# Semantic Web for an Integrated Urban Software System

**H.A.J. Schevers and R.M. Drogemuller**

***CSIRO Manufacturing & Infrastructure Technology, Melbourne, Victoria, Australia***
***email: Hans.Schevers@csiro.au***

***Keywords***: *Urban development; Semantic Web; interoperability; internet; software integration.*

**EXTENDED ABSTRACT**

Many software applications are available to support the assessment of urban development for specific domains such as land use, transportation, energy usage, rainfall run-off and urban water systems. However, addressing all these applications in one project is hardly feasible.

Currently no framework is available for integrating these applications. This means that integration efforts in this domain largely rely on proprietary solutions.

For the past few years, a lot of effort has been made to develop the next generation of the internet, called Semantic Web. The Semantic Web is an extension of the current web, and offers technology to make data on the web machine-interpretable (Figure 1).

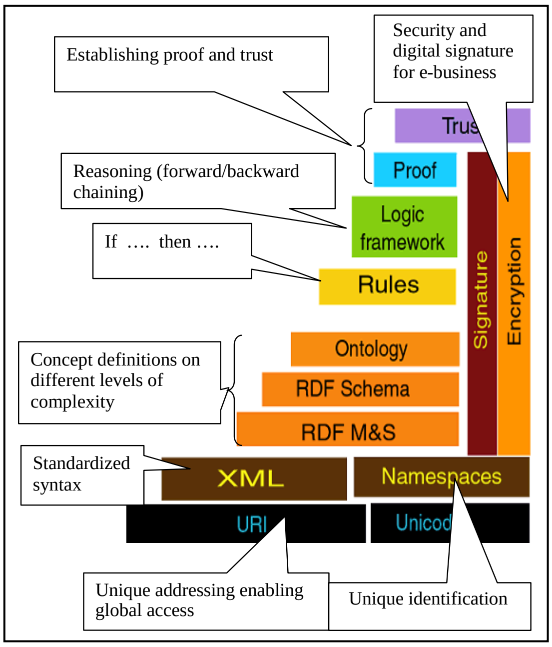


Figure 1. The different layers of technology of the Semantic Web

Machine-interpretable data, often called ontologies, allows the computer to infer on the information. Applications can be built upon this machine-interpretable information, which can work in collaboration with other services.

In order to establish this collaboration, interoperability is necessary. By relating machine-interpretable information to each other, inference support can help consistency and can support further interoperability. For example, consider that *a*, *b* and *c* are information pieces that are necessary for different applications to run. When $a = b$ and $b = c$, then you could infer that $a = c$. This means that information piece *a* can also be used as *c* for another application. Semantic Web supports these kinds of reasoning processes.

This paper proposes to use Semantic Web technology for integrating applications for urban development. A prototype implementation has been developed based upon this technology as a proof of concept (Figure 2).

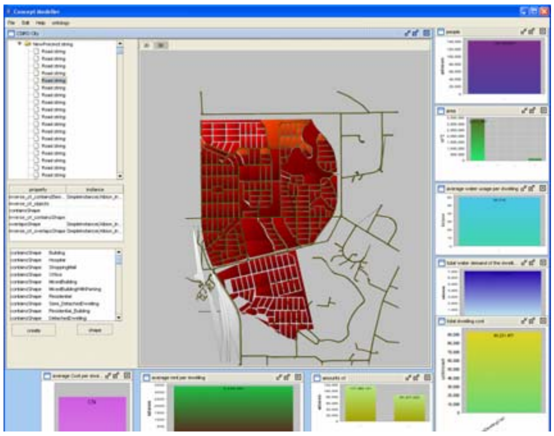

Figure 2. Screenshot of the prototype using Semantic Web Technology

This prototype is able to use spatial information and combine it with both urban master plan designs and water demand information. Reasoning mechanisms are used to analyse simple characteristics of the urban master plan.

## 1. INTRODUCTION

Building prototypes in the construction industry is hardly feasible, and consequently virtual prototyping is perceived as one of the solutions (CRC for Construction Innovation 2005) for improving design solutions. Using virtual prototypes, stakeholders can inspect the end result before it is actually built. Ideally, all relevant perspectives or views of the potential end result should be supported. This includes making multidisciplinary predictions on behaviour. Virtual prototype systems for urban development designs such as master plans and neighbourhood designs are not readily available. However, many computer-based models are available to support different assessments. For example, data and models are available that deal with land use, such as geographic information systems (GIS) and land use datasets, transportation analyses, costs estimation, energy usage (demand and supply), urban water (demand and supply), noise, airflows, shading, Australian Model Code for Residential Development (AMCORD), accessibility (public transport), (fire) safety and construction planning. Consulting all these models in one project is hardly feasible because:

- People are not always aware of the existence of these applications.
- Users do not have the time/budget/expertise to use the applications. Often expert knowledge is required to use these applications.
- Software applications are not integrated, making their usage cumbersome. Data re-entry because of syntax and semantic differences makes the use of many applications cumbersome.
- Software applications can operate on different levels of detail, and therefore the necessary input information may not always be available or it may be in the incorrect format, etc.

It seems that the elements for an integrated modelling environment are available (the individual software applications), but there is no framework enabling these applications to collaborate.

## 2. SEMANTIC WEB FOR URBAN DEVELOPMENT

### 2.1. Requirements for a software framework

Some key requirements can be formulated for a software framework for an integrated urban development system capable of assessing urban development plans:

- The framework needs to make urban development applications interoperable. Integrating the models means that information can be reused across the models. Consistency and the relationships between pieces of information need to be known.
- As people may be unaware of the existence of relevant software applications, the framework should support the search for available models. The framework needs to make the applications available and accessible automatically.
- Obviously developing a large integrated system all at once is not practicable and therefore a more evolving approach is more feasible. In addition, different parties should be able to extend the system. To support this evolution the system should be based on open standards.

### 2.2. Semantic Web

'Semantic Web' is a term coined by Berners–Lee *et al*. (2001) to define the goal to make data on the web machine-interpretable. This means that computers are able to infer on information. One consequence is that this would enable computers to perform knowledge-intensive tasks using the internet as a resource. It also means that it can provide inference support for interoperability. The SW provides a standardised framework that allows data to be shared and reused across applications, enterprises and community boundaries. Other aspects of the Semantic Web are:

- Its open approach – this means that people can contribute to the Semantic Web without having to go to a central organisation. Basically, it is similar to the current internet where people can publish their own websites. The new Semantic Web allows people to create their content as well.
- Its evolutionary approach – different computer languages exist that enable the development of machine-interpretable data. These languages are built upon each other and can be used in conjunction to offer different levels of functionality.
- Its (by nature) distributed approach – this means that people/organisations can publish their own machine-interpretable data on the web, but still be (loosely) connected to each other.

The standardisation of communication languages of the Semantic Web reduces syntax problems between applications. The semantic differences are of course more difficult to resolve. However, the machine-interpretability of information enables one to infer on the information. Consequently, linkages between different information sources can be inferred. By developing machine-interpretable input and output models for each application, inference support can help to link these applications together (Figure 3).

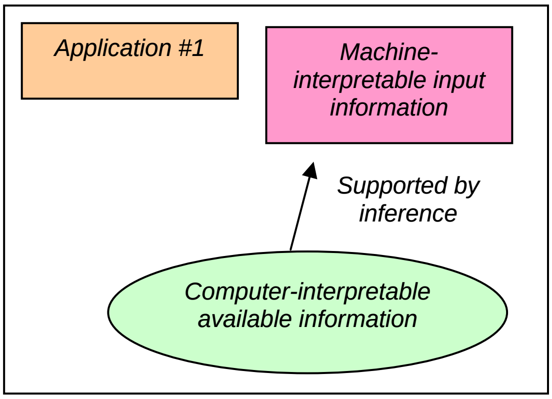


**Figure 3.** Machine-interpretable information can support interoperability by reasoning on the available and necessary information.

Obviously this mechanism can also be applied for software related to urban development. It means that urban development information such as design information has to become machine-interpretable. Having applications residing on the web with machine-interpretable input and output information, inference can help to support the interoperability.

Furthermore, the open and distributed aspects of the Semantic Web support the creation of different assessment software applications independently. This means that the integration of the software is done by loosely coupling the software. This approach is modular and thus more manageable than one large software application. The evolutionary aspects enable the insertion of new (independently developed) prediction models. In time, an integrated assessment could emerge.

### 2.3. Overview of Semantic Web technology

This section provides an overview of the key technologies that form the basis of the Semantic Web framework (Figure 4). To make data on the web machine-interpretable, the Semantic Web uses meta-data to describe the data on the web. Languages such as RDF (Resource Description Framework) (World Wide Web Consortium 2005a) and OWL (Web Ontology Language) (World Wide Web Consortium 2005b) have been developed to make data more machine-interpretable. These languages allow the 'marking-up' data with meta-data (data on data). Consequently, these languages support the creation of ontologies that are computer-interpretable formalisations of concepts. The next step is to standardise rules and other knowledge-based assertions. Figure 4 shows the key technology for the Semantic Web, starting with unique addressing, towards ontologies, logic framework and eventually trust.

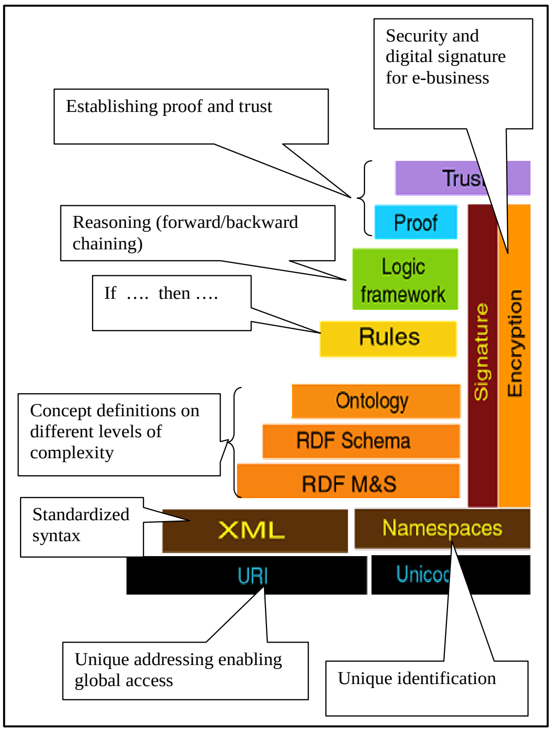


**Figure 4.** Layers of the technology that form an important framework for the Semantic Web.

### 2.4. Making data machine interpretable

RDF is a language that is able to mark up data in such a way that it becomes machine-interpretable. RDF's expressive capability to mark up data is less than OWL, but still very useful. The marking up is done by statements. In RDF, a statement can have the following structure: <Object> <Attribute> <Value>. Figure 5 shows an example of two statements.

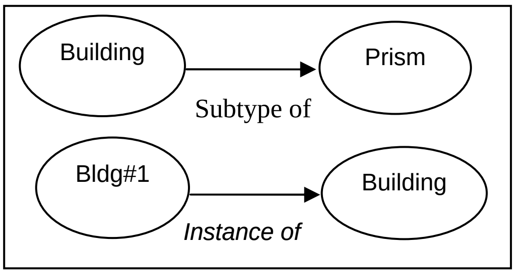


**Figure 5.** An example of two statements.

An RDF document can contain many of these statements. All objects are uniquely defined by a web address or a Uniform Resource Indicator (URI) like http://internal.csiro.au//Building#. Objects in different RDF documents can therefore refer to each other (Figure 6). For example, another RDF document could make statements like, for example, that http://internal.csiro.au//bldg34#, (defined in another RDF document,) is 30 meters high'.

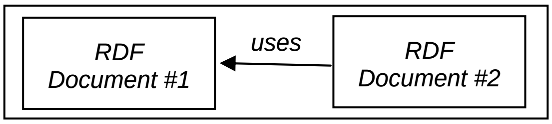


**Figure 6.** Elements in an RDF document can relate to other elements in other RDF documents, due to the unique addressing of each object within an RDF document.

A more object-oriented presentation can be inferred from all these statements. This object-oriented presentation contains classes that can be perceived as computer-interpretable formalisations of concepts or ontology. These classes can have properties and relationships with other classes, called 'slots'. This approach allows you to create your own class model reusing concepts of others. For example, an object 'Building' with a slot 'building height' defined in a RDF source can be extended by another RDF source with properties such as 'energy consumption' or 'water demand'.

OWL is based on RDF but offers more functionality. For example, OWL supports the classification of individual objects, which means that reasoners can classify individual objects as members of certain classes. For example, a high-rise building can be defined as a building with a certain minimum height. Classifiers can determine if instances of the type building also belong to the high-rise class as well.

Using all these machine-interpretable information sources, SW supports the chaining of web services together. When the ontologies/RDF/OWL sources are interrelated, interoperability can be achieved and consequently the web services can reuse each others data (Figure 7).

Chaining multiple web services together increases the number of knowledge-intensive task that can be carried out automatically (Fensel 2002). As the Semantic Web supports technology for describing, discovering and accessing web services, ultimately a chain of interoperable web services can be formed dynamically (Daconta *et al*. 2003).

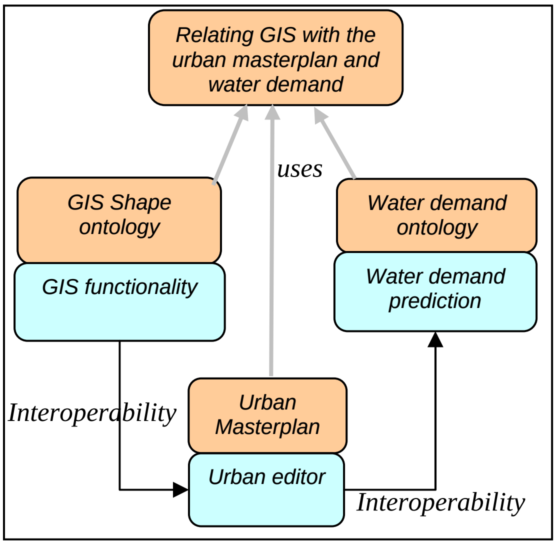


**Figure 7.** The interrelationships between Semantic Web services.

### 2.5. Semantic Web for integrated urban development

Software applications for urban development have been developed independently. These applications can be converted into ontology-driven applications. The ontologies can become aligned by relating them to each other by asserting new statements. For example, the concept 'building' that is available in both ontologies can be declared similar. Obviously this approach does not mean that mapping problems have disappeared or that interoperability is something that comes free. However, interoperability support is available by using inference engines, which can infer new hierarchies, based on the information sources and can check ontologies on their consistencies. This is particularly handy when ontologies are getting complex (when, for example, many different ontologies are being interrelated).

Having this architecture in mind, it is easy to adopt this technology and vision for urban development projects. For example, an urban development application could use GIS information that is stored in an ontology-driven GIS system. Already projects have been initiated to develop a standard GIS-RDF output such as RDFMap (Map Bureau 2003). The *Building* concept in the GIS ontology could be extended by a slot *buildingtype*. A water usage prediction application could extend the *Building* as well with properties such as *water usage*, *buildingtype*, *waste water* etc. Another application capable of calculating energy consumption can extend the ontology with energy slots and relationships. Figure 8 shows a network of software applications forming a collaboration which is possible when the ontologies are interoperable.

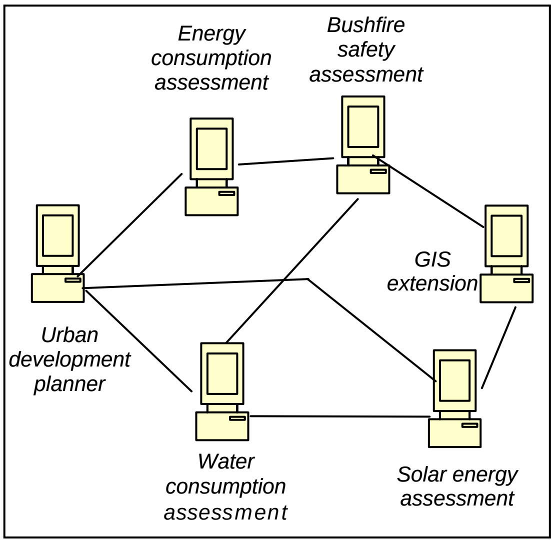


**Figure 8.** A potential Semantic Web for urban development.

The applications extend the ontology with their own concepts and slots. Inferring on all the ontologies when several interrelationships have been asserted may result in the introduction of new relationships. For example, linking a *concept1* (in an application) with another *concept2* (in another) application2 creates interoperability. If *concept2* is also linked with *concept3*, then *concept1* is also linked with *concept3*. This can be inferred by the computer enabling interoperability between applications 1 and 3. Figure 9 demonstrates how interoperability can be inferred between three systems. By several interoperability statements, new interoperability statements can be inferred.

*'buildingtype'(GIS ontology) = 'building type' (energy ontology)*

*'buildingtype' (water ontology) = buildingtype (energy water ontology)*

***(Inferred)***
*'buildingtype' (GIS ontology) = buildingtype (water ontology)*

**Figure 9.** Inferring interoperability.

## 3. A PROTOTYPE IMPLEMENTATION

### 3.1. Interrelating ontologies for urban master plans

A simple shape model has been developed to capture geometry into an ontology. Geometric objects such as a *Polygon*, *Polyline2D* (a polyline with a width to represent line infrastructure such as roads and canals. Based on this ontology, a viewer with some simple spatial functionality has been implemented.

To capture information related to urban development, classes such as *Precinc*', *Zone* and *SiteElements* are formalised. A *Precinct* can contain *Zone objects* such as a *Residential_Zone* or a *Commercial_Zone*. In addition, a *Precinct* can also contain *SiteElements* such as *Road*, *Building* or *OpenSpace*. This simple ontology is able to capture some basic information of an urban master plan.

A relating ontology connects the concepts in the master plan ontology with the shape ontology (Figure 10).

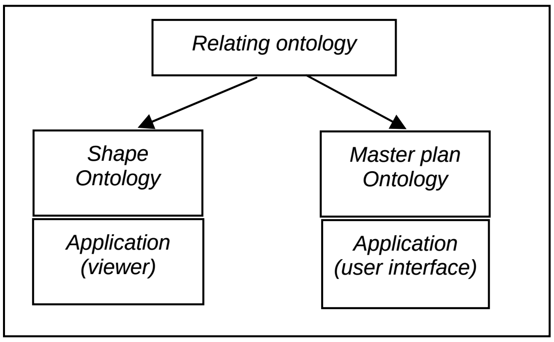


**Figure 10.** The ontologies for the prototype.

The relating ontology contains statements that the *SiteElement* classes are subtypes of *Shape* class. This means that these classes inherit the properties and relations of the *Shape* class and consequently can capture geometry. For the *Building* class the supertype *Prism* is used. This means that a *Building* can be represented by a *Prism*. For *LineInfrastructure* classes, *Polyline2D* is used to capture the geometry of roads, canals etc. The *Polygon2D* class is used for *Zones*. Reasoning on both information sources (shape and urban ontology), a new class diagram can be inferred. Figure 11 shows an abstraction of the urban ontology used in the prototype.

Another ontology is constructed containing different dwelling types and their average water demand, based on the number of people living in a dwelling. This source of information is also included in the existing ontology by extending the *Building* class with slots such as *Average Water Demand* and *House Type*. Simple rules can be attached to define the average house size based on type (large dwelling, townhouse, apartment etc.).

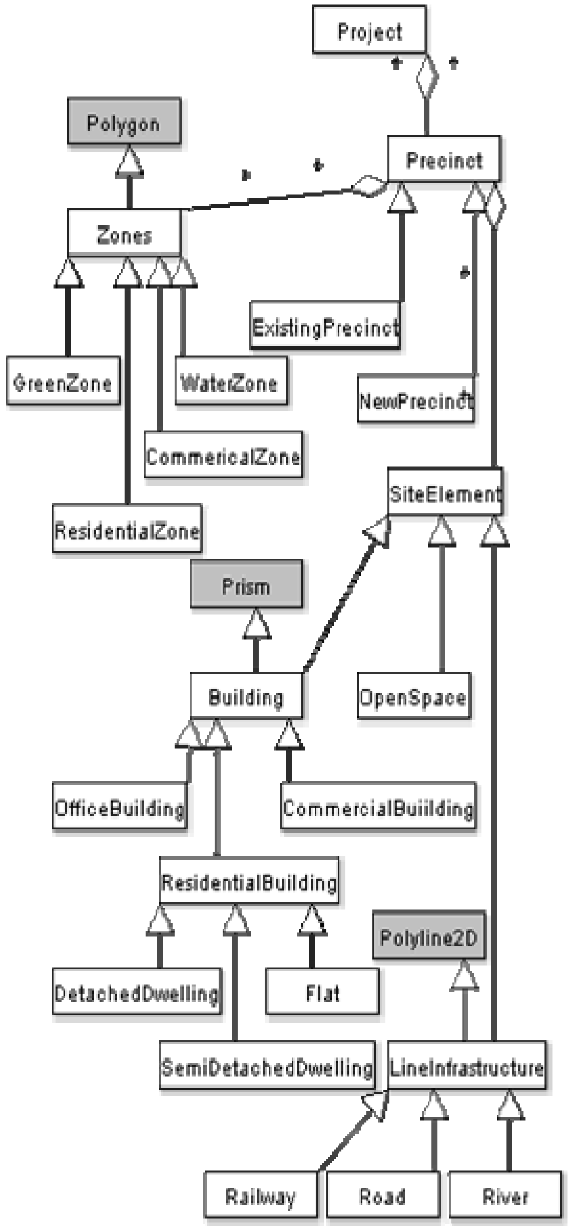


**Figure 11.** The inferred class model. The grey objects are described in the shape ontology. The white objects are described in the master plan ontology.

### 3.2. The prototype

The prototype uses Protégé (Stanford Medical Informatics 2005) as the ontology editor and database. This means that the classes and instances are managed by that application. Instances containing shape information have been developed and are used to capture geometry. A 2D user interface has been developed to visualise the content in Protégé. The classes in the shape model (classes containing geometry information) can be extended to more semantic classes such as precinct, zones, buildings and roads. All these classes can be instantiated and visualised (Figure 12).

Using Protégé, classes can be defined. Zone instances can be created by importing geometry from CAD files. A geometry agent makes sure that the *area* and *distances* etc. are available in the ontology. Using available search and query applications, the model can be browsed. In addition, writing rules using several different rule engines supports a quick assessment. For example, by putting in *the amount of dwellings per acre*, the *total amount of dwellings* can be calculated. When changing the amount per dwelling or changing the geometry, the system recalculates the amount of dwellings automatically.

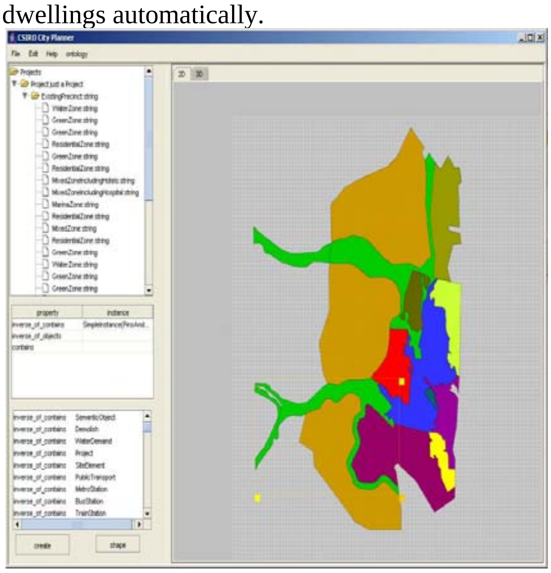

**Figure 12.** A screenshot of the use of a shape model to visualise zones.

Not only zones can be put in the system, but also more detailed objects such as roads, individual lots and buildings (Figure 13).

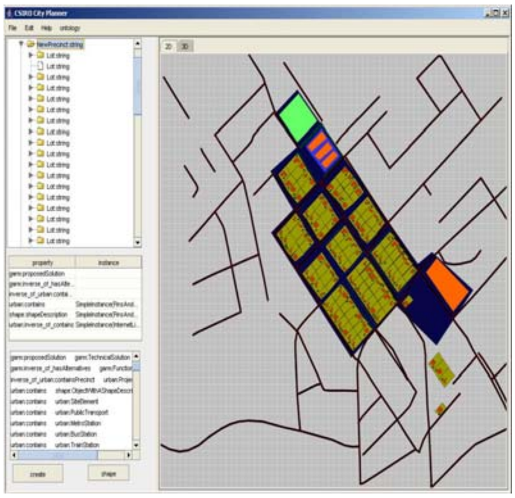

**Figure 13**. A screenshot of the use of the shape model to visualise roads, lots and houses.

Again, the geometry agent calculates the area of polygons, the length of polylines and the volume of prism. This basic information enables the writing of simple rules about, for example, average lot size, total lot size, ratio infrastructure/lot size, amount of buildings, and absolute and average distances to public transport or parks or roads. This information can of course be presented in chart (Figure 14).

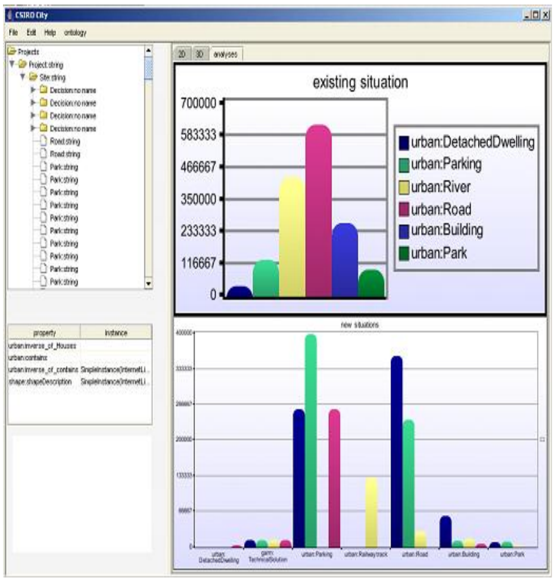


**Figure 14.** Screenshot of a chart displaying the amount of area in $m^2$ used per function.

## 4. CONCLUSIONS

Many applications are available for analysing urban development designs; however, these are not integrated. The Semantic Web technology provides a suitable framework to support the integration of these applications by machine-interpretable data. This approach enables inference on federated data sources, which supports consistency and interoperability. The evolutionary characteristics of loosely coupled integration allow individual software to be further developed.

## 5. REFERENCES

Berners–Lee, T., J. Hendler, and O. Lassila (2001), *The Semantic Web*, Scientific American [http://www.sciam.com/].

CRC for Construction Innovation (2005), *Construction 2020* [http://www.construction-innovation.info/] (accessed Sept. 2005).

Daconta, M.C., L.J. Obrst, and K.T. Smith (2003), *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*, Wiley Publishing Inc., Indianapolis, Indiana.

Fensel, D. (2002), *The Semantic Web: Yet Another Hip?*, Vrije Universiteit Amsterdam, The Netherlands.

Map Bureau (2003), *RDFMap* [http://www.mapbureau.com/rdfmap1.0/] (accessed Sept. 2005).

Stanford Medical Informatics (2005), *Protégé* [http://protege.stanford.edu/] (accessed Feb. 2005).

World Wide Web Consortium (2005a), *RDF* [http://www.w3.org/RDF/] (accessed Sept. 2005).

World Wide Web Consortium (2005b), *OWL* [http://www.w3.org/2004/OWL/] (accessed Sept. 2005).