

LIVE ENERGY TRNSYS – TRNSYS SIMULATION WITHIN GOOGLE SKETCHUP

Malcolm C Murray¹, Dr. Neil Finlayson¹, Dr. Michaël Kummert², John Macbeth¹

¹Greenspace Research, Lews Castle College UHI, Stornoway, Isle of Lewis, United Kingdom, HS2 0TW

²Energy Systems Research Unit, University of Strathclyde, James Weir Building, 75 Montrose Street, Glasgow, United Kingdom, G1 1XJ

ABSTRACT

With the current focus on energy performance certification, factors such as increased and unstable fuel prices and consequent heating and electrical costs as well as the large number of buildings that are going to require energy analysis, a real need has surfaced for powerful, fast, and easy to use energy analysis solutions that will not be limited to energy experts.

Live Energy (formerly called Demeter) [Finlayson et al, 2008], an energy analysis plug-in for Google's SketchUp utilises the SketchUp application programming interface and rich internet technologies to integrate into the SketchUp environment and make the process of building energy analysis as simple as possible. Live Energy builds a rich energy framework around the SketchUp building model, which can be sent to an accompanying energy analysis engine or be exported for use with other tools.

TRNSYS is a powerful transient systems simulation environment that allows for the creation of component-based models to represent energy systems (including localised weather, equipment, building structure etc.) based on real life scenarios and then to calculate various outputs based on the user's requirements. Due to a steep learning curve and a complex user interface for inputting building data, TRNSYS has long been the domain of academics and building services engineers. This paper outlines an attempt at simplifying entry to the TRNSYS environment while still maintaining a degree of control for advanced users.

The TRNSYS extension to Live Energy allows for the generation of a TRNSYS Multi-Zone building (Type56) component from a Google SketchUp model and will create an executable TRNSYS simulation that can be run directly from Google SketchUp, thereby combining the power of the TRNSYS simulation environment with the simplicity and ease of use of Google SketchUp.

INTRODUCTION

In a time of volatile fuel and material costs and a growing awareness of environmental issues, it is becoming imperative to those planning and constructing buildings to be aware of the energy performance of buildings throughout the life cycle – from conceptual design to building maintenance. In addition, with the introduction of the European Union Energy Performance of Buildings Directive (EPBD) [DIAG, 2009] it is now necessary for all building owners to analyse and produce energy performance certificates for their buildings.

With this huge increase in the demand for energy performance analysis, various software tools have been developed to analyse different aspects of a building's energy performance. These substantially differ in analysis complexity. For example, some tools analyse a building with the aim of generating an energy performance certificate while others simulate a building's energy systems in detail and provide various outputs depending on the needs of the user.

Many energy analysis tools require a building model to be created specifically for that tool, often graphically but sometimes via text input. With the vast number of energy analysis tools available and with their varying capabilities, it is often desirable to use more than one tool in a building project to acquire a more complete picture of performance. This can, however, be extremely time-consuming as a new model of the building may need to be created for each analysis tool, which may not match exactly due to user error or the limitations of the modelling software.

XML

Extensible Mark-up Language (XML) is a powerful and flexible text format specification, which was originally designed to meet the challenges of large-scale electronic publishing and now plays a dominant role in the exchange of electronic data [Bray, 2006].

XML files are made up of elements, which can contain sub elements as well as textual data values so that a document is represented in a tree-like format. Elements are enclosed in tags containing the

element's name surrounded by the '<' and '>' characters. As well as containing sub-elements and textual values, each element can contain attributes which are defined within the opening tag as a key - value pair e.g. id="some number". An XML document consists of self-describing data so that the author has complete freedom to name and represent data in any format they wish. This gives XML the flexibility to be applied in a large number of scenarios.

Most modern programming languages now have XML processors built in or available as external libraries making the use of XML a possibility in most applications. Technologies such as Document Object Model (DOM) [Le Hors, 2004], XML Language Path (XPath) [Clark, 1999] and E4X [Schneider et al, 2005] allow for the navigation of an XML tree structure. Xpath and E4X allow an XML structure to be navigated in a similar fashion to using uniform resource identifiers in navigating an operating system's file structure e.g. an XML element could be retrieved as follows:

/DocumentRoot/Element1/Element2

XML can also have associated styling information using XSLT technology. This allows an XML file to be transformed into other formats e.g. to conform to a file format, to display the information in a readable manner, or to transform from one XML vocabulary to another.

In any project where a given XML format is to be used between more than one developer or application, or where the developer is trying to create a standard data format, it is essential that the document format be formally specified and documented. To achieve this many XML technologies have been developed but one in particular is commonly used, namely XML Schema files [Van der Vlist, 2002].

XML Schema files define an XML document as a tree structure of elements, which consist of attributes and references to other elements. In addition to data types, various other restrictions can be placed on elements such as lists of enumerations that an element or attribute value can have, numerical ranges, order of elements etc.

In addition to validating XML documents, XML Schemas can be used to document XML vocabularies since the Schema's precision and conciseness can be difficult to replicate with natural language [Van der Vlist, 2002]. XML Schemas also aid in the understanding of XML as they are generally human readable and various tools exist to create graphical representations of XML Schemas (helped by the fact that XML Schemas are in fact XML documents themselves). Another benefit of using an XML Schema is the ability to bind the Schema to an application's object model so that a conforming XML document can automatically be represented as generated objects native to the application.

gbXML

Green Building XML (gbXML) [Kennedy, 2008] is an XML format designed to facilitate interoperability between building information tools and analysis systems. gbXML is an open standard that is available as an XML schema and offers a rich building representation which includes geometrical and energy related information.

gbXML now has support across the building information industry and has been adopted by companies such as Autodesk, Graphisoft and Bentley for use in their tools.

TRNSYS

TRNSYS [Klein et al., 2007] is a transient system simulation environment with a modular structure that allows independent components, consisting of validated mathematical models representing individual components in an energy system, to be joined together based on (real life) scenarios to produce an output based on the model requirements.

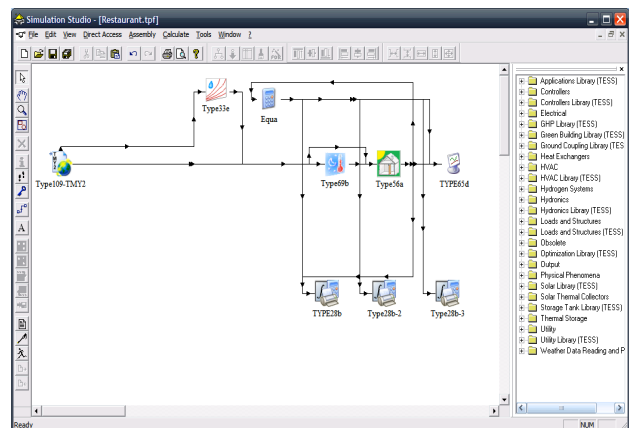


Figure 1 - TRNSYS Simulation Studio

TRNSYS originated in the 1970s as part of a joint project between the University of Wisconsin – Madison's Solar Energy Laboratory and the University of Colorado's Solar Energy Applications Laboratory. TRNSYS is currently at its 16th version and under continual development by a joint team consisting of the Solar Energy Laboratory (SEL) at the University of Wisconsin-Madison; The Centre Scientifique et Technique du Bâtiment (CSTB) in Sophia Antipolis, France; and Transsolar in Germany [Thermal Energy System Specialists, 2007]

The TRNSYS simulation engine was originally written using the FORTRAN programming language [Thermal Energy System Specialists, 2007] and uses a Dynamically Linked Library (DLL) system architecture to allow for a modular structure that can be extended by adding new DLLs to the system. This means that TRNSYS users can create custom components (e.g. to represent a new type of heating component) fairly easily using their choice of DLL compiling programming languages including FORTRAN, Pascal, C, C++ etc. [Klein et al., 2007]. TRNSYS also allows the user to link to external

programs for further analysis of results (e.g. Microsoft Excel, MATLAB).

The current version of TRNSYS (version 16) comes as a package containing various user interfaces and system elements that have been built around the TRNSYS simulation engine to allow the user to build and control simulations. The main interface used to control TRNSYS is the Simulation Studio (shown in Figure 1) which acts as a manager for TRNSYS – linking all user interfaces and system elements required for a simulation to operate and providing a platform for visually creating a representation of an energy system.

The TRNSYS Simulation Studio, originally called “IISiBat”, was developed by CSTB and has been included in the TRNSYS package since 1996. IISiBat was originally developed as a general-purpose simulation control environment before being adapted specifically for TRNSYS [Thermal Energy System Specialists, 2007]. As well as allowing the user to control the TRNSYS system, the simulation studio acts as a drag and drop assembly area where a user can select system components in one section of the simulation studio, drag and drop them onto the assembly area and then join the individual components using connectors and match component outputs with inputs. After the user has built their scenario in the assembly area, the simulation studio then generates a textual deck (DCK) file, which can be run through the TRNSYS engine with the results being displayed in a user defined output component (e.g. a graph plotter or file printer).

Building models are represented in TRNSYS using the Type56 (multi-zone building) component [SEL, 2007] which consists of an external file, called a BUI file, containing a building’s relevant geometric data. The BUI is essentially an ASCII text file written in the format dictated by TRNSYS’ BID Language, which is a set of syntactical rules defining the building.

A BUI file can be created manually, using a standard text editor, or by using the standalone TrnBuild tool that is included in the TRNSYS package and was originally developed by TransSolar [TRANSSOLAR, 2009]. TrnBuild provides the user with a graphical user interface where a building’s thermal zones and other information required by the BUI file format can be defined. TrnBuild also allows the user access to various libraries (including numerous construction and glazing libraries) to reduce the amount of data that a user has to input to create a BUI building descriptor. TrnBuild’s libraries are a series of ASCII text files located in various folders within the TRNSYS program directory so that TrnBuild’s libraries can be expanded by simply creating (or purchasing) library text files.

A major issue with the use of TrnBuild is that it does not feature a graphical representation of the building

model and instead relies on text-based entry of geometrical data, which can be error prone especially in the case of complex buildings.

Although a relatively complex tool, TRNSYS currently has a large international user base of engineers and academic researchers who both use the tool and develop extension components based on their own individual needs and projects.

Google SketchUp and LiveEnergy

Google SketchUp is a general-purpose 3D modelling tool with an intuitive user interface and feature-set to simplify the creation of 3D models and provides novice users with the ability to start creating content within a short period of time [Google, 2009]. Among one of the more innovative features found in Google SketchUp is the Push / Pull functionality [Schell et al, 2003] allowing users to create complex data structures using a single mouse click and drag. Features such as context aware tools also help enhance the user experience.

Google SketchUp is especially suited to the conceptual design stages in product design and is typically used in this role by architects, engineers and designers.

Google SketchUp also features integration with Google Earth [Google, 2009], Google’s 3-dimensional GIS application. Google Earth integration means that users can create a model of their actual or planned property, place it on Google Earth and view the model in its real world location. Features such as Google Earth integration and an intuitive interface mean that Google SketchUp also has many private as well as professional users.

To allow for third part add-on development, Google SketchUp features a Ruby based [Ruby-Lang Webmaster, 2009] API which allows access to Google SketchUp’s feature-set as well as a built in web browser for use in custom add-ons.

LiveEnergy, formerly called Demeter [Finlayson et al, 2008] is an add-on for Google SketchUp, that enables building specific properties to be associated with a SketchUp model and creates an associated gbXML based [Kennedy, 2008] representation of that model. The gbXML model can then be sent from LiveEnergy to the Green Building Studio (GBS) [Autodesk, 2009] energy analysis engine, via a web service, to analyse the SketchUp model and return an energy report.

LiveEnergy also allows for the importing and exporting of gbXML models to allow interoperability between Building Information Modelling tools so that, for example, a gbXML model exported from a tool such as Autodesk Revit can be imported into Google SketchUp and then be modified or analysed using the GBS engine.

As a general purpose-modelling tool, Google SketchUp does not include detailed building information parameters such as surface types, construction information or room spaces. LiveEnergy compensates for this by allowing the addition of missing building information properties in as intuitive a manner as possible. For instance, LiveEnergy allows the user to identify surface types by “painting” them with LiveEnergy specific materials in the Google SketchUp materials palette. Room spaces can be identified by double clicking on a room’s floor using a helper tool added to Google SketchUp as part of LiveEnergy.

The LiveEnergy user interface was created using Adobe Flex technology [Adobe, 2009] and communicates with the Google SketchUp API via a JavaScript interface. The LiveEnergy system architecture was built in a modular manner using various coding standards and frameworks (including the Cairngorm Flex framework [Adobe, 2009]) so that addition of new functionality is fairly straightforward and requires minimal modification to the existing code base, allowing developers to focus on the problem at hand rather than on the integration of new functionality.

DEVELOPMENT AND ARCHITECTURE

Data Mapping, Assumptions and Derivations

The first step taken in the creation of LiveEnergy TRNSYS was to create an XML representation of the BUI file format. This was carried out manually as a paper exercise by examining existing BUI models and using the TRNSYS Type 56 documentation [Klein et al, 2007] as a reference document. After creating an XML representation of a BUI model, this was then translated into an XML Schema file labelled BUI XML.

The next step was to create a map between the BUI XML schema and the gbXML schema. A document was written containing a description of all elements in the BUI XML format and a solution for representing the information based on the data available in the gbXML standard. Figure 2 shows an example mapping between gbXML Surfaces and BUI Walls.

At this stage, necessary and optional elements in the BUI format were identified and assumptions were made about data required by the BUI format but not available in gbXML.

For instance, in the BUI format several properties are required which do not exist in gbXML. To populate these properties, sensible default values were assumed based on the TRNSYS Type 56 documentation [Klein et al, 2007] in order to get as close an approximation as possible. Subsequently, these values are hard coded into the resulting BUI file as default values. For example, air properties

(specific heat and density) are listed in BUI files and they were set to their standard values at 20°C.

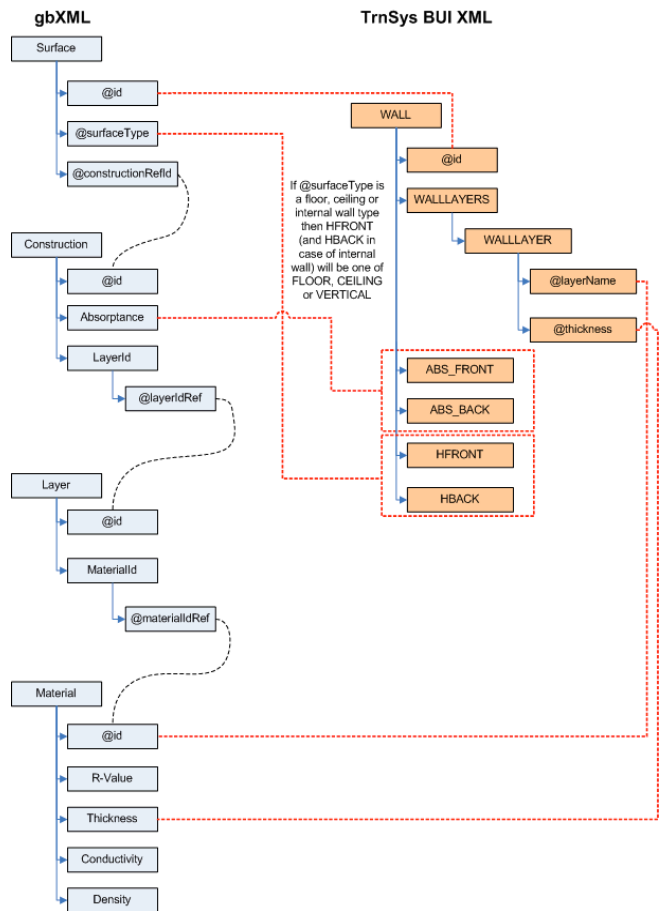


Figure 2 - Mapping between gbxml Surfaces and BUI Walls

Some of the information required by the BUI format is not directly available in gbXML but can be derived from the values that are. For example, wall representations in the BUI format require an area value while gbXML does not store the area of its surfaces. gbXML does represent width and height, however, so the area value can easily be derived.

Another example of a required derivation of gbXML values is in the case of Zone Capacitance. Zone capacitance is not represented in gbXML but is approximately derived from the gbXML values and assumed default properties using the following formula:

$$\text{air volume} \times \text{air density} \times \text{air specif icheat} \times \text{user value}$$

Where the constant user value is 5 in LiveEnergy TRNSYS (this multiplier accounts for thermal mass not directly modelled).

In a future version of LiveEnergy TRNSYS these default and derived values could be dynamically altered via the user interface.

Processing XML

After identifying the relevant mappings between gbXML and the BUI format, an XSLT style sheet was developed to convert a gbXML file into the BUI XML format based on the mappings and implementing any data derivations, assumed data etc.

A second XSLT style sheet was also created to process the BUI XML and output a valid BUI text file. The BUI XML schema closely follows the original BUI structure and naming schemes so that the second style sheet can export BUI XML elements directly as BUI commands without having to manipulate values.

Using the two style sheets, a gbXML file can be processed (using an XML processor) to output a TRNSYS BUI file that can be added to a TRNSYS model, or opened and edited in the TRNSYS TrnBuild tool. Figure 3 shows the process a user will take in converting a Google SketchUp model into a BUI file that can be executed in a TRNSYS simulation.

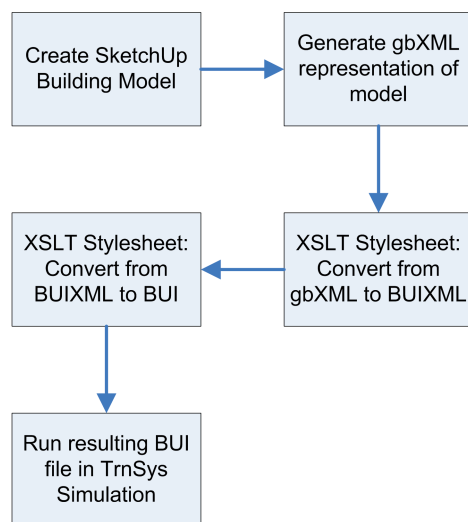


Figure 3 - Process diagram for converting a SketchUp model to TRNSYS BUI

Integration with LiveEnergy

The XSLT style sheets are the core elements in converting between gbXML and the TRNSYS BUI format and can be run manually using an XML processor.

In order to be used practically, the process of generating the BUI file from a Google SketchUp model has a requirement to be as straightforward and user-friendly as possible. With this in mind, the gbXML to BUI XML style sheets were integrated into the LiveEnergy application.

To process the style sheets, the SAXON [Kay, 2004] XSLT processor was utilised. For the requirements of LiveEnergy TRNSYS, the basic, open source, version of the SAXON processor was deemed adequate. The latest version of the processor features

a full implementation of the XSLT 2.0 standard [Kay, 2007], which has an extended function set allowing for more advanced mathematical operations and string processing than the XSLT 1.0 standard [Clark,1999]. This allows for the implementation of any data derivations and calculations within the XSLT style sheets and has the advantage of not requiring external helper applications to achieve this.

Ideally, an XSLT processing solution native to the core technologies used in LiveEnergy (Ruby, Adobe Flex and JavaScript) would have been used to process the style sheets. However, due to the unorthodox nature in which the Ruby environment is configured in Google SketchUp and due to the lack of support for XSLT in the version of Adobe Flex used, it would have been unnecessarily complex to enable a native XSLT processor.

The SAXON processor can be used in a standalone manner as either an executable .NET or Java program. This, along with the robust support for XSLT 2.0 made it desirable to make an external call from SketchUp to a bundled version of the SAXON processor. The main disadvantage of doing this was that the overall file size of the LiveEnergy TRNSYS plug-in increased dramatically due to the bundling of the SAXON processor. The use of the .NET version of SAXON means that LiveEnergy TRNSYS is only compatible with Microsoft Windows; the use of the Java version would have allowed LiveEnergy TRNSYS to be used on other platforms but would need Java to be installed as a pre-requisite. Since TRNSYS is currently only compatible with Microsoft Windows, the decision was taken to use the .NET version of SAXON. This can easily be changed at a later date to allow for a multi-platform version of LiveEnergy TRNSYS.

The original version of LiveEnergy was designed to interface with the web based Green Building Studio (GBS) energy analysis engine [Autodesk, 2009]. GBS features an intelligent defaults system where it accepts a gbXML file as an input and populates empty or missing values with defaults based on various conditions (e.g. building type, building regulations etc). For LiveEnergy, this meant that at a minimum only building geometry had to be constructed and sent to GBS.

When converting a SketchUp model to the TRNSYS BUI file format, construction and glazing information must be specified along with the relevant geometric and zone data. Ideally, information such as the building's usage schedules, infiltrations and gains would also be specified but were ignored in the initial version of LiveEnergy TRNSYS.

An option considered when developing LiveEnergy TRNSYS was to use the GBS platform to manage construction information, default values etc and return a populated gbXML file, which would be processed into the BUI format. It was desirable,

however, for LiveEnergy TRNSYS to not rely on having access to an internet connection or have a dependence on an external web application so it was decided to create a built-in constructions and glazing manager.

The LiveEnergy TRNSYS construction manager allows users to create their own specialised constructions to apply to surfaces or to choose constructions from various gbXML based construction libraries and apply these to their models. Construction library files based on gbXML can be dropped into a folder in the LiveEnergy program directory and will then be available for use in the application. For testing purposes, the SBEM [BRE, 2009] construction library was converted into the gbXML construction format (using XSLT style sheets) to be used in LiveEnergy and in TRNSYS upon exporting the model.

After implementing the core functionality, the next stage was to wrap the resulting BUI file in the TRNSYS DCK file format. A TRNSYS DCK file contains a description of all components within a TRNSYS simulation and can be executed by the TRNSYS engine. Normally, TRNSYS DCK files would be created with the TRNSYS Simulation Studio but can be generated externally as they are in a text based format.

DCK file support in LiveEnergy TRNSYS allows a user to specify various external parameters to build up a basic TRNSYS simulation that can be executed at the touch of a button from within LiveEnergy. For example, a user can specify which weather location the building is in, which variables to output from the simulation run etc.

RESULTS

Using LiveEnergy TRNSYS, a user can create a gbXML building representation and then successfully execute the model in TRNSYS.

LiveEnergy TRNSYS allows for the generation of a gbXML model from within Google SketchUp. This model can either have been created within Google SketchUp or can have been imported from other sources using LiveEnergy's gbXML import functionality.

LiveEnergy TRNSYS features the tools required to create and edit a gbXML file so that it can represent the information required by TRNSYS. This includes functionality to represent custom, project specific, construction information and access to pre-existing construction libraries.

LiveEnergy TRNSYS enables the user to quickly create a BUI or DCK file, which can then be loaded into the TRNSYS Simulation Studio or run through the TRNSYS engine from within LiveEnergy.

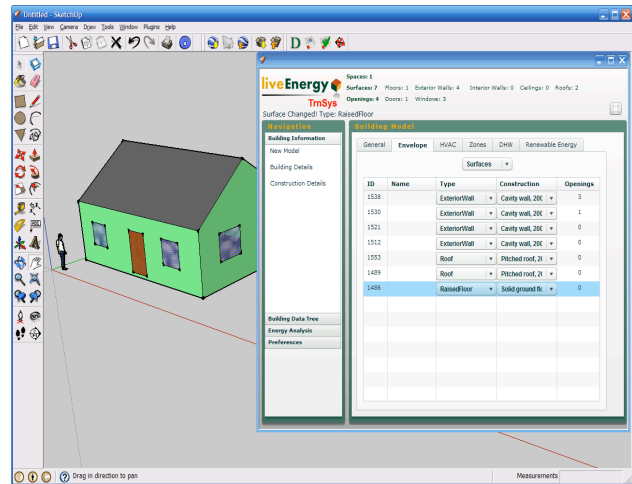


Figure 4 - LiveEnergy TRNSYS within Google SketchUp

LiveEnergy has no dependence on external applications or the existence of an internet connection, although LiveEnergy TRNSYS can use a user's TRNSYS installation to execute a DCK file, and if an internet connection is available then LiveEnergy TRNSYS can connect to a GBS server to utilise its services.

At the time of writing, LiveEnergy is fully compatible with versions 6 and 7 of Google SketchUp and creates BUI and DCK files that are compatible with version 16 of TRNSYS.

DISCUSSION AND FUTURE WORK

LiveEnergy TRNSYS creates a bridge between the gbXML building description format and the TRNSYS simulation environment. In effect, this means that a user can create a building model in any tool that can produce gbXML (including commonly used architectural tools such as Autodesk's Revit and Graphisoft's ArchiCAD) and use that model in TRNSYS.

LiveEnergy TRNSYS has the potential to expand the use of the TRNSYS simulation environment across user groups who would not currently consider it due to the steep learning curve. The ability to create and run TRNSYS simulations from within Google SketchUp with a simplified user interface masks the complexity of creating a full TRNSYS model and allows novice users to create simulations appropriate for a wide range of basic needs.

With the ability to export BUI files directly, LiveEnergy also has the potential to be of benefit to expert users by making the process of simulating building models more efficient. Using LiveEnergy TRNSYS, expert users can create, or use existing, rich building models within their TRNSYS simulations.

An example scenario could be where a client requests TRNSYS simulations featuring a portfolio of

buildings of which CAD models are available. In this case, rather than create TRNSYS building representations from scratch the TRNSYS expert could simply use LiveEnergy TRNSYS to export the models as BUI files and then drop these into their simulation, allowing them to focus on other critical simulation concerns.

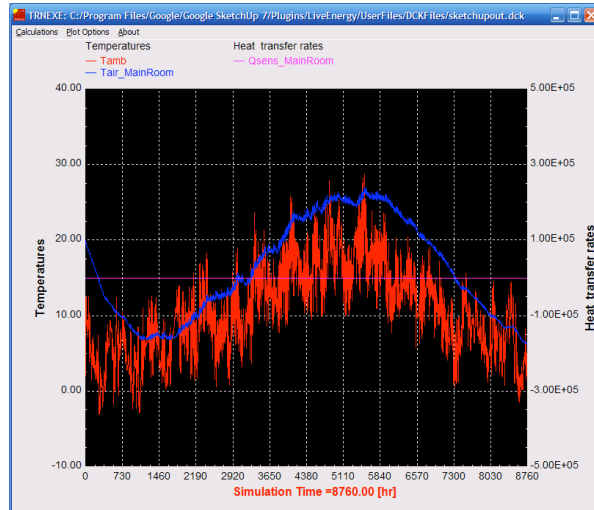


Figure 5 – Results from a TRNSYS simulation called directly from LiveEnergy TRNSYS

There is considerable scope for future extension in both LiveEnergy and LiveEnergy TRNSYS. Support for more advanced parameters such as HVAC systems within a building model could be implemented. Support for creating more complex TRNSYS simulations from within LiveEnergy and the ability to add more TRNSYS components to the exported TRNSYS DCK file could be added.

Various features to improve the LiveEnergy TRNSYS user experience could be implemented such as the ability to automatically identify and zone rooms, surfaces etc with minimal human input. Various construction libraries can also be built from existing sources and utilised within LiveEnergy TRNSYS, this could lead to the creation of online construction repositories, or create a market for the sale of third party construction libraries.

A major limitation with the current version of LiveEnergy TRNSYS is that if a user wants to use the resulting BUI or DCK files then they must have a copy of TRNSYS installed. There could be a potential here for the implementation of a web-based version of TRNSYS - available through web services so that applications like LiveEnergy TRNSYS could connect and use the TRNSYS engine (e.g. on a per run or subscription basis). Such a service would open out TRNSYS to a wider audience and would also allow users of different platforms to use the TRNSYS engine remotely (e.g. TRNSYS is currently a Microsoft Windows only application, but Google SketchUp and LiveEnergy run on both Microsoft Windows and Apple OS X). It would be down to the

TRNSYS developers to licence and allow for the implementation of such a web-based version of TRNSYS.

CONCLUSION

This paper has described the LiveEnergy TRNSYS plug-in for Google SketchUp, which facilitates the conversion of gbXML models into a format compatible with the TRNSYS simulation engine, consequently allowing the use of SketchUp and other CAD models in TRNSYS simulations.

LiveEnergy TRNSYS can be of benefit to both novice and expert users. The simplicity of the Google SketchUp environment and the ability to include default TRNSYS simulation models surrounding the user's building model means that the complexity of creating a TRNSYS simulation is masked in a user-friendly manner. This allows the user to access the power of TRNSYS with minimal effort. LiveEnergy also allows expert users to focus on specific complexities outwith the building model and allows them to easily use multiple large and complex buildings in their simulations within a feasible time frame.

LiveEnergy TRNSYS combines the power of the TRNSYS simulation engine with the ease of use of the Google SketchUp environment and the flexibility of gbXML.

ACKNOWLEDGEMENT

This work has been carried out as an extension to the LiveEnergy project [Finlayson et al, 2008] which was supported by the European Regional Development Fund, Comhairle nan Eilean Siar and Highlands and Islands Enterprise.

Grateful acknowledgment is extended to the Greenspace Research software development team and to Adam Jarvie at Strathclyde University who assisted with the development of LiveEnergy TRNSYS.

REFERENCES

- Anon, TRNSYS - TESS : thermal energy system specialists. Available: <http://www.tess-inc.com/trnsys> [15/10/2008, 2008].
- ADOBE, 06/02/2009, 2009-last update, about cairngorm [Homepage of Adobe], [Online]. Available: <http://opensource.adobe.com/wiki/display/cairngorm/About> [06/02, 2009].
- ADOBE, 29/01/2009, 2009-last update, adobe flex 3 [Homepage of Adobe], [Online]. Available: <http://www.adobe.com/products/flex/> [29/01, 2009].
- AUTODESK, 29/01/2009, 2009-last update, autodesk green building studio [Homepage of Autodesk],

- [Online]. Available: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=11179508> [29/01, 2009].
- BRAY, T., PAOLI, J., SPERBERG-MCQUEEN, C.M., MALER, E. and YERGEAU, F., 2006. Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Recommendation. W3C.
- BRE, 29/01/2009, 2009-last update, BRE: simplified building energy model [Homepage of BRE], [Online]. Available: <http://www.bre.co.uk/energyrating/page.jsp?id=694> [29/01, 2009].
- CLARK, J., 16/11/1999, 1999-last update, XSL transformations (XSLT) version 1.0 [Homepage of W3C], [Online]. Available: <http://www.w3.org/TR/xslt> [06/02, 2009].
- CLARK, J. and DEPOSE, S., 1999. XML Path Language (XPath) Version 1.0. W3C Recommendation. W3C.
- CRAWLEY, D.B., HAND, J.W., KUMMERT, M. and GRIFFITH, B.T., 2008. Contrasting the capabilities of building energy performance simulation programs. *Building and Environment*, 43(4), pp. 661-673.
- DIAG, european energy performance of buildings directive. Available: <http://www.diag.org.uk/key-information/key-documents-.aspx> [24/04/2009, 2009].
- FINLAYSON, N., MURRAY, M., MACIVER, D., MORRISON, J., MORRISON, M., MACDONALD, D.I., MACASKILL, D., MACRITCHIE, D. and RIZOS, I., 2008. Demeter - A Green Building Plugin For SketchUp. SIMBuild 2008 - Third National Conference of IBPSA-USA.
- GOOGLE, 02/01, 2009-last update, google SketchUp [Homepage of Google], [Online]. Available: <http://sketchup.google.com> [29/01, 2009].
- KAY, M., 18/01/2007, 2007-last update, XSL transformations (XSLT) version 2.0 [Homepage of W3C], [Online]. Available: <http://www.w3.org/TR/xslt20/> [29/01, 2009].
- KAY, M., 11/06/2004, 2004-last update, saxonica: XSLT and XQuery processing [Homepage of Saxonica], [Online]. Available: <http://www.saxonica.com/> [29/01, 2009].
- KENNEDY, J.F., 3/09/2008, 2008-last update, gbXML schema . Available: <http://www.gbxml.org/schema.htm> [10/14, 2008].
- KLEIN, S.A., et al., 2007. TRNSYS 16 – A Transient System Simulation Program. University of Wisconsin-Madison Solar Energy Laboratory, Madison, WI, USA.
- LE HORS, A., LE HÉGARET, P., WOOD, L., NICOL, G., ROBIE, J., CHAMPION, M. and BYRNE, S., 2004. Document Object Model (DOM) Level 3 Core Specification. W3C Recommendation. W3C.
- MARSHALL, A.C., 09/09/1997, 1997-last update, modular programming with fortran 90 [Homepage of University of Liverpool], [Online]. Available: <http://www.liv.ac.uk/HPC/HTMLF90Course/HTMLF90CourseSlides.html> [06/02, 2009].
- McDowell, T.P., Bradley, D.E., Thornton, J.W. and Kummert, M., 2004. SIMULATION SYNERGY: EXPANDING TRNSYS CAPABILITIES AND USABILITY, IBPSA-USA National Conference Boulder, CO, 2004.
- PELLETRET, R. and SOUBRA, S., 1990. IISIBat project - intelligent front-end shell for building simulation tools. VTT Symposium (Valtion Teknillinen Tutkimuskeskus), (118), pp. 159.
- RUBY-LANG WEBMASTER, 29/01/2009, 2009-last update, about ruby. Available: <http://www.ruby-lang.org/en/about/> [29/01, 2009].
- SCHELL, B., ESCH, J.L. and ULMER, J.E., 2003. System and Method For Three Dimensional Modelling. 345/420 edn. Colorado/USA: G06T 17/00.
- SCHNEIDER, J., YU, R. and DYER, J., 2005. ECMAScript for XML (E4X) Specification . 2nd edn. Rue du Rhône 114 CH-1204 Geneva: ECMA International.
- SEL, 14/04/2009, 2009-last update, official TRNSYS website. Available: <http://sel.me.wisc.edu/trnsys/> [28/04/2009, 2009].
- SEL, 2007. TRNSYS 16 – A Transient System Simulation Program – Volume 6, Multizone Building modelling with Type56 and TRNBuild. University of Wisconsin-Madison Solar Energy Laboratory, Madison, WI, USA.
- THERMAL ENERGY SYSTEM SPECIALISTS, 26/10/2007, 2007-last update, simulation studio: the graphical interface for TRNSYS [Homepage of Thermal Energy System Specialists], [Online]. Available: <http://www.trnsys.com/studio.htm> [10/14, 2008].
- THERMAL ENERGY SYSTEM SPECIALISTS, 11/7/2007, 2007-last update, TRNSYS the transient energy system simulation tool [Homepage of Thermal Energy System Specialists], [Online]. Available: <http://www.trnsys.com/> [10/10, 2008].
- TRANSSOLAR, , TRANSSOLAR | KlimaEngineering. Available: <http://www.transsolar.com/> [24/04/2009, 2009].
- VAN DER VLIST, E., 2002. XML Schema . 1st edn. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media Inc.