Using Protégé for Automated Assessments for Urban Designs

Dr. Hans Schevers & Robin Drogemuller
CSIRO Manufacturing & Infrastructure Technology
PO Box 56 (37 Graham road)
Highett, Victoria 3190, Australia

Introduction

Currently software applications are able to predict certain aspects of urban designs such as energy usage (demand and supply), urban water (demand and supply), accessibility (public transport), safety (fire), etc. However an automated multi disciplinary assessment is hardly available. This presentation abstract discusses the adoption of protégé for a prototype software application for automating integrated assessments for urban development projects.

Overview of the City Planner

For a research project within CSIRO Australia, Protégé is being used for developing an application supporting the automation of integrated assessments for urban development. Different interlinked ontologies have been created to model urban development using objects such as streets, houses, parking spaces, etc. Based on these ontologies, several small software applications have been implemented on top of Protégé using the Protégé API such as visualisation using a shape ontology, a rule engine/editor and a DXF loader which can load Computer Aided Design files and convert them into individuals, etc (Figure 1) .

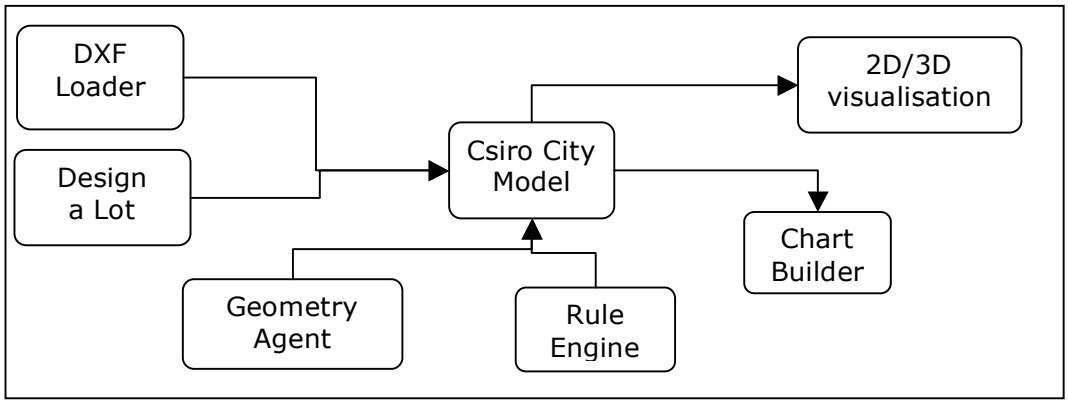

Figure 1 shows an overview of the current prototype

Using the DXF loader, building footprints, roads and sites can be imported including the geometry. All these objects can be visualised and basic manipulations are implemented, etc (Figure 2).

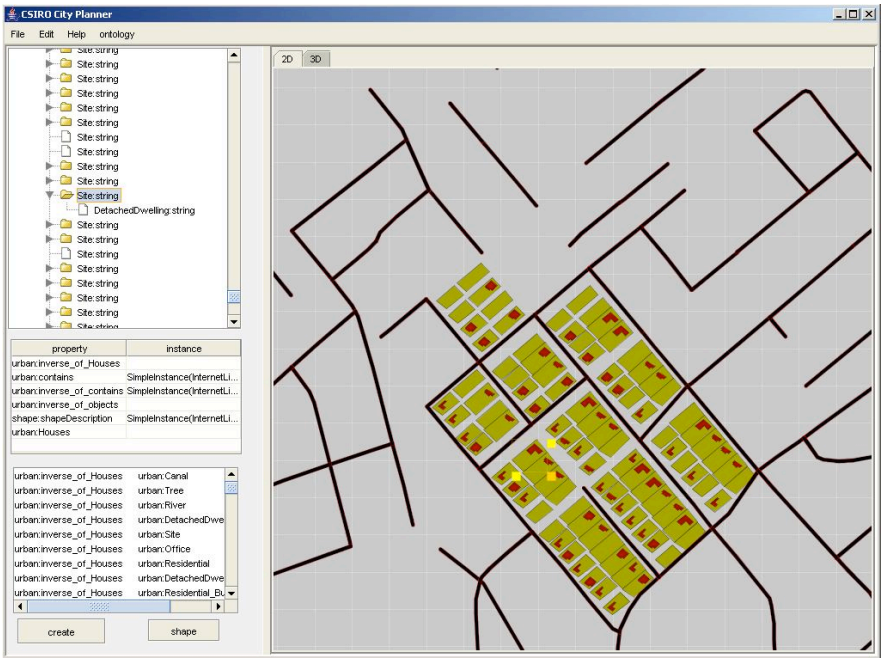

Figure 2 shows a screenshot of the urban design.

Using the chart builder application several charts can be defined and generated using the ontology defined in Protégé (Figure 3).

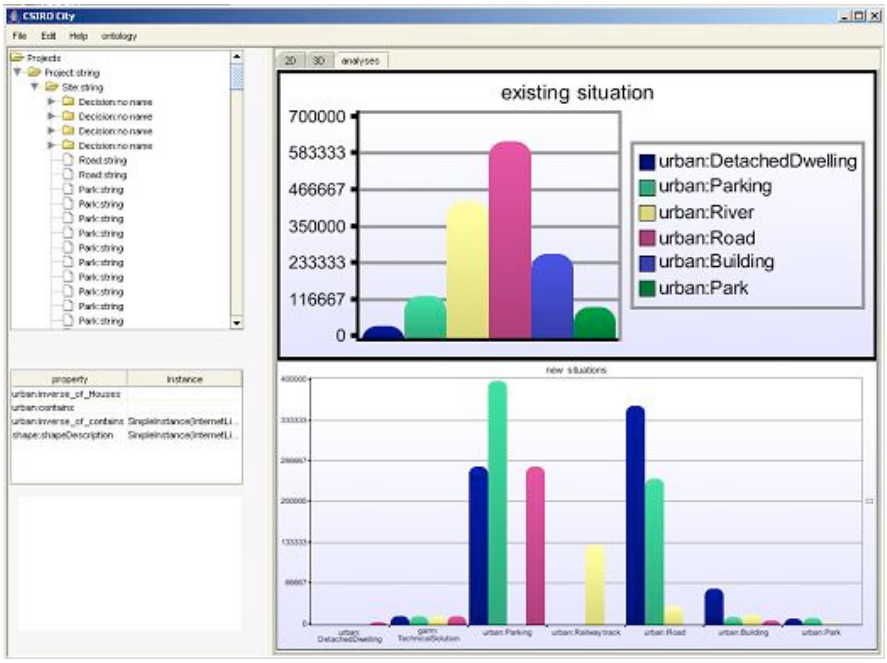

Figure 3 shows some basic characteristics of the design.

Furthermore using specially developed rule engine and rule editor, rules can be created and executed using Java BeanShell. Basically conditions can be specified as a pattern of individuals and their relations which trigger a Java script to be executed. Using this rule editor simple, rules can be created such as volume calculations of buildings by multiplying the height with the base area.

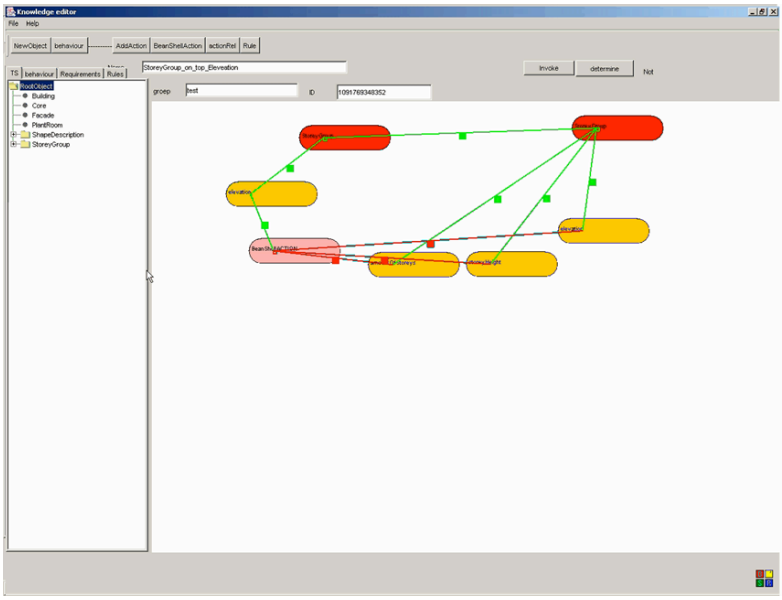

Figure 4 rule editor specifiying a condition clause containing a set of relating instances. When these instances are available a Java script will be executed

Implementation

Several components register listeners to the classes and individuals using the Protégé API. The 2D visualisation application creates a class listener reacting when an individual of a certain class is created. A new listener is created to keep track of the changes of this new individual and notifies the 2D visualisation application to redraw its content for that individual. This simple approach allows the creation of various independent applications working on one shared data model. Furthermore an agent has been developed linking the individuals of the shape model to a Java package called the Java Topology Suite [JTS]. This packages offers geometry related functionality especially for polygon shapes. This agent uses this functionality to determine which polygons are convex, overlapping and containing other polygons, area calculation etc.

Conclusions

The listeners and the clear protégé API support an evolutionary development of these several applications working on the same data model. The import of CAD data however generates a large amount of objects in the database resulting in an enormous use of RAM. Basically for each line in a CAD drawing 2 point objects and a line object is created in Protégé resulting in memory usage easily above 500mb. The performance of certain commands such as erasing objects becomes in these situations very slow. The available functionality to use a database is not yet explored.

References

[Protégé] http://protege.stanford.edu/
[JTS] http://www.vividsolutions.com/jts/
[BeanShell] http://www.beanshell.org/