A FRAMEWORK FOR LINKING DESIGN AND FABRICATION IN GEOMETRICALLY COMPLEX ARCHITECTURE

ABSTRACT

While the development of specialized geometric design tools for geometrically complex architecture becomes increasingly common, an easily accessible link to fabrication processes is still missing. Common approaches to link design and fabrication suffer from various inefficiencies. We present a fabrication planning framework capable of extending specialized geometric design tools to seamless digital chains that directly connect designer and CNC fabrication machinery. By integrating design and fabrication planning processes, our framework avoids common data exchange problems and allows for feedback on fabrication planning results at design time. In combination with a geometrically versatile representation of physical parts, we achieve a reusable solution that is applicable to the diverse requirements of geometrically complex architecture. In this paper, we present the concept, design and implementation of our solution together with experiments and results.
INTRODUCTION

Compared with conventional architecture, design and fabrication processes are particularly challenging in geometrically complex architecture. Here, special requirements are emerging regarding aspects of design as well as fabrication. In particular, these requirements concern the development and use of specialized geometric design tools rather than insufficient general purpose CAAD systems for design (Füssler 2008), and the employment of CNC machines for fabrication (Kolarevic 2001).

A solution that aims to support design and fabrication of geometrically complex architecture should link geometric design tools and CNC machines. We identify three major requirements:

1. The solution must be adaptable to geometric requirements of varying design tasks,
2. data exchange between design and fabrication must be seamless, and
3. information regarding the impact of a design on fabrication processes must be available to the user at design time.

Current approaches usually link design and fabrication either sequentially, one after the other, or in an integrated but project-specific way (Figure 1). While both have their benefits, neither approach fulfills all our requirements. In this paper, we describe a framework for linking design and fabrication to support the development of geometrically complex architecture that aims to meet these requirements in an optimal way.

The paper is organized as follows. We review current approaches for linking specialized geometric design tools and CNC machines for the fabrication of geometrically complex architecture (Section Current Approaches), present a framework for linking design and fabrication processes (Section A Framework for Linking Design and Fabrication), describe the design and implementation of the framework (Section Framework Details), and discuss experiments and results (Section Experiments).

CURRENT APPROACHES

Current practice follows two approaches to realize geometrically complex architecture by linking design and fabrication. Both exhibit significant limitations, which impact design quality and cost.

SEQUENTIAL WORKFLOW

This approach completely separates design and fabrication processes. Typically, specialized geometric design tools are used for the generation of producible parts representing a design, for example a building hull. Resulting data is passed to a fabricator responsible for fabrication planning and subsequent fabrication (Figure 1a).

Design tools typically target specific aspects of fabrication. For example, whereas some generate geometries that match specific material properties, such as single curved panels (Pottmann et al. 2008), or smooth unions of ruled surface strips (Flöry et al. 2013), others approximate design surfaces by groups of identical parts, which is suitable for subsequent fabrication (Zimmer et al. 2013; Singh and Schaefer 2010; Sechelmann, Rörig and Bobenko 2013). However, resulting part geometries still need to be separately processed for fabrication. Due to the isolated nature of process steps, obtaining feedback from fabrication at design time is not feasible in this workflow. Thus, iterations between design and fabrication processes are highly inefficient. Moreover, the typical project-based pairing of design and fabrication facilities is prone to data exchange problems, which may result in redundant labour and additional, non-value-added effort for designers and fabricators (Anderberg 2012).

CUSTOM DIGITAL CHAIN

Specialized design software is developed in order to interface with project-specific fabrication facilities and processes. In contrast to the sequential workflow, here, design and fabrication planning can be integrated, which allows for fabrication feedback while designing and minimizes interfacing problems between design and fabrication planning.
fabrication (Figure 1b). For example, some practices offer the development of project-based digital chain solutions that automate the detailing of building parts and generate machine code instead of engineering drawings (Scheurer and Schindler 2006; Scheurer 2008). Sass (2007) presents a rule-based method to produce designs by lateral contouring, which drives various types of digital fabrication devices across scales from one geometric file. Schmieder and Mehrten (2013) describe a complete digital chain from design to fabrication that covers the cladding of doubly curved building surfaces with metal panels.

However, the main drawback of these solutions is that they are typically confined to specific projects or special geometric problems and as such cannot be easily reused or adapted to different geometric requirements.

A FRAMEWORK FOR LINKING DESIGN AND FABRICATION: OVERVIEW

We present a fabrication planning framework designed to be linked with a custom geometry design tool provided by a user. Thereby, while the user is free in developing the design logic, geometry design and fabrication planning become part of one software (Figure 2). As a consequence of this integration, design data is transferred to fabrication planning internally, which avoids the problem of error-prone interfacing with external software, and allows for simultaneous feedback on fabrication planning. To achieve a versatile workflow that satisfies needs of architectural projects with varying requirements regarding geometries and machines, our framework employs a specific representation of physical parts (Figure 3). From this representation, it is easy to derive fabrication instructions for CNC machines (NC code).

With the framework, specialized custom design software for all scales of geometrically complex architecture can be extended into seamless software solutions integrating design and fabrication. While the resulting software is still confined to a specific problem, it is much easier to get there for an average user. Traditionally, custom design tools address machines directly, while our framework unifies machine-specific CAM processes. The user can conveniently build a design tool by implementing the framework interface without knowing about specific machines (Figure 4). Details regarding machine or stock material can be adjusted by fabrication parameters at design time.

2 Integrated design tool achieved by combining a user-provided geometry design software with the proposed fabrication planning framework

3 For the proposed framework, we employ a versatile representation of physical parts applicable to different scenarios in geometrically complex architecture. (a) Two-dimensional structure for facade construction or formwork. (b) Bar-shaped structure for support structures such as roof frameworks. (c) Solid structure for mold making or formwork.
FRAMEWORK ARCHITECTURE

Adjusting parameters on the geometry design side results in the generation of geometrically complex structures composed of producible parts. Internally, the user-defined geometry design software sends lists of custom parts resulting from current design activity to the framework. These must comply with our representation of physical parts (see Figure 3).

Lists of design parts and fabrication constraints are the two major inputs for the framework. The latter concerns properties of the targeted fabrication setup. From these two inputs, a CAM solution is generated, that represents a preliminary suggestion for an efficient fabrication. Together with fabrication constraints, the CAM solution serves as a basis for the computation of fabrication feedback, that is, information based on fabrication suggestions concerning factors such as cost, time or logistics. This feedback is processed by the user’s geometry design software. With the presentation and integration of fabrication feedback, we establish an iterating loop between design and fabrication planning.

FRAMEWORK USAGE

The usage of the framework can be divided into tool development on the one hand, and designing with a resulting integrated design tool on the other.

For development, we provide a data structure for transmitting design parts to our framework. Doing so will trigger the computation of updated fabrication feedback. The user may incorporate feedback information in the geometry design tool. For instance, feedback can be used to identify parts that are too heavy, too expensive or too large for production, or to restrict design parameter inputs to values that comply with actual fabrication constraints.

The design workflow will typically involve the continuous editing of design geometry and the tuning of fabrication constraints followed by adaptions of these parameters based on a review of feedback data. As an example, (Figure 5) outlines a customized integrated design tool based on our framework, dedicated to designing structures consisting of intersecting wooden panels in non-orthogonal positions. The user-implemented geometry design logic is responsible for
Outline of an integrated design tool based on our framework. (a) Geometry design interface defined by the user. (b) Structure resulting from design parameter changes. (c) Fabrication constraints interface provided by our framework. (d) Example for feedback information as implemented by the user: tight packing of parts and related tool path.

Two ways to define location and direction of a linear cutting tool based on the base plane of a stock panel. (a) Two pairs of translational axes on either side of the stock panel. (b) Two translational and two rotational axes.

re-calculating part geometries based on parameter-controlled movements of panels, and for passing these to the framework. Subsequently, feedback such as the number of required stock panels, estimated fabrication time, or material cost is returned to and displayed by the geometry design logic, on which the user can react by re-adjusting design or fabrication parameters. Eventually, machine-specific NC code is produced to be run on a CNC machine in order to fabricate the design.

PHYSICAL PART GEOMETRY
To minimize machining complexity while still being able to produce geometrically complex shapes, we focus on flat or moderately curved stock material for the fabrication of physical parts. Thereby, tool movement can be constrained to a plane, which limits the degrees of freedom required for machining. However, by joining or stacking flat parts, it is straightforward to achieve three-dimensional structures as required for the complex architectural geometries we aim at (see Figure 3).

With the exception of tagging and labeling, machining operations that only partially penetrate the stock material are rarely applied. Thus we assume the most important machining operation to be the one-pass, full depth cutting of closed shape outlines. In order to join panels at arbitrary angles (miter joints), we need to describe successive positions of a linear cutting medium at various angles with respect to a stock panel.

REQUIRED DEGREE OF FREEDOM
CNC machines usually control their tools by translational and rotational axes that demand for coordinate and angle values, respectively. For describing straight lines that represent successive tool positions, we consider two options that each employ four axes (Figure 6).

1) Definition of a straight line by two points restricted to two parallel planes. The degree of freedom is represented by two translational axes in either plane.

2) Definition of a straight line by a base point restricted to a plane and a direction vector. The degree of freedom is represented by two translational axes for the base point and two rotational axes for the line direction.

FRAMEWORK DETAILS
In this section, we illustrate the previously presented concept in the context of a concrete setting. Please note that our concept could be easily extended to other scenarios.
FABRICATION METHODS

The cutting of panel outlines is often performed with the side of a milling tool, so-called flank milling (Schindler 2007). Though widespread, this technique has issues that our approach wants to avoid (Figure 7). For milling tools, there exists a direct relation between tool length and tool diameter. While longer end mills may be capable of fully penetrating a given stock panel, these will lack precision at producing non-convex corners (Figure 7a). Also, in order to avoid machine damage when penetrating stock material, horizontally mounted stock panels demand for an additional substructure, which usually results in redundant mounting operations and material waste (Figure 7a). Finally, driving an end mill through stock material along a ruled surface yields the ruled surface’s offsets on both sides of the tool (Figure 7b). For the general case of a non-torsal ruled surface, these do not contain straight rulings and therefore are not ruled surfaces (Pottmann et al. 2007).

Thus, with flank milling being a trade-off between cutting depth and precision, we are instead putting a focus on water jet and hot wire cutting (Figure 7c). These machines employ cutting media of sufficiently small diameter and adequate length for architectural needs, and do not require substructures.

PART REPRESENTATION FOR FABRICATION PLANNING

CUT PATCH

For the framework-internal representation of parts, we employ ruled surfaces, surfaces that consist of straight lines, so-called rulings (do Carmo 1976). We are subdividing a cutting path into ruled
TIMING OF TOOL MOVEMENT

With endpoints of a straight line moving along two separate curves, speed variations for either end will generally result in different geometries. To account for that, we are restricting movement to constant speed for each segment, such that both line endpoints start and arrive at the same time. This may require different speeds for each segment. In geometrical terms, both curves need to be parametrized on the same interval by scaled arc-length parametrization (do Carmo 1976).

CUT PATCH SEQUENCE

Describing the cutting process for an entire part requires the addition of cut patches into a sequence (Figure 8e). Segment curves of one cut patch must be oriented in a way that their endpoints match the respective start points of the next cut patch’s segments.

Please note that the base planes of segments do not need to coincide with the faces of panels (Section CUT PATCH ) and that the segments might differ from an actual tool path required by a specific machine. (Figure 9) depicts a scenario related to hot wire cutting, where a linear cutting tool is driven by two vertical portals. Thereby, endpoints of a line representing the cutting tool must be elements of two parallel planes that are offsets of the vertically mounted stock panel’s base plane. However, it is straightforward to derive specific NC code for such scenarios from our part representation.

DATA STRUCTURE

We see that cut patches are uniquely defined by a pair of curves. Design parts generated by the user’s software can be passed to the framework via a data structure that stores sequences of cut patches.

A data structure object is initialized by a reference plane along with the an offset distance, representing the panel’s thickness. When cut patches are added to the object, their segment pairs are validated for the first segment curve being element of the stated reference plane and the second one being element of the offset plane at a distance given by the panel thickness. Also, segment curve start points must match endpoints of the previous cut patch’s segments within a reasonable tolerance. Internally, all curves are represented as NURBS curves (Piegl and Tiller 1997).

FABRICATION CONSTRAINTS

All framework computations depend on constraints regarding the actual fabrication setup. These include geometrical, physical, and financial properties of a targeted CNC machine and stock surface patches representing discrete cutting steps. As an advantage of cutting processes involving four axes, such movement of a linear cutting tool can be described by a pair of planar and possibly curved segments with respect to two parallel planes situated on either side of a stock panel. Note that these planes do not need to coincide with the panels’ faces, which we do not require to be planar. A segment is one of:

1. Straight line
2. Curve
3. Point (Degenerate Curve)

CUT PATCH CLASSES

Combinations of segments yield different kinds of cut patches, and, by that, different classes of ruled surfaces:

Orthogonal General Cylinder Surface: Two congruent curves that coincide when viewed normal to the stock panel base plane. This surface represents an orthogonal cut (Figure 8a).

Non-Orthogonal General Cylinder Surface: Two congruent curves that do not coincide when viewed normal to the stock panel base plane. This surface represents a non-orthogonal cut at a constant tool angle (Figure 8b).

General Ruled Surface: Defined by two non-congruent curves, this surface represents a non-orthogonal cut at varying tool angles (Figure 8c).

General Conical Surface: As a special case of two non-congruent segments, one segment is allowed to degenerate to a point (Figure 8d). This allows for kinks involving only one side of a panel.
material and can be defined by the user via an interface provided by the framework. To avoid ambiguities such as selecting stock material that exceeds the targeted machine’s production space, user-supplied constraints are validated before they are made available to the framework.

**CAM SOLUTION**

A list of design part definitions received by the framework is tightly packed into the smallest possible number of stock material units as defined in fabrication constraints, and for each of these units, a tool path is generated (see Figure 5d). We are referring to the resulting list of stock panels containing packed parts and tool paths as the CAM solution.

Employed algorithms depend on targeted production machinery and employed material as defined in fabrication constraints. For our implementation, we are targeting hot wire cutting, which, because of stock material being mounted vertically, requires a cutting tool to work its way through a stock panel from top to bottom. Respectively, a recursive bin packing algorithm packs bounding rectangles of parts (Lodi, Martello, and Monaci 2002), which are stored in binary trees (Knuth 2005). In order to generate tool paths, these binary trees are searched recursively.

**FABRICATION FEEDBACK**

Based on CAM solution and fabrication constraints for the current design state, the framework computes fabrication feedback (see Figure 2). This feedback mainly involves performance indicators and NC code, but also includes information about the current CAM solution, such as packed part geometries and tool paths.

**PERFORMANCE INDICATORS**

CAM solution results such as tool path lengths, part volumes and the amount of required stock panels are combined with information regarding the actual fabrication setup provided by fabrication constraints. (Figure 10) gives three examples for the calculation of performance indicators by information gained from CAM solution and fabrication constraints.

**NC CODE**

Instead of outputting proprietary CAM data that needs to be translated to actual machine code by machine-specific post-processing software, our framework targets machines directly via NC code, particularly a subset of G-CODE (ISO 6983). G-CODE provides motion commands to be successively executed by the tool and modal commands such as tool activation/deactivation, or feed velocity rates for each step. This standard is considered to be understood by most machines and can be easily derived from our part representation. The actual code depends on factors such as machine type, material parameters, and on precision requirements set by the user.

**EXPERIMENTS**

The display case structure shown in (Figure 11c) was part of an exhibition design achieved by an early software version. Here, an MDF sub-construction comprising miter joints was fabricated with a CNC buzz saw. The roof framework shown in (Figure 12) also resulted from a predecessor, that generated outlines of steel and timber parts from a given NURBS surface. Even though both solutions were still closely confined to their geometric project requirements, they achieved a considerable reduction of effort and cost compared with common fabrication techniques. Both projects’ geometric requirements are compatible with our framework and strongly motivated the more versatile solution presented in this paper.

Hence, we are currently adapting previous solutions to work with our framework. (Figure 5) outlines an integrated geometry design tool based on the geometric requirements shown in (Figure 11). This design tool was used for the realization of a prototypical structure (Figure 11a). With an end mill cutter driven by a 5-axis steel mill, we fabricated fiber cement panels exhibiting non-orthogonal bounding surfaces with respect to the panel base planes. The panels were held in place by a mounting construction specifically developed for this application. This construction fastened the stock material by bolts on the back side along with vacuum applied by an external suction pump (Figure 11b).
CONCLUSION

We presented a solution for an efficient and versatile way of establishing geometrically complex architecture. By combining our fabrication planning framework with a custom geometry design tool developed by a user, we achieve fabrication feedback while designing and eliminating the need for error-prone data exchange between design and fabrication planning.

Even though it is perfectly sufficient for getting an impression about what kind of architecture we are addressing, technically, “geometrically complex architecture” is a rather vague term. Hence, we do not aim for an all-in-one system suitable for every purpose. Instead, we restrict inputs to a special kind of geometry, which, though, is versatile enough to be applied to complex architectural applications of all sorts, from formwork over facade constructions to support structures and mold making.

Our method constitutes a compromise between designing with pre-defined standard parts such as bricks on the one hand, and freeform modeling without any fabrication awareness on the other. We demand the user to think about design parts from the ground up, but allow a lot of flexibility concerning these parts. Eventually, our framework makes it easy to establish a link to fabrication facilities.

FUTURE WORK

We are going to explore additional architectural applications and we will extend the possible feedback. Currently, we are preparing a hot wire cutter to serve as a hardware platform for the fabrication of physical prototypes. Most importantly, we are going to enhance our part representation: We want to allow panels to also include gaps. Of course, this is limited to machines such as water jet cutters, that can turn their cutting medium on and off while operating. Further, we are looking into ways of enhancing our part representation to define more than one cut per side. Finally, we are going to experiment with unequal parametrization of curves defining cut patches, which will allow varying feed velocity within cut patches and facilitate the fabrication of complex shapes such as oloid.
ACKNOWLEDGEMENTS
This work was supported by the “Innovative Idee” program and the office of the Vice Rector for Research, Vienna University of Technology. We want to thank Filip Petrusheski, Milos Sipetic and Gabriel Wurzer for their valuable feedback.

REFERENCES


IMAGE CREDITS
Figure 1-10. Image credit to Authors (2014).

Figure 11c. Pichler, Klaus (2011) Display case, Niederösterreichische Landesausstellung 2011.

Figure 12: S. Nawara, Goga (2010) Roof framework, Leopoldsdorf.

HEINZ SCHMIEDHOFER graduated from Vienna University of Technology, Faculty of Architecture. He has been contributing to several research projects in the field of Architectural Geometry. Currently, besides working on his PhD thesis, he is associate at feasible geometry-consulting OG, a company dedicated to assisting architects and designers with the realization of geometrically complex architecture.

MARTIN REIS graduated from Vienna University of Technology, Faculty of Architecture, in 2006. From 2004 to 2011 he was part of the Viennese architectural office Baar-Baarenfels Architekten, where he actively participated in several award-winning architectural projects. Since 2006 he has been employed as a research assistant at Vienna UT, with an emphasis on process optimization in digital fabrication. Currently, Martin Reis is associate at Vienna-based start-up feasible geometry-consulting OG.

SIMON FLÖRY received his PhD in mathematics from Vienna University of Technology in 2010. He is leading Vienna-based start-up Rechenraum, developing bespoke software and consulting solutions in geometry processing and architectural geometry. Having a great interest in various aspects of software development, he has been maintaining and contributing to open-source software projects for more than a decade. His ongoing research focuses on effectively processing and optimizing geometric data for facade engineering, geodesy and medical applications.

FLORIAN RIST studied mathematics and architecture at the TU Munich, worked as an architect in Germany, and is currently employed as a Senior Scientist at Vienna University of Technology at the department of Arts and Design. His main research interests are reverse engineering processes in artistic and architectural design and their seamless integration into rapid prototyping and rapid manufacturing workflows as well as real-time on-line control of digital fabrication machines to establish new design techniques. At present, he is enrolled at Vienna UT as a doctoral student and working on his thesis on point cloud based manufacturing processes.

GEORG SUTER is Associate Professor at the Faculty of Architecture and Planning, Vienna University of Technology, Vienna, Austria and Head of the Design Computing Group. His research interests are in the areas of Building Information Modeling, Building Automation, and Digital Fabrication. He is a committee member of the European Group for Intelligent Computing in Engineering. He received a Dipl. Arch. degree from the Swiss Institute of Technology (ETH) in Zurich, Switzerland, and MSc and PhD degrees in Building Performance and Diagnostics from Carnegie Mellon University in Pittsburgh, USA.