CAD/CAM INTEGRATED BUILDING PREFABRICATION BASED ON A PRODUCT DATA MODEL

Shutao Li[1], Jörg Isele[2], Karl-Heinz Häfele[3], Andreas Geiger[4]

ABSTRACT

This paper presents an approach pursued by the Forschungszentrum Karlsruhe (Karlsruhe Research Center), Germany, to prefabricating wall elements on the basis of the IFC (Industry Foundation Classes) product data model [1]. Prefabrication of wall elements for individually planned houses will only be successful, if the information flow between planning (architect) and manufacturing (machine) can be automated. Continuous information flow requires a neutral data model. The most promising product data model for buildings is the IFC model which is designed to cover the whole lifecycle and supported by various software vendors.

This paper describes information flow from the planning stage to the manufacturing process. Steps necessary to prepare the building model for prefabrication will be outlined in detail. Conversion of the geometric IFC building model into a machining feature-oriented BauXML model will be shown. In the end, an example of a simple house built with floor-to-ceiling AAC (Autoclaved Aerated Concrete) wall elements will be given.

KEYWORDS

Prefabrication, Product Data Model, IFC, BauXML, CAD/CAM

1. INTRODUCTION

Building prefabrication has a very long history. Three outstanding examples are:

- In 1851, the Great Exhibition was opened in London. It was the time when Europe was industrialized. People trusted in industrialized processes and in this way, one of the most exciting buildings was built within a fabulously short time. *Crystal Palace* was erected in London's Hyde Park within a few months only [2].
- A hundred years ago, the world famous architect Walter Gropius started his campaign for prefabrication in the building industry. This was the time when Henry Ford was optimizing the manufacturing process of Model T. When Konrad Wachsmann came to America at the beginning of World War II, Wachsmann and Gropius started to make their dream of prefabricated houses come true. Millions of dollars were spent,

1 Dipl.-Ing., Research Engineer, Karlsruhe Research Center, Institute for Applied Computer Science (IAI), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany, Phone +49/7247-825789, FAX +49/7247-825786, Shutao.Li@iai.fzk.de

2 Dr.-Ing., Senior Researcher, Phone +49/7247-825766, Joerg.Isele@iai.fzk.de

3 Dipl.-Ing., Research Engineer, Phone +49/7247-825745, Karl-Heinz.Haefele@iai.fzk.de

4 Dipl.-Ing. (FH), Research Engineer, Phone +49/7247-825789, Andreas.Geiger@iai.fzk.de

but the *General Panel Corporation* had built a few dozens of houses only when their prefabrication plants were closed [3].

- In parallel, the *Lustron Corporation* shared almost the same fate. A huge amount of government money was wasted, trying to copy the automotive success story to the domestic construction industry. About 2500 enameled metal sheet homes were built by 1950. Lots of them are still in use [4].

In the mid–nineties, construction industry in Germany reached its maximum due to the effects of German reunification. Since then, industry has suffered from a depression. These days, there are two popular proposals for managing the situation:

- One group of people believes in the traditional way. They focus on low-price bricks and low-wage workers from Eastern Europe.
- A second philosophy is to develop standard houses. Selling ready-made houses off the shelf is supposed to decrease costs.

Will these proposals work? It will not take long and the low-wage workers will bring their own bricks from their countries. Other industries will discover mass customization as a marketing and selling argument. Should construction industry go into another direction? It is interesting to see that the companies selling prefabricated wooden houses in Germany advertise their ability to deliver individual floor plans.

With modern IT (information technology), there is no need, neither for the retro approach nor for conformity. Lustron's problem was the missing flexibility – they had only very few types of houses. Gropius's and Wachsmann's concept was mass customization of prefabricated elements, but they had no IT tools to manage it and perhaps their architecture did not suit the common taste. The key statement of this article is: With a continuous IT-based workflow, individual houses or at least elements of houses can be prefabricated almost automatically at competitive costs.

2. WORKFLOW

The basic requirement when pursuing this prefabrication approach is a three-dimensional building information model created by an IFC-compliant CAAD (Computer Aided Architectural Design) system. Two-dimensional floor plans and non-object-oriented exchange formats like DXF are not sufficient.

A program called IfcWallModificator imports the IFC building model and provides comfortable access to the building elements, e.g. *IfcWallStandardcase* or *IfcOpeningElement*. Not only the geometry, but also properties like material and material layer can be displayed and edited. Relations between building elements are part of the building model and substantial to prefabrication. An example is the relation between a wall and its openings.

Wall connections are the relations between two or more walls. It is especially important to the structural analysis and prefabrication. The type of connection depends on the material and the assembly technique. Corner joints and miter joints are distinguished. Even though a miter joint usually is not suited for the practice, most CAAD systems use this type as default. If the walls are connected, then they are geometrically transferred to the IFC data with a miter joint. The first step when generating data for the prefabrication consists in verifying the wall connections from the constructional, technical, and assembly technology points of view.

The correction of the wall connections is to be accomplished just after importing the IFC model. The result can be exported into a new IFC file and illustrated and processed later in a CAAD system.

The next step of prefabrication is the segmentation of walls. According to the initial dimensions of the raw stones in the factory, the geometry of the wall and its openings (for windows, doors, etc.) and other potential constraints, a wall is segmented automatically into stone elements. All information on the stone elements, e.g. the dimension, the material, the placement, is exported into a model, called BauXML (explained in Sect. 3). The BauXML model is a machining feature-oriented semantic model. All cut-offs in the stone are represented and generated in BauXML as machining features. The BauXML model is imported into the production cell control program WallProducer. Within this program, all machining features are mapped to CNC macros and manufactured in the production. Figure 1 shows the complete workflow.

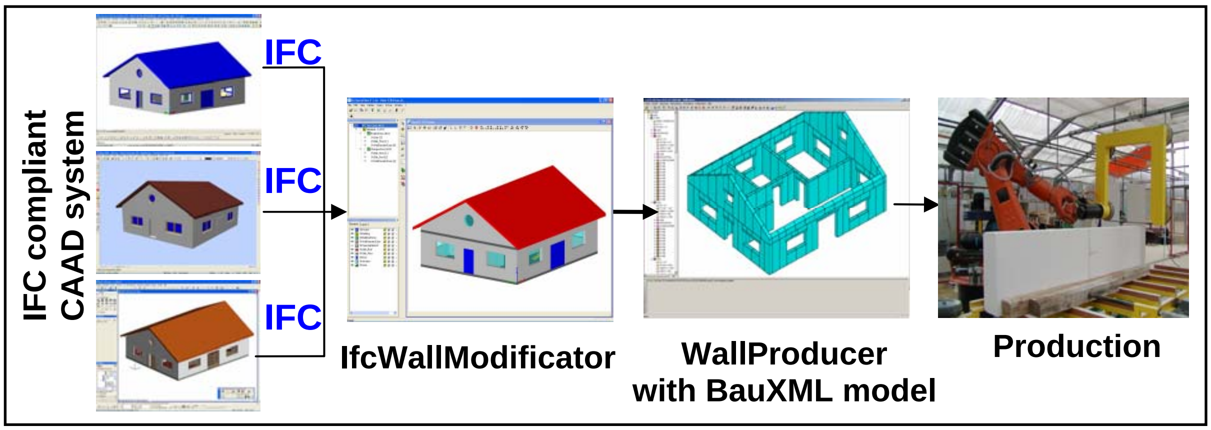


Figure 1: Workflow of building information exchange.

3. BAUXML SEMANTIC MODEL FOR MANUFACTURING

The IFC product data model supplies an enormous amount of information. For prefabrication, however, this IFC information is not always directly useful and necessary. In IFC, for instance, a wall has a geometry representation and all openings are positioned relative to the associated wall. However, no data type exists for stone elements. Instead, prefabrication focuses on stones. If a wall was divided into individual stones, the wall geometry must be converted and adapted to individual stones. An extension of IFC would be time-consuming. Therefore, a new data exchange format for prefabrication, BauXML, is used here. BauXML contains a subset of IFC information, enriched by some data especially for the manufacturing process. There is no reason why BauXML should not be integrated in IFC in the future.

BauXML is based on machining-feature and XML technology. Each stone element has an initial base form and several machining features. Of course, some stone elements may have no features at all. “A machining feature identifies a volume of material that shall be removed to obtain the final part geometry from the initial stock” [5]. Extrusion bodies, cones, planes, etc. are defined as features. Each feature has a GUID (Global Unique Identifier), an optional description, and an attribute “*IsFabricated*” which marks the manufacturing status of the feature. In default, this feature is "false" and it will be entered as “true” after manufacturing (Fig. 2).

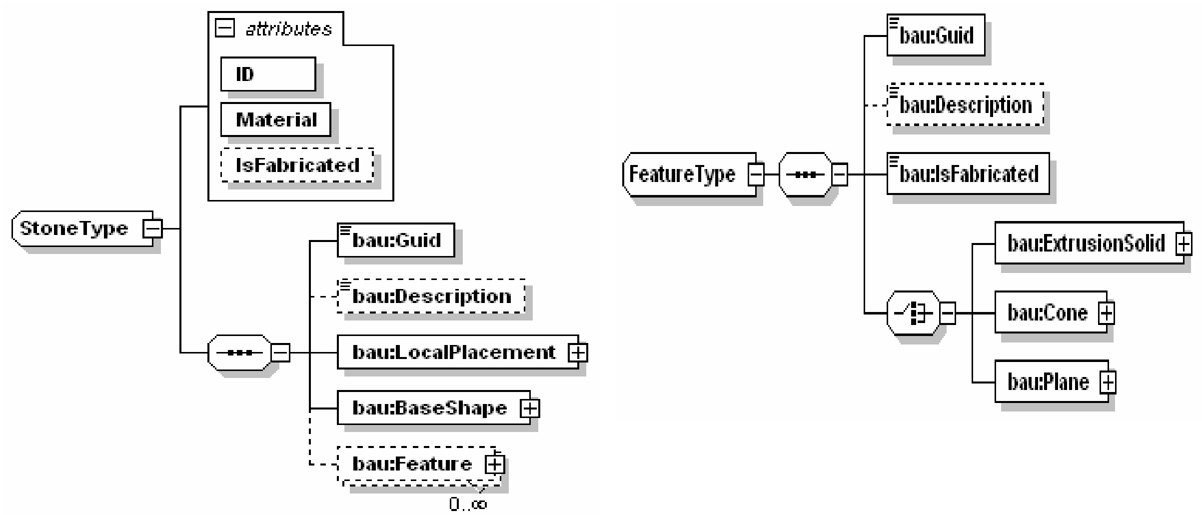


Figure 2: Definition of stone (left) and feature (right) in BauXML.

Most machining features correspond to the type of extrusion body. An extrusion body is swept from a 2D profile along a vector which is defined as “*ExtrusionPath*” having a certain orientation and magnitude. The 2D profile may be of different type, e.g. “*RectangleProfile”*, “*CircleProfile”*, “*GeneralProfile”*. Each 2D profile requires a placement and an attribute “*isArea*”. A placement specifies both direction and location. The presetting of the attribute “*isArea*” is “true”. It identifies a profile as surface or contour. As an example, the definition of the "*CircleProfile*" is given. The cut-outs for power sockets, peg holes or other circular openings can be defined with it. The circle is defined by a placement assigned to the center and a given radius. Data associated with a circle profile are shown in Figure 3:

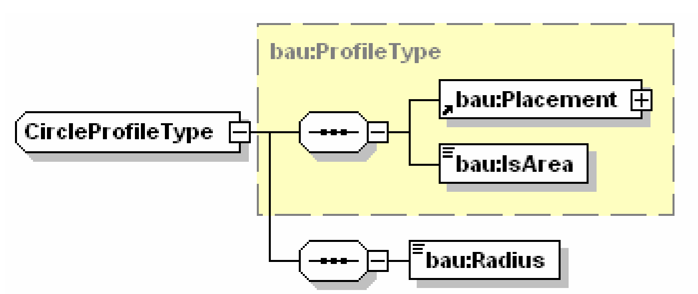


Figure 3: Definition of the “*CircleProfile”*.

4 SOFTWARE IMPLEMENTATION

4.1 SOFTWARE ARCHITECTURE AND USER INTERFACE

This project consists of two main programs: IfcWallModificator and WallProducer. Due to the limited space, the IfcWallModificator will be introduced only.

The IfcWallModificator is a standard MFC (Microsoft Foundation Classes) multi-document application. The user interface is based on the standard windows user interface. It has several features for the easy reading and display of the contents of an IFC file both graphically and alphanumerically as shown in Figure 4.

The geometry can be displayed as a 2D floor plan or a 3D building model. The IFC browser toolbar is used to navigate in the hierarchical tree structure of the IFC model. From the project to the building to the building stories to the different building elements, all the elements are shown and can be selected to obtain the properties or to view this element with all of its sub elements. The details of the currently selected object are displayed in the properties toolbar. This toolbar is subdivided into the element properties, the IFC properties, and the IFC relations. The IFC element toolbar is used to control visualization. This can be done by activating/deactivating the element types or layers.

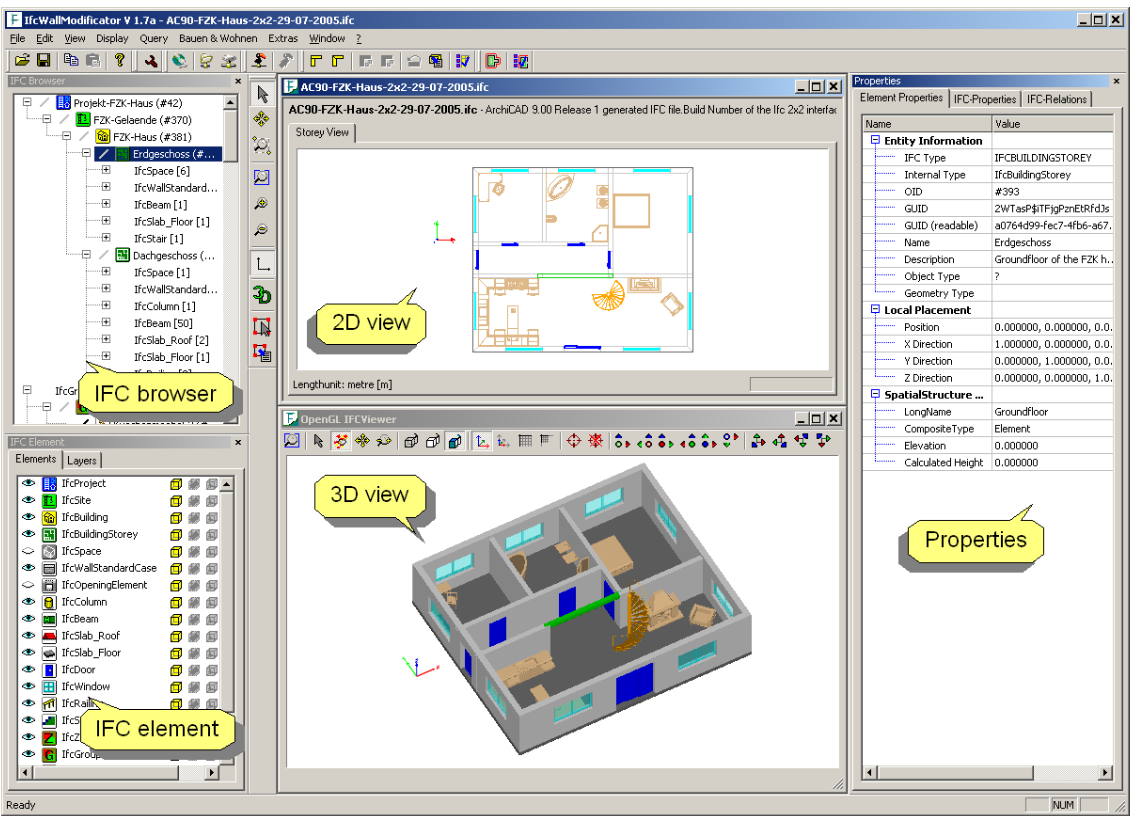


Figure 4: User interface of the IfcWallModificator program.

Implementation of the IfcWallModificator is sketched in Figure 5. Basically, there are three parts: IFC data import, data management and processing, BauXML export.

The framework IfcDb is implemented in C++ and statically linked to the IfcWallModificator. In the framework IfcDb, a hierarchical class structure is defined to map the IFC geometry and classes for all IFC entities and their relations. This approach ensures a flexible and efficient data access. The IfcDb not only is the internal "database", but also a geometry kernel to perform Boolean operations based on facetted boundary representations. IfcDb uses the ifcdb_ecco_dll generated by the ECCO toolkit for reading and writing IFC data. The

ECCO toolkit from PDTec [6] is a scanner and parser for STEP physical files based on ISO-10303, part 21, or according to the XML definition of part 28.

The functionalities of BauXML are implemented in the class *cBauXMLGenerator* of IfcWallModificator. To read and write the XML data, a C++ class structure is generated from the BauXML scheme with Altova XMLSpy [7].

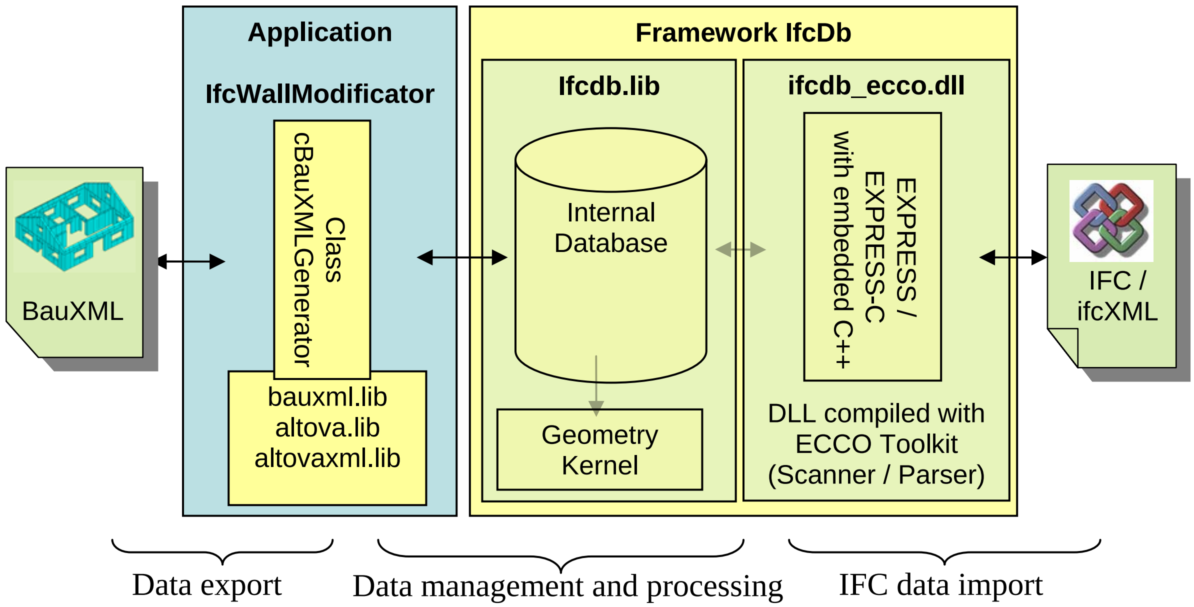


Figure 5: Functional architecture of the IfcWallModificator program.

Apart from these basic functionalities, IfcWallModificator is responsible for the following three main steps of prefabrication automation:

- Manipulation of wall connections
- Segmentation of walls into stone elements
- Generation of the machining feature-oriented BauXML model

4.2 Manipulation of wall connections

The IFC data among others comprise information about the wall material and the relations between walls. The algorithm is based on the IFC entity *IfcRelConnectsPathElements* (one by one relation between walls). If not available, this relation will be generated automatically. For the recalculation of the wall connections, up to five parameters can be taken into account:

- Wall length: The connection of long walls has a higher priority to be cut.
- Wall thickness: The higher priority to be cut is assigned to the thinner walls.
- Wall material/name: An optional list of wall material/name can specify the cut order.
- Critical angle: It decides whether a corner joint is more practicable than a miter joint. In this way, very acute angles (sharp corners) can be avoided in the resulting wall elements.

In case the automatic manipulation of wall connections might not give the desired result, the connection between two walls can be modified manually.

4.3 Segmentation of walls

Constraints for the segmentation are:

- Material: In reinforced concrete, component cuttings may not go through openings. The final "stone" with its openings is poured as a whole unit. From the structural analysis point of view, the permissible minimum distance of the cutting to the opening is interesting and defines a "*non-cut area*". When using floor-to-ceiling AAC stones, cuttings are set on both sides of the opening defined as "*PreferCuts*".
- Stone size: The minimum and maximum stone segment lengths in the factory.
- Openings: All openings of a wall are divided into two groups: Smaller and larger than a segment minimum. A smaller opening is assigned to a stone as a feature. In case the opening is cut, a feature will be assigned to each of the two stones. Only the large openings are considered geometric constraints for segmenting.

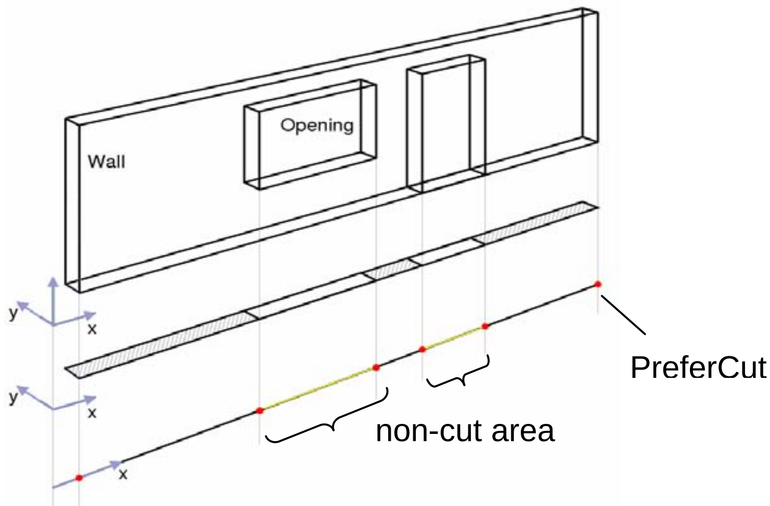
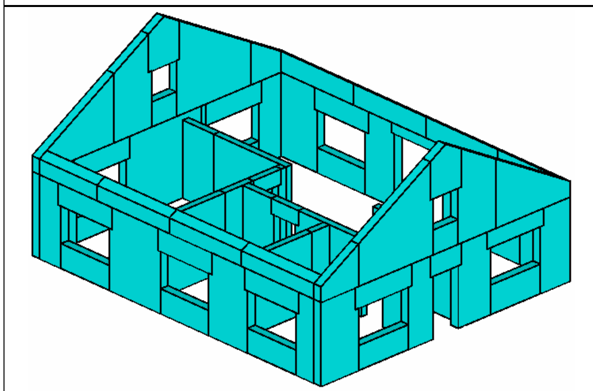


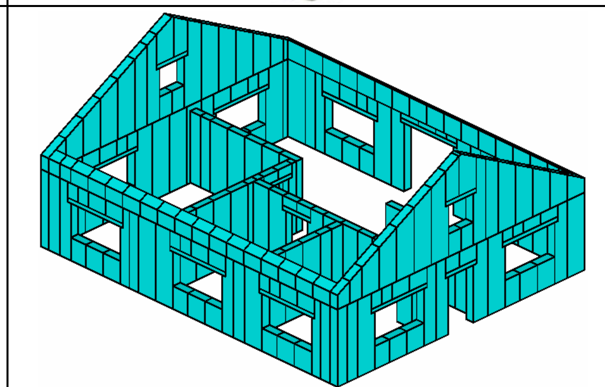
Figure 6: Treatment of a wall as a line.

Usually, the thickness of a wall remains constant over its length and height. This corresponds to the IFC entity *IfcWallStandardCase* represented with a wall axis. The "*non-cut areas*" and "*PreferCuts*" resulting from large openings will be marked on this wall axis. The segmenting algorithm can thus be reduced to handle a one-dimensional problem (Figure 6). The segmenting algorithm works recursively. As geometrical indication, the function gets the segmentation range and looks for "*non-cut areas*". Once a "*non-cut area*" is found, the two remaining ranges before and after will be segmented again. Basically, the segmentation cuts the segmentation range into stones of maximum stone length. Only if a resulting stone length is below the minimum stone length, is the stone merged with its predecessor, before this new element is cut into two equal stones.

4.4 Generation of the machining feature-oriented BauXML model

According to the segmentation, single stone elements will be created. Beginning with the start point of the wall, a rectangular stone body is produced from two cuttings. All stone bodies have the same thickness. The length of a lintel is described by the length of the opening and the bearings on two sides. According to the position of the stone in the wall, the height of a stone can be defined either as the wall or as parapet height, or as the height over an opening. Thus, a square base form of the stone element is created.





Query BauXML File

Element	Amount
Building	1
Storey	2
Wall	13
Stone	71
Feature	56

Stone Minimum: 200 mm
Stone Maximum: 3000 mm
Lintel height: overall
Bearing length: 150 mm

Query BauXML File

Element	Amount
Building	1
Storey	2
Wall	13
Stone	235
Feature	108

Stone Minimum: 200 mm
Stone Maximum: 700 mm
Lintel height: 150 mm
Bearing length: 120 mm

Table 1: BauXML models with different parameter settings.

The machining features are derived from the comparison between the base and the finished form of the stone element. Three categories of features are obtained:

- Features resulting from the wall footprint: The ground profile of the wall in IFC is compared with an ideal rectangle of wall thickness and wall length. After a footprint overlapping check, the resulting discrepancy profiles are assigned to one or more stones.
- Features resulting from the droop: Walls with bevels are often found within the gable walls. In IFC, they are described by the entity *IfcBooleanClippingResult*. An intersection contour of the plane with each concerned stone is computed and transformed into the local stone coordinate system. For these stone elements, a plane feature will be created.
- Features resulting from openings: Based on the geometry of openings, the stones around the openings will be generated with all features. For example, the cut-out

for the bearing of the stone element will be defined as an "*ExtrusionSolid*" feature with a "*RectangleProfile*". It will be assigned to the two stones on both sides of an opening.

Of course, all necessary information, such as the information about the project, the building, the stories, and walls, is stored in the BauXML model.

A 3D visualization of the BauXML model serves to examine whether the original model can be regenerated from the BauXML data. The following building model made by the Institute for Applied Computer Science (IAI) consists of 2 stories with 13 walls. By different configurations of the segmenting constraints and the parameters of the lintel, different BauXML models are generated (Table 1).

From the example presented above, it is evident that all walls of the original building model can be regenerated completely by the BauXML data. The BauXML model will be applied by a wall prefabrication factory.

5. PROTOTYPE OF A PREFABRICATION FACTORY

A prototype factory was put up at Forschungszentrum Karlsruhe, Germany. As wall material, the floor-to-ceiling AAC blocks were chosen. They meet all demands outlined in energy regulations and they can be cut and milled easily. These stones are up to 750 mm wide and may have variable thicknesses. An industrial robot is used to handle a diamond wire saw or a high-speed milling spindle (Figure 7).

All programs for tool exchange, adjustment of cutting depth, feed, and revolution speed were implemented in the robot control and can be adjusted using the production cell control program WallProducer. BauXML features are mapped to macros in the robot controller. The robot is fed by the feature of name and the feature of parameters, which may be a "hole" with "450 mm" radius. WallProducer reads the BauXML model, communicates with the robot and its peripheries via OPC (OLE in Process Control), and coordinates the entire automatic manufacturing schedule for every feature.

Figure 7: Prototype of a robot cell (left with a wire saw, right with a milling spindle).

6. CONCLUSIONS

Compared to on-site construction, prefabrication of AAC elements ensures high design flexibility, enables production automation, significantly improves quality, and reduces costs. In order to benefit from these advantages, a continuous IT workflow is essential. Therefore, the key technologies are the neutral product data model IFC and data standards like XML.

The approach presented for the prefabrication of wall elements is based on the IFC product data model. Individual architectural designs from different IFC-compliant CAAD systems are the input of the workflow. Wall connections which are not suitable for prefabrication are corrected automatically in IfcWallModificator. The wall segmentation is implemented for floor-to-ceiling AAC and basically for precast concrete. Other rules and constraints for further prefabrication concepts can be added easily.

For manufacture, a machining feature-oriented model, called BauXML, is defined and implemented. BauXML focuses on the stone elements. The machining features are created automatically. Other special machining features can be added manually in WallProducer. Once all stone elements are described completely, this program coordinates the robot and other peripheries and guarantees an automatic production procedure.

The outputs of the prototype factory are mass-customized building elements. Due to the high production precision, assembly on the construction site is time- and cost-efficient.

7. ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the support by Data Design System, Graphisoft, G. E.M. Team Solutions, Nemetschek AG, and Solibri.

8. REFERENCES

1. Shutao Li: Development of a procedure for the automation of the CAD/CAM chain in the individual manufacturing by the example of stonework walls. Ph.D. Dissertation, Universität Karlsruhe (TH), Germany, 2006.
2. Konrad Wachsmann: Wendepunkt im Bauen. Dt. Verl.-Anst., Stuttgart, 1989.
3. Gilbert Herbert: The Dream of the Factory-Made House. The MIT Press, Cambridge, Massachusetts, London, England, 1984.
4. Thomas T. Fetters: The Lustron Home – The History of a Postwar Prefabricated Housing Experiment. McFarland Company, Inc., Publishers, Jefferson, North Carolina, 2002.
5. ISO 10303-224: Mechanical product definition for process planning using machining features. ISO 10303-224, P. 96, second edition, 2001.
6. http://www.pdtec.de/
7. http://www.altova.com/