPA., 2007. Urbanization: A Majority in Cities. <http://www.unfpa.org/pds/urbanization.htm> (2 January 2012)
- Wood, J., Brodli, K., Seo, J., Duke, D., & Walton, J., 2008. A Web Services Architecture for Visualization. In: *4th IEEE International Conference on e-Science*, Indiana, U.S.
- Wu, H., Zhengwei, H., & Gong, J., 2010. A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes. *Computers, Environment and Urban Systems*, 34, 291-298.

Acknowledgments

Authors would like to acknowledge the support of Department of Infrastructure Engineering, Geomatics discipline, and members of the CSDILA in The University of Melbourne, and in particular the contribution of Dr. Alex Tao Chen in this work that assisted in the development of the platform prototype. However, the views expressed in this paper are authors' and do not essentially reflect any of these parties' points of view.