Building Model Checker for Automated Code Compliance

Bikramjit Singh, Narinder Singh, Hardeep Singh Rai, Satinder Preet Kaur

Abstract- This paper presents an automated code checking system Pristine that enables quick and easy compliance assessment and assists designers in finding potential problems early. There has been an extensive amount of research conducted internationally in the area of automated code compliance for the Architecture, Engineering and the Construction (AEC) industry. There is an increased uptake of building information modeling (BIM) and the Industry Foundation Classes (IFC) open standard data model for interoperability due to a recent productivity improvement and innovation in the AEC industry. The availability of high performance personal computers, efficient web-based technology, and new initiatives in legal knowledge representation modeling should make the development of commercial compliance checking systems more viable than ever. However, the quest for an industry agreed unified approach seems to be far from over. The system allows for checking of model for different model attributes using the ifcXML schema which is created using either ArchiCAD or Revit architectural soft wares. Once the checking is completed, the interactive reporting interface offers a viewing option of the validated file.

Keywords- IFC, BIM, Automated Code Compliance, J2SE.

I. INTRODUCTION

We live in a built environment designed around regulations to ensure our safety and well-being. A building is subjected to a cycle of regulatory compliance assessments throughout its entire life and building designers ensure that every aspect of their design adheres to various regulatory requirements.

- During the **approval process**, the design is subjected to formal analysis by the consent processing authority.
- During **construction and commissioning**, every building component is then checked before and after installation to ensure that the quality of products and workmanship conforms to the specified standards.
- The **facility management** of a building also requires regular compliance checking to ensure that the building is used and maintained as required and as designed.
    - At the **demolition stage**, compliance checking is important to ensure safety of occupants in the neighboring buildings and to protect the surrounding environment during the work.

Manuscript Received on February 2015.

Er. Bikramjit Singh, A.P. Department of Civil Engineering, Khalsa College of Engineering & Technology, Amritsar, India.

Er. Narinder Singh, A.P. Department of Civil Engineering, Khalsa College of Engineering & Technology, Amritsar, India.

Prof. Hardip Singh Rai, Guru Nanak Dev Engineering College of Engineering & Technology, Ludhiana, India.

Er. Satinder Preet Kaur, Department of CSE, Khalsa College of Engineering & Technology, Amritsar, India.

Legislation requires the construction industry to perform a complex task of checking building designs for compliance against numerous building codes. This may result in high, long-term costs if there is a failure to correctly assess designs for compliance can. For example, in a large scale housing project in south London (2003), the wheelchair ramps were found to be too steep and narrow and cost £800,000 in construction and design changes. Thus, an automated code checking software tool is needed by the construction industry to enable designers to identify potential faults earlier.

The study of code compliance checking has had a long history of development (Gero,
1982; Rosenman et al, 1986; Balachandran et al, 1991; Fenves et al, 1995;
Drogemuller et al, 2000; Woodbury et al, 2000; Maissa et al, 2002; Ding et al, 2004).

However, the major drawback is that there are only fewer applications for use in the construction industry. The obstacle to more widespread use in construction industry lie in the lack of common models to integrate building codes with various application environments, object-based representations of building codes to support sophisticated and less complex computation and reasoning and support for use of design standards during the design process. Industry Foundation Classes (IFCs) provide a common standard for data interoperability and have been used as a common model in architecture, engineering and construction domains. It provides object-based rule bases and is therefore an ideal platform for encoding building codes and linking them with building models. Singapore developed the e-PlanCheck system (Solihin, 2004) that uses the IFC model and provides code compliance assessment and acts as an internet-based application or standalone application. However, in e-PlanCheck, support for use of design standards at different stages of design during the design process is not provided. Another such system is the Solibri Model Checker (Solibri, Inc.), developed in Finland that also uses the IFC model and focuses on ‘design-spell-check’. Due to a restricted range of objects and parameters for encoding building codes and domain knowledge, Solibri Model Checker is restricted in its application to code compliance checking. This paper presents an automated code checking system – Pristine, for Construction Innovation and currently under improvement in India. The Pristine system develops an internal model based on IFCs for model checking according to Indian building code. The advantages in the Pristine system beyond an automated code checking process, flexibility by allowing a design to be checked by selected clauses of design during the design process.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

II. METHODOLOGY

In the intended system, the compliance checking component would read an IFC-based building model and check the model against a selected set of open standard constraints. This is done by parsing the IFC based building model on the Java platform into objects. A common approach to automated compliance checking is systematic comparison that is comparing each object or system in a building model representation with the constraints in a standard. The returned result is then represented as a report to provide a user-level view of code compliance.

1.1 Building Information Modeling and pristine

The process of automating code checking requires an adequate building model to begin with. Information currently provided within the object-based CAD model and the IFC model is inadequate for many building code requirements. This section illustrates the requirement and method of constructing better building models in object-based CAD systems using IFCs and the development of a Pristine system. The Pristine system uses ArchiCAD as the object-based CAD system since ArchiCAD supports building information modeling and distinguishes itself as an information-rich architectural CAD tool rather than a drafting tool. Figure 1 shows a 3D object-based model produced in ArchiCAD that is tested for Pristine. Elements, properties and relationships of elements are the distinguishing features in the IFC model. Rich relationships of elements enable meanings between elements to be identified. For example, IfcRelSpaceBoundary provides the bounds relationship between a Space and a building the IFC model and displays the IFC attributes and property sets of selected objects. Designers can select an element and then define the attributes and properties associated with building codes in the IFCTreeView as it is particularly useful since
information associated with building codes to be inputted and modeled. IFCTreeView lists the element mappings between the CAD model and

Figure 1: A 3D rectangular tank model produced in ArchiCAD

Customizing object properties and IFCTreeView are two existing approaches available in ArchiCAD that allow extended design. It allows designers to easily specify additional properties compatible with the IFC model. Element such as a Door. This enables the spatial relationship between an entrance Space and an exterior Door to be identified in order to check an accessible entrance for disabled people. Examples of the relationship mappings, supported by ArchiCAD and mostly needed by the code compliance checking, are listed as follows:

• IfcBeam – IfcBeam defines the occurrence of any beam, common information about beam types (or styles) is handled by IfcBeamType. The IfcBeamType (if present) may establish the common type name, usage (or predefined) type, common material layer set, common set of properties and common shape representations (using IfcRepresentationMap). The IfcBeamType is attached using the IfcRelDefinedByType.RelatingTypeobjectified relationship and is accessible by the inverse IsDefinedBy attribute.

• **IfcSlab** – The IfcSlab defines the occurrence of any slab, common information about slab types (or styles) is handled by IfcSlabType. The IfcSlabType (if present) may establish the common type name, usage (or predefined) type, common set of properties, common material layer set, and common shape representations (using IfcRepresentationMap). The IfcSlabType is attached using the IfcRelDefinedByType.RelatingTypeobjectified relationship and is accessible by the inverse IsTypedBy attribute.

• **IfcColumn** –IfcColumn defines the occurrence of any column, common information about column types (or styles) is handled by IfcColumnType. The IfcColumnType (if present) may establish the common type name, usage (or predefined) type, common material layer set, common set of properties and common shape representations (using IfcRepresentationMap). The IfcColumnType is attached using theIfcRel Defined By Type.Relating Type objectified relationship and is accessible by the inverse Is typed By attribute.

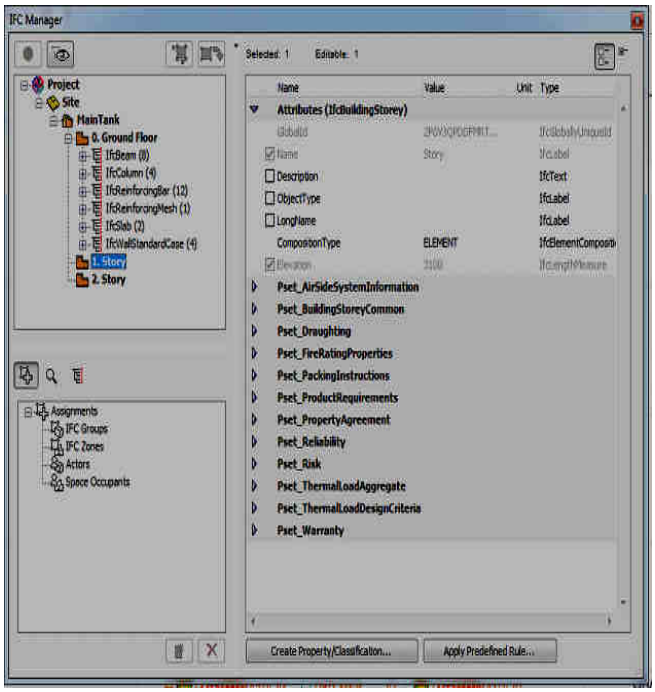

Figure 2: IFC Tree View allows users to define extended properties required by building codes

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

• **IfcWall** – IfcWall (or the subtype IfcWallStandardCase) defines the occurrence of any wall, common information about wall types (or styles) is handled by IfcWallType. The IfcWallType (if present) may establish the common type name, usage (or predefined) type, common material layer set, common set of properties and common shape representations (using IfcRepresentationMap). The IfcWallType is attached using the Ifc Rel Defined ByType.RelatingType objectified relationship and is accessible by the inverse Is Defined By attribute.However, the existing IFCTreeView in ArchiCAD is restricted in customizing enriched element and relationship mappings onto the IFC model. The existing element mappings cover Building, Building Storey, Space, Wall, Door, Stair, etc., but mappings onto Stair Flight and Ramp Flight are not available.

1.2 Mappings between ArchiCAD Model and IFC Model to Pristine System

The building model produced in object-based ArchiCAD systems is exported to the IFC model and then mapped onto the Pristine internal model for compliance assessment. A mapping schema is required to facilitate automated translation from the IFC model to the internal model. The mapping schema is implemented using the ExpressG language. ExpressG contains mapping specific functions which allow users to efficiently convert models. The mapping schema for the Pristine system is well structured and can be readily modified and extended in future.

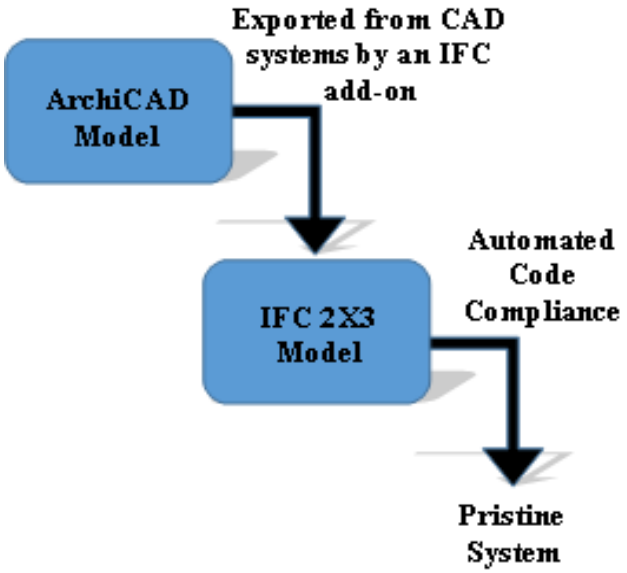

Figure 3: A process of mapping from the ArchiCAD model to the IFC2x3 model and then to the Pristine

III. SYSTEM ARCHITECTURE

The architecture of the Pristine system is illustrated in Figure 4. It consists of three main components: main user interface, code compliance system and the report system.

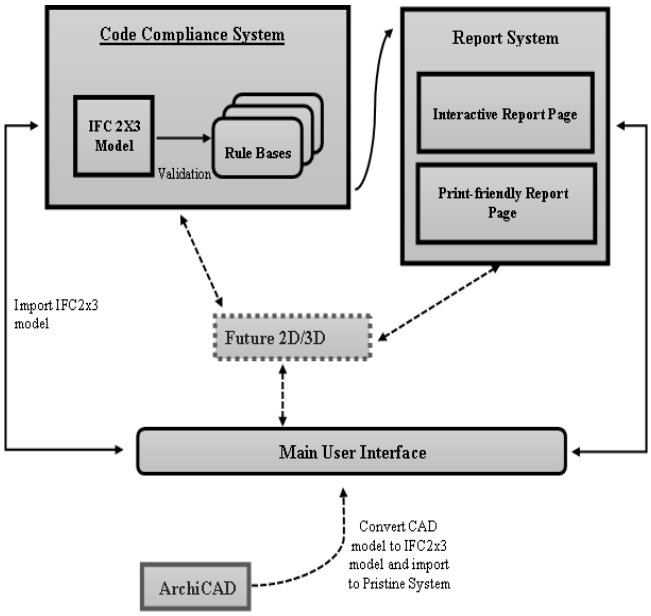

Figure 4: Architecture of the Pristine system
The Pristine system runs as standalone software.

1.3.1 Main User Interface

The building model created in object-oriented ArchiCAD systems is exported as an IFC2x3 model and then imported to the Pristine system for compliance checking. If it is required, a direct interface to object-oriented ArchiCAD systems could be developed in future.

1.3.2 Code Compliance System

It is the core component of the Pristine system. The model is validated against rules in the rule bases. The rules encode object-based interpretations and performance requirements from building codes. This validation files is then created to store the check results.

1.3.3 Report System

It has a direct interface to the Code Compliance System. It reads the check results from, and writes the specifications/comments to the validation file. The report system provides both an interactive report page and a print friendly report page. Once the checking is completed an interactive report page appears to the user, which offers a variety of viewing options and enables the user to view results according to the constraints used. The interactive report page links to a print-friendly report page that allows designers to list all details in the report and print it out. A 2D/3D model viewer, shown in Figure 4 by dashed lines, will be integrated with the Pristine system in future. It will provide a visualization of the building model and allow problem elements to be highlighted. Examples of the relationship mappings, supported by Archi CAD and mostly needed by the code compliance checking, are listed as follows:

IV. SYSTEM IMPLEMENTATION

The implementation of the Pristine system is illustrated in Figure 5. The main user interface is a Java Standard Edition (J2SE) based. It allows users to monitor the information flow commencing from importing building models to reporting check results. The interactive report page and print-friendly

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

report page are implemented in Java as well and can be represented as .htm file, .doc file, etc.

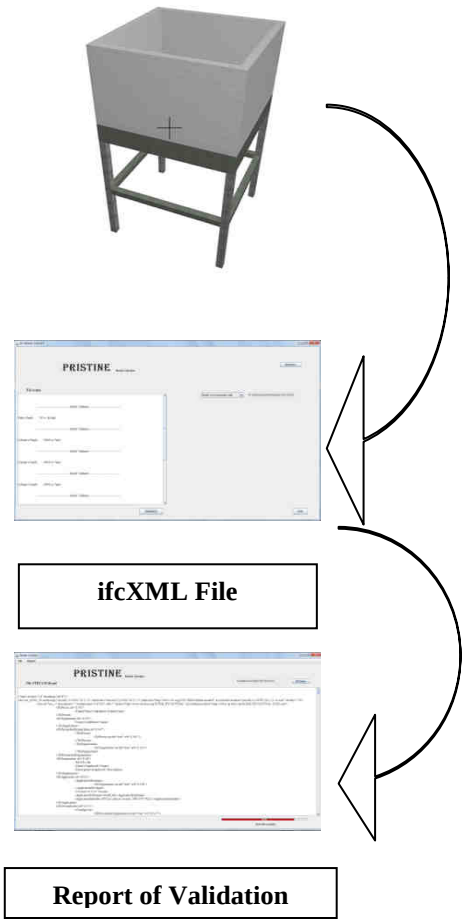

Figure 5: Implementation of the Pristine system.

The main user interface is divided into two sections. One to display the imported ifcXML file and the second called the 'Validation' to implement automated code compliance which is connected to the Code Compliance System as a part of it. Figure 6 illustrates the working of Pristine system.

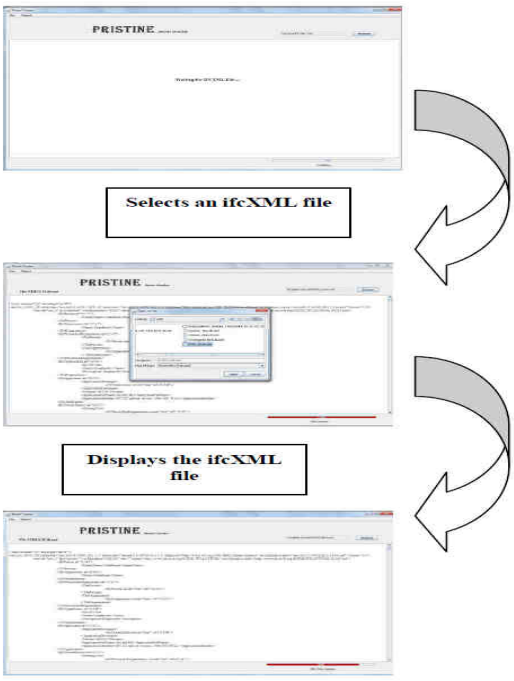

Figure 6: Illustration of main user interface in Pristine system

The Code Compliance System as already discussed provides the functionality of automated code compliance. It creates a validated file which in the .doc, .htm, .bim, or any other file format as per user requirements. Figure 7 shows how an IFC xml file is selected and how using various checks the validation file is created using code compliance.

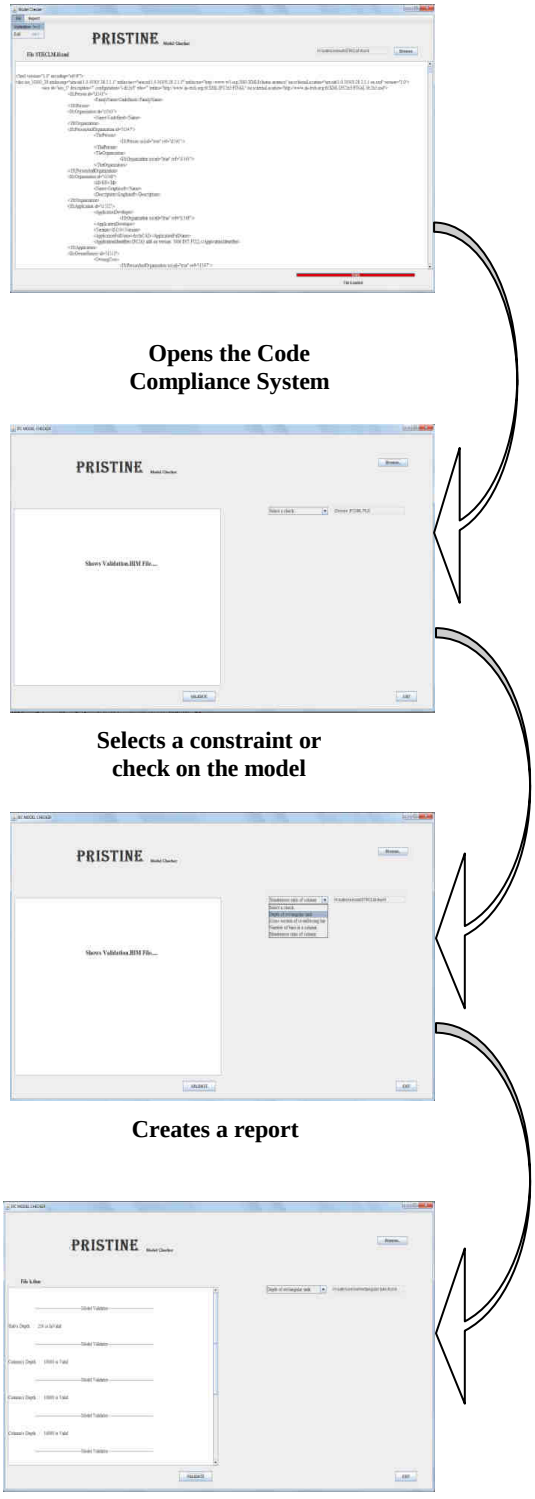

Figure 6: Illustration of code compliance sub-system in Pristine system

The Report System is another part of the Pristine system that is used to read the validated file or produce a printed document of the validated file.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

V. CONCLUSIONS AND FUTURE

The development of the Pristine system uses an efficient platform and provides functionalities towards industry needs. As an advanced software tool, the Pristine system will reduce the risk of non-compliance with its associated rectification costs and significantly improve the efficiency in the building code checking process. Direct benefits to architects, designers, building consultants and engineers can be gained from Pristine. The benefits include:

- Automating the design checking process for compliance with building codes.
- Providing more reliable assessment with fewer errors.
- The ability to interrogate 3D object-based CAD systems.
- Allowing the checking of a design by selected building object types.
- Providing a friendly and interactive reporting system.

Pristine is currently being tested for validation and feedback. Future development of the Pristine system relating to the interest areas in both research and practice includes: the development of a consistent manner for building code interpretation such as using decision tables, the development of semantic models and expert knowledge, and system improvement including the development of structured specification to allow users to input specification easily and the integration with a 2D/3D model viewer. Collaboration with CAD vendors is required to enable the CAD-IFC interface to be improved to support automated code checking application.

REFERENCES

[1] Johannes Dimyadi, Robert Amor, "Automated Building Code Compliance Checking-Where is it at?"
[2] James Nyambayo, Robert Amor, Ihsan Fraj and Jffrey Wix, "External Product Library System- An Implementation of the Industry Foundation Classes 2.0 Library Model"
[3] IAI(1999a) IFC Object Model Architecture Guide, Vol3: IFC Object Model Reference, IFC Release 2.0 Documentation 1999
[4] Ding, L., Drogemuller, R., Roseman, M., Marchant, D. & Gero, J. (2006). Automated code checking for building designs- DesignCheck.
[5] Balachandran M., Roseman M.A. and Gero J.S. (1991) A knowledge-based approach to the automatic verification of designs from CAD databases, in Gero J.S. (ed.), *Artificial Intelligece in Design '91*, Butterworth-Heinemann, Oxford, pp. 757-781.
[6] Ding L., Drogemuller R., Jupp J., Rosenman M. A. and Gero J. S. (2004) Automated code checking, CRC CI International Conference 2004, Gold Coast.
[7] Solibri, Inc. http://www.solibri.com
[8] IAI (2004) IFC 2x2 Edition 2 Addendum 1, IAI.
[9] Graphisoft (2001) ArchiCAD IFC Reference Guide, Version 1.0, Graphisoft.
[10] IAI (1999b)I FC Specifications Development Guide: Appendix F, IFC Properties and Property, IFC Release 2.0 Documentation 1999

Er. Bikramjit Singh B.tech civil, Master in computer,M.tech Structure(persuing),paper published in IGC dec 13-15,2012, research work on BIM Technology,Presently working for 5 years with khalsa college of engineering & Technology, Amritsar with Past Experience of 10 years field as well as research

Er. Narinder Singh B.tech civil, PGDCA,M.tech Soil,paper presented & published in IGC dec 13-15,2012, research work on BIM Technology,Presently working for 4 years with khalsa college of engineering & Technology, Amritsar with Past Experience of 15 years field as well as research

Prof. Hardeep Singh Rai B.tech civil, ,M.tech Structure, Ph.D Civil more than 100 paper published, research work on BIM Technology,Presently working with Guru Nanak Dev college of engineering & Technology, Ludhiana with Experience as Dean Consultancywith field as well as research

Er. Satinder Preet Kaur B.tech CSE(Persuing) , from khalsa college of engineering & Technology, Amritsar

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.