CONSISTENT CAD - FEM MODELS ON THE BASIS OF OBJECT VERSIONS AND BINDINGS

A.H. Olivier[1], K. Beucke[2], B. Firmenich[3], and G.C. van Rooyen[4]

ABSTRACT

This paper focuses on the sharing of information between CAD (Computer-Aided-Design) and FEM (Finite Element Method) applications. There are two different approaches in integrating CAD and FEM information. The first approach would use one product model that contains both the CAD and FEM information (e.g. IFC). This approach leads to complicated models that are difficult to maintain.

The second approach would use domain specific models and duplicate data that is shared by related models, e.g. geometrical information. Specialized, standalone software is used to analyse certain aspects of the final product. Data is transferred between these specialized software applications via standardized interfaces e.g. DXF or XML.

This paper investigates the second approach. The aim is to briefly introduce problems that could occur when data is shared between different specialized applications and to discuss possible solutions.

An approach of bindings (links) between CAD and FEM objects is introduced to assist in keeping information consistent between separate but dependent models.

KEY WORDS

CAD, FEM, consistent models, distributed work, bindings.

INTRODUCTION

INFORMATION EXCHANGE

In everyday engineering practice it is a straightforward task to convey information from a CAD model to an analysis model. This information sharing is made possible via standard data exchange interfaces e.g. DXF. Usually some additional information is added to the FEM model, e.g. material attributes and boundary conditions. Should a change occur in the original CAD model, it might or might not affect the numerical analysis, and it is the

[1] Dept. of Civil Engineering, University of Stellenbosch, Banhoek Road, Stellenbosch, South Africa, Phone +27(0)21/808-4437, FAX +27(0)21/808-4947, bertie@sun.ac.za / bertie.olivier@bauing.uni-weimar.de

[2] Fakultät Bauingenieurwesen, Bauhaus-Universität, Coudraystraße 7, D-99423 Weimar, Germany, Phone +49(0)3643/58-4214, FAX +49(0)3643/58-4216, karl.beucke@informatik.uni-weimar.de

[3] Fakultät Bauingenieurwesen, Bauhaus-Universität, Coudraystraße 7, D-99423 Weimar, Germany, Phone +49(0)3643/58-4230, FAX +49(0)3643/58-4216, berthold.firmenich@informatik.uni-weimar.de

[4] Dept. of Civil Engineering, University of Stellenbosch, Banhoek Road, Stellenbosch, South Africa, Phone +27(0)21/808-4437, FAX +27(0)21/808-4947, gcvr@sun.ac.za

engineer's responsibility to adapt the numerical model to be consistent with the latest CAD model.

If all engineering tasks were done only once, and no changes ever occurred, the use of a data exchange format to transfer information from one application to another would be sufficient. However, once changes occur in the source data set, as it frequently does in engineering practice, it is a tiresome and error-prone task to determine their effect in the target application since there is no intelligent connection between source and target. This task may be so complicated that it may be easier to discard the existing numerical model and start over from the beginning. When the exchange of information between CAD and FEM applications only takes place at file level, it is not possible to calculate change-effects downstream, and the situation described above arises.

In this paper an approach of binding definitions between CAD and FEM models is introduced. The bindings assist an engineer in keeping FEM model information consistent with an evolving CAD model.

THE CONCEPT OF BINDINGS AND VERSIONS

A binding represents a dependency between two objects (Pahl and Beucke 2000). In a versioned object model, a binding would describe a dependency between two object versions (Firmenich 2002).. Thus, if a new version of the binding object (primary) version is derived, a graph calculation indicates that the bound (dependent) object depends on an outdated object. The user is then responsible for updating the bound object and establishing a new binding if the change affects the bound object (operation modernize). If the change to the bound object version does not affect the binding object version, then the user could freeze the binding. Freezing the binding would allow an object to be bound to an earlier version of a specific object (Beer and Firmenich 2003)

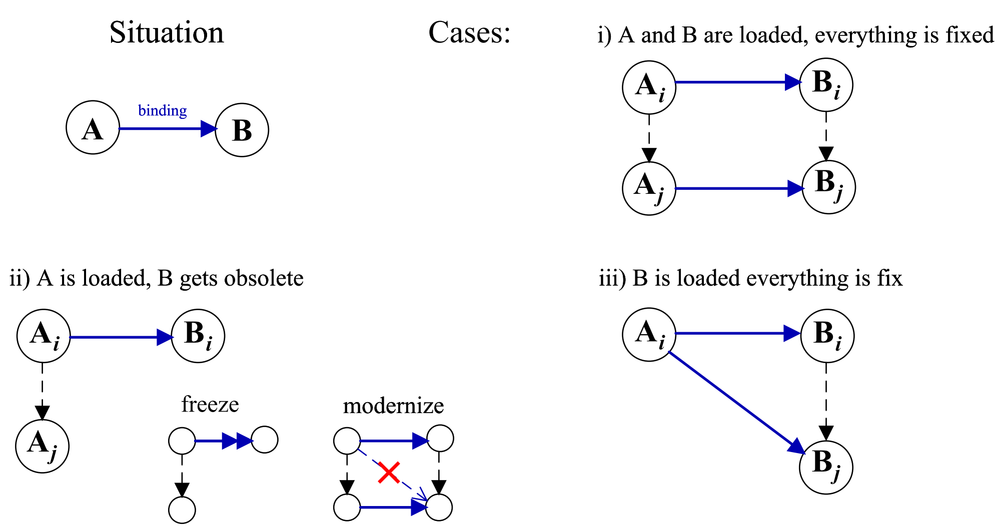

Figure 1: Bindings between versioned objects

In figure 1 object A represents a CAD object and object B represents a FEM object. Case i allows a user to change the CAD and FEM object simultaneously and to establish a new binding. In the second case, the CAD object is changed. The FEM object is now obsolete because it depends on an outdated CAD object. On the FEM side, the user could either update the FEM object (operation modernize) or allow the FEM object to be bound to an outdated version of the CAD object (operation freeze). The third case shows that two FEM version may depend on one CAD version.

An application of the above approach would be the creation of semantic bindings. Semantic bindings are objects that know the contents of related object versions and when the contents change, an inconsistent state between dependent object versions can be identified. Thus, the bindings would be intelligent objects that bind object versions based on contents.

COMMON GROUND BETWEEN CAD AND FEM

GEOMETRY

The first step towards model consistency is to define the common ground between the models. The geometry of a structure is described by a CAD model. The CAD model could either use objects with building semantics e.g. beams, slabs and columns or simpler geometry objects like lines and arcs. A FEM model describes the structural behaviour of a structure given the geometry, material properties and loading as input. In both models, geometry forms an integral part of the modelling process.

GEOMETRIC DEPENDENCIES

The mapping of geometry between CAD and FEM models is not a one-to-one mapping. For example, the CAD model may describe a beam as two flanges and a web, while the centre line of the beam and the cross-sectional properties are the only important geometric properties from the perspective of the FEM analysis. Thus, a thorough understanding of the modelling differences is important in defining useful bindings between CAD and FEM geometry.

If a structure is described by lines in a CAD system, the lines do not share end points. Each line would have its own start and end point, and lines are considered connected when two distinct points have the same coordinates. However, in FEM analysis two beam elements should share one node if the beams are connected in reality (unless constraint equations are used to connect the two beams).

CLASSES OF GEOMETRIC DEPENDENCIES

In both CAD and FEM models, geometry may be encapsulated in geometry-type objects, or it may be contained as attribute values of more general objects. Consequently four classes of geometric dependencies exist.

Object – Object dependency

This is the simplest class of geometric dependency. An example would be a FEM Node-object that depends on a CAD Point-object (figure 2a).

Object – Attribute dependency

This dependency exists between a FEM object and an attribute of a CAD object. An example would be a FEM Node-object that depends on the coordinates of an end point of a CAD Line-object (figure 2b).

Attribute – Object dependency

This dependency exists between an attribute of a FEM object and a CAD object. An example would be the thickness of a FEM Element-object that depends on a text description in CAD (figure 2c).

Attribute – Attribute dependency

This dependency exists between attributes of a FEM object and a CAD object. An example would be the rotation angle of a local coordinate system that is dependent on the angle of a line. The angle of the line could either be an attribute or a derived property (figure 2d).

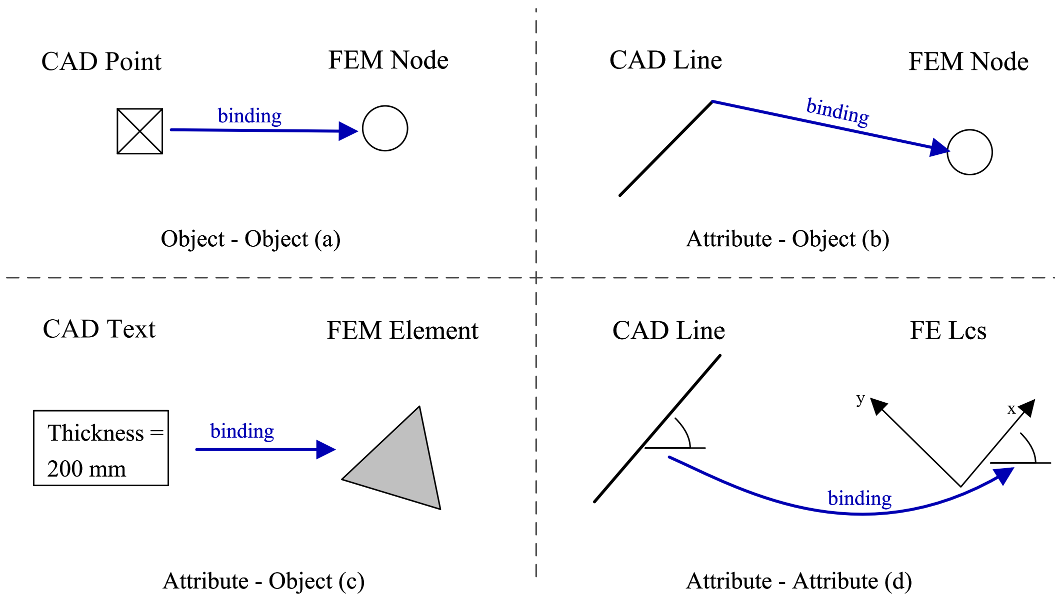

Figure 2: Classes of Geometric dependencies

These dependencies could exist in 1:n, n:1 and m:n relationships.

SEPARATION OF CAD AND FEM MODELS

Certain properties of the FEM model, e.g. support conditions, material properties and loading do not belong to the CAD domain although it can be represented graphically. The aim of the approach presented in this paper is to maintain clear boundaries between CAD and FEM models. This requirement demands that properties that belong to the analysis part of the problem should not be specified or controlled by the CAD model. If the boundary between

CAD and FEM applications is not clearly defined the end result would be an integrated application that operates on one model.

COMPLEXITY

In order to discuss the complexity of defining and maintaining bindings, and reacting to information propagated by bindings, a simple roof truss is considered:

EXAMPLE 1: CAD TRUSS vs. FEM TRUSS

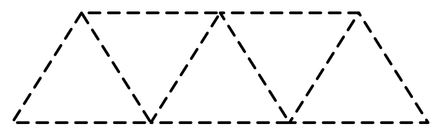

Figure 3: Simple roof truss example

CAD model of truss

Two approaches can be followed to model the truss in a CAD environment. In both approaches the graphical representation of the truss is exactly the same, while the underlying structure of the CAD model is completely different:

- The truss is drawn using lines and some text to describe the member sizes. In this CAD environment, there only exists geometric objects e.g. lines. (traditional drafting tool)
- The truss is represented as a roof truss object in the CAD environment. The roof truss object is an aggregation of objects that map a real roof truss to its CAD model.

FEM model of truss

The FEM model of the truss is an aggregation of FEM objects like Nodes, Elements, Materials, Supports, DOFs (Degrees of Freedom) and Loads. An Equation is formed that represents the FEM system equations.

Binding of CAD and FEM trusses

Given the CAD truss and the FEM truss objects, it is possible to create a binding object that connects a specific version of the CAD truss to a specific version of the FEM truss. Should a truss element be added or moved, a new CAD version of the truss would be created. The binding object between the CAD and FEM trusses would show that the FEM truss is bound to an outdated CAD object. The engineer would have to either update the FEM truss and create a new binding between the two new versions of the CAD and FEM truss objects (operation modernize), or if the change does not affect the structural behaviour of the FEM truss, the engineer would have to freeze the binding to verify the link between the FEM truss to the old version of the CAD truss (operation freeze).

This approach would be sufficient if the complexity of the objects on either side of the binding layer is not very high i.e. an engineer would be able to recognize changes without any additional support from the user interface. Once the complexity of objects on either side of the binding layer is too high, it is not sufficient just to know that there is a new version of the binding object. One also needs to know exactly what was changed in order to make it possible to adapt the bound objects.

CAD OPERATIONS THAT SHOULD PROPAGATE THROUGH THE BINDING LAYER

The following events should propagate via the binding layer to the FEM model.

Change

Geometry changes should be tracked. Should the geometry of the truss in the CAD model change e.g. the depth of the truss decreases, the binding layer should be able to propagate the change to the FEM model (geometrical binding)

Add

Additional truss members added to the truss also affect the FEM model. Monitoring geometry only would not be sufficient to convey an add event.

Remove

Deleting truss members in the CAD model should also be detected by the binding layer.

To satisfy the last two conditions, bindings should be established between sets of objects on both sides of the binding layer. By monitoring the sets, add and remove actions can be detected. On the CAD side, sets could be formed by using layers or by using specialized building components.

EXAMPLE 2: GRANULARITY OF DEPENDENCIES

Figure 4 shows the CAD representation of a slab on the left and the finite element representation of the same slab on the right. The following bindings would be possible.

- One FEM slab object bound to one CAD slab object.
- One FEM slab object bound to eight line objects.
- Eight FEM boundary line objects bound to eight CAD line objects.
- Many FEM node and element objects bound to one CAD slab object or eight lines.

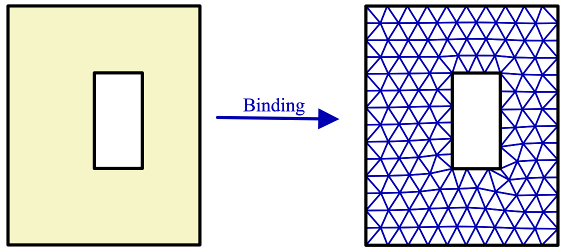

Figure 4: Granularity example

The choice of binding determines the granularity at which the CAD and FEM models are linked. With one binding (coarse granularity), it becomes more difficult to determine the exact effect of a change in the CAD model. This is equivalent to a document based system where a single change moves all the objects in the document to a new version.

Care should also be taken not to bind objects that are not related to each other. In this example, the FEM Node-objects are not directly dependent on the boundary of the CAD –object. They depend on the outline of the slab and on the analysis intention of the engineer that performs the analysis. The engineer might require mesh refinements in certain areas to perform a more accurate analysis; subsequently the mesh is derived from the boundary of the slab (FEM domain) which is linked to the boundary of the slab in the CAD model.

IMPLEMENTATION REQUIREMENTS

USER INTERFACE

The engineer should have tools that allow the creation of a FEM model based on the underlying CAD geometry. A simple tool would allow the engineer to view the geometry of the underlying CAD model as a background, as well as having access to the geometry when the FEM model is created. The engineer would then define the geometry of the structural model based on the underlying CAD geometry. The structural components e.g. nodes, element, supports and loads could be defined while the CAD geometry is displayed. The geometric dependencies should also be added during this process. Once the geometry of the FEM model is defined, it must be "binding" dependent on the underlying CAD model.

PERSISTENT NAMES

Objects in both the CAD and FEM model must be persistently identifiable. Without persistent names, it would not be possible to establish dependencies outside of a single runtime session.

STANDALONE MODELS

The CAD and FEM models should be independent from each other. The engineer should have the ability to work on structural analysis without being affected by changes in the related CAD model and visa versa.

COLLABORATION

The solution should support engineers to work together on a project although they are separated in time and location. Persistent storage of objects is achieved with the use of objectVCS (object version control system). (Firmenich, Koch, Beer and Richter 2005) A user would have a workspace on a local file system and functionality to upload and download objects from a central repository.

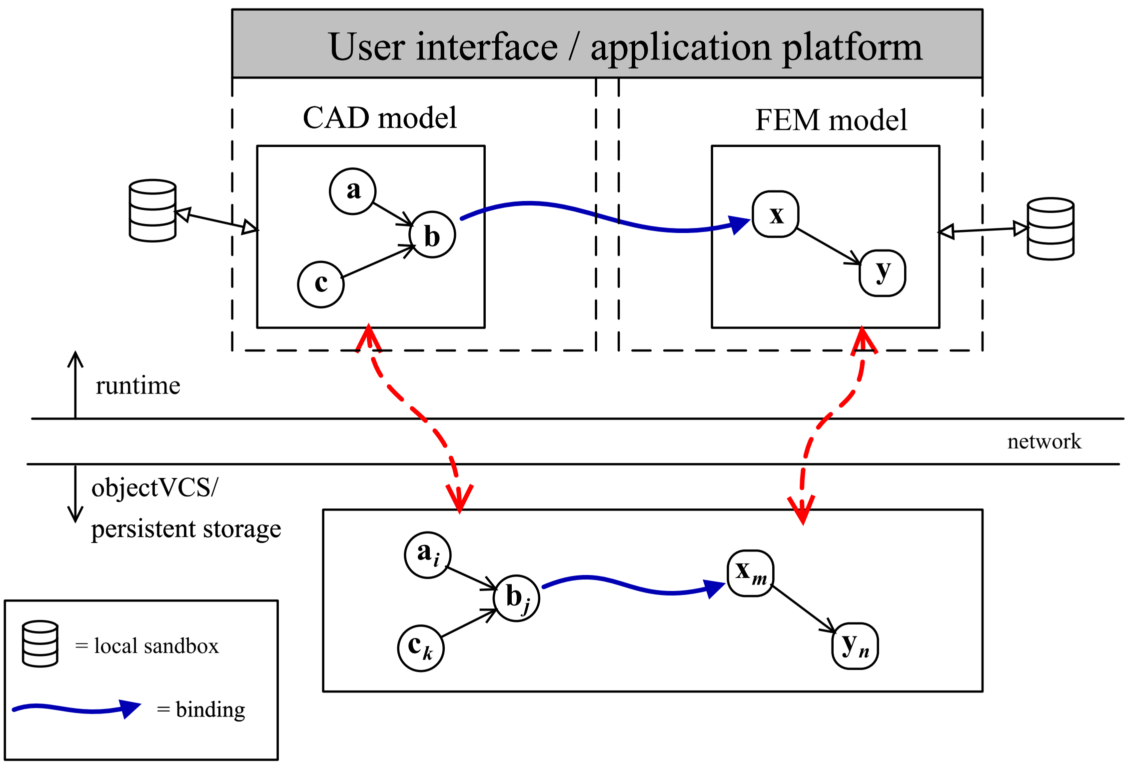

Figure 5: Software architecture

USER CONTROL

The software should provide powerful and flexible tools for the user in order to specify bindings and to update models. However, the process should be completely controllable by the user e.g. deferred updating (Van Rooyen 2002).

CURRENT STATUS OF RESEARCH

CADEMIA, an open source project that is currently under development by the Bauhaus University of Weimar, is used as the engineering platform for this project. An object-oriented framework for Finite Element Analysis was developed at the University of Stellenbosch (Olivier 2002) This framework serves as the FEM basis for the project. An objectVCS is also under development at the Bauhaus University of Weimar. This system provides functionality for storing versioned objects in a central repository via a network, and for selecting objects from the repository and loading it into a workspace sandbox (Beer 2006).

CONCLUSIONS

The separation of specialized models has the advantage of reducing the complexity of software applications. This, in turn, reduces the development and maintenance cost of the applications.

The main disadvantage of using multiple models rather than one comprehensive model is that consistency is not guaranteed because common information must be duplicated. The development of an efficient binding layer between the specialized models should contribute towards keeping information consistent.

ACKNOWLEDGMENTS

This project is sponsored by the German Academic Exchange Service (DAAD), the South African Institute of Steel Construction and Mittal Steel South Africa.

REFERENCES

Pahl, P.J. and K. Beucke (2000); *Neuere Konzepte des CAD im Bauwesen: Stand und Entwicklungen;* Digital Proceedings des Internationalen Kolloquiums über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen (IKM). Bauhaus-Universität Weimar, Weimar, DE.

Firmenich, B (2002). *CAD im Bauplanungsprozess: Verteilte Bearbeitung einer strukturierten Menge von Objektversionen,* Dissertation, Shaker, Aachen, DE.

Beer, D.G. and Firmenich, B.(2003): *Freigabestände von strukturierten Objektversionsmengen in Bauprojekten.* Digital Proceedings des Internationalen Kolloquiums über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen (IKM). Bauhaus-Universität Weimar, Weimar, DE.

Firmenich, B., Koch, C., Richter, T. and Beer, D.G. (2005): *Versioning structured object sets using text based Version Control Systems*. Proceedings of 22nd CIB-W78 Conference on Information Technology in Construction. Dresden: Institute of Construction Informatics, Dresden, DE.

Beer, D.G.(2006). *Systementwurf für verteilte Applikationen und Modelle im Bauplanungsprozess*. Dissertation, Bauhaus-Universität Weimar, DE.

Olivier, A.H. (2002). *Object-Oriented Framework for Finite Element Analysis*, MScEng thesis, Department of Civil Engineering, University of Stellenbosch, Stellenbosch, ZA.

Van Rooyen, G.C. (2002). *Structural analysis in a distributed collaboratory*, Dissertation, Department of Civil Engineering, University of Stellenbosch, Stellenbosch, ZA.