Modelling Alberti's Column System

Generative modelling and digital fabrication of classical architectural elements

Eduardo Castro e Costa[1], Filipe Coutinho[2], José Pinto Duarte[3], Mário Krüger[4]
[1,2,4]University of Coimbra, Portugal, [1,3]TU Lisbon, Portugal
[1]http://www.fa.utl.pt/~castroecosta, [2]http://www.darq.uc.pt./~filipecoutinho, [3]http://www.fa.utl.pt/~jduarte, [4]http://www.ces.uc.pt/whee/investigadores/index.php?target=cv&id_investigador=115
[1]castroecosta@fa.utl.pt, [2]filipecoutinho@darq.uc.pt,, [3]jduarte@fa.utl.pt, [4]kruger@ci.uc.pt

Abstract. *The research presented further is part of the Digital Alberti research project, which aims to assess the influence of Leon Batista Alberti's theory on Portuguese architecture, through the use of digital technologies. One of the project tasks implied computational and physical modelling of Alberti's column system. Development of the computational model implied decoding Alberti's treatise on architecture De re aedificatoria into a consistent set of parameters and relationships, and then implementing these into generative parametric computer programs through visual programming language Grasshopper. This computational model is able to automatically generate physical models of classical columns according to Alberti's canons. These digital models were then materialized through production of physical models, through rapid prototyping and digital fabrication technologies. Special attention is given to the CNC stone milling of a Corinthian capital.*

Keywords. *Alberti; De re aedificatoria; Column system; Generative modelling; Digital fabrication.*

INTRODUCTION

This paper presents part of the latest developments in the Digital Alberti research project. In this project, Alberti's treatise on architecture, De re aedificatoria – actually its Portuguese translation, "Da arte edificatória" (Alberti, 2011) – is decoded through the use of new technologies. The project's main objective is to determine the influence of Alberti's theory on Portuguese architecture in the Counter-reform period.

This can be achieved by converting parts of the treatise into shape grammars, and then determining the changes required to those shape grammars to account for the generation of Portuguese classical buildings (Coutinho, et al., 2011). Therefore, two shape grammars were developed, one focusing on the design of the column system, the other focusing on the design of sacred buildings, i.e., churches. Among so many issues addressed in the treatise, these two themes were chosen because Alberti's description of the rules for both themes is particularly thorough.

As a parallel task to the translation into shape grammars, other types of models were developed – digital, physical, computational and virtual –, both for the column system and for the churches. The elaboration of such models contributed to a deeper understanding of the treatise, and thus aiding to the construction of the shape grammars. This paper focuses on the development of both the computational and the physical models of the column system.

Methodology

Three main tasks, intertwined rather than sequential, are featured in the modelling process: a) the development of the computational model, b) its implementation, and c) the production of the physical models.

De re aedificatoria can be interpreted as a set of instructions regarding the art of building (Coutinho, et al., 2011). Therefore, Alberti's treatise could be translated into a parametric generative system that carries out his instructions. Rather than an implementation of the shape grammar developed for the column system, the parametric computational model was built directly from the treatise, aiming at complementing the grammar development. The two cooperative approaches enriched the research on Alberti's column system, through sharing of ideas, namely during the reading and interpretation of the treatise.

The parametric computational model was implemented as a generative computer program that automatically generates three-dimensional (3D) digital models of columns, according to Alberti, as well as plausible variations. From the 3D digital model, we produced the corresponding physical models through rapid prototyping and digital fabrication technologies. Both from the 3D digital model and the computational model itself, a virtual model could be developed, and the two-dimensional (2D) digital models, or drawings, can be extracted (Duarte, et al., 2011).

Among the several physical models produced, particular focus goes to the production of a natural scale stone model of a Corinthian capital, which resulted from the cooperation between the fields of academic research and industry.

DEVELOPMENT OF THE COMPUTATIONAL MODEL

Rules as single entities

The first step in understanding Alberti's treatise was its analytical reading, focusing on the chapters describing the column system. In these chapters, several rules can be found regarding the various elements that form the system. All the elements were thoroughly analysed in order to better understand these rules, which were written down in tables. Most of these rules describe numerical relationships between elements of the column system, so they were translated into mathematics in a consistent fashion, whenever possible.

Table 1 contains part of the rules for the Corinthian capital, adapted to the English translation of the treatise (Alberti, 1988). Each row corresponds to a rule, containing a transcription of the original text and the corresponding mathematical interpretation, as well as the rule's location in the treatise and its internal numbering (Castro e Costa, 2012, p. 47).

Relationships and hierarchical trees

Analyzing the tables, three main types of rules, or relationships, can be found. In **subdivision** rules, like for example rule #02, Alberti divides previously defined elements of the column system into smaller ones. By doing so, the author also determines the order of these smaller elements, assigning them positional relationships of "above" and "below". In **proportion** rules, like for example rule #01, the author assigns dimensions to the elements. These dimensions are always determined in function of previously defined elements. In **detailing** rules, Alberti determines the shape of the elements; for example in rules #14 and #15, information is given about the sprouting stalks.

Analysis of the relationships within the system, namely of those defined in subdivision rules, sug-

Table 1
Interpretation table of Alberti's rules for the Corinthian capital.

page nr.	line nr.	Rule transcribed from the treatise's page and line ⇔ **Mathematical translation of the transcribed rule** (using the following variables: D=diameter; H=height; W = width; L = length; M = module, auxiliar variable; d_ = differential)	internal rule numbering
208	06	The height of the Corinthian capital is equal to the diameter at the base of the column ⇔ **Hcapital = Dimoscape**	01
	07	and is divided into seven modules. The abacus takes up one module and the remainder is occupied by the vase, ⇔ **M = 1/7 · Hcapital; dHabacus = 1 · M; dHvase = 6 · M**	02
	08	whose base has the same width as the top of the column, without its projections, ⇔ **W0vase = Dsumoscape**	03
	09	and whose upper rim has the same width as the bottom of the column. ⇔ **W1vase = Dimoscape**	04
		[…]	
	17	The vase is girt with a fillet and an astragal, ⇔ **(rule dependent on shaft and rule #18)**	10
	17	which cover it with two interlapping rows of leaves standing out in relief; each row contains eight leaves. ⇔ **Lleaf =1/8 · Lleafrow**	11
	19	The first row is two modules high, as is the second. The remaining space is taken up by the stalks sprouting out. ⇔ ⇔ **dHleafrow = 2 · M; dHstalkrow = 2 · M**	12
	21	These stalks are sixteen in number; four of them unfold on each face of the capital, two from the same knot on the right, and two from the same knot on the left; ⇔ **Lstalk = 1/16 · Lstalkrow; (shape rule: see scheme)**	13
209	01	the two end ones hang below the corners of the abacus in a form of spiral, ⇔ **(shape rule: see scheme)**	14
	02	while the middle ones also curl, so that their ends meet in the center. ⇔ **(shape rule: see scheme)**	15
		[…]	
	05	Each leaf should be articulated into five or, possibly, seven lobes. ⇔ **Llobe € {1/5 · Lleaf , 1/7 · Lleaf }**	19
	06	The tip of the leaves hang forward half a module. ⇔ **dWleaf = 1/2 · M**	20
	07	As with all carving, deeply incised lineaments will add great charm to the leaves of the capital. ⇔ **(shape rule: see scheme)**	21

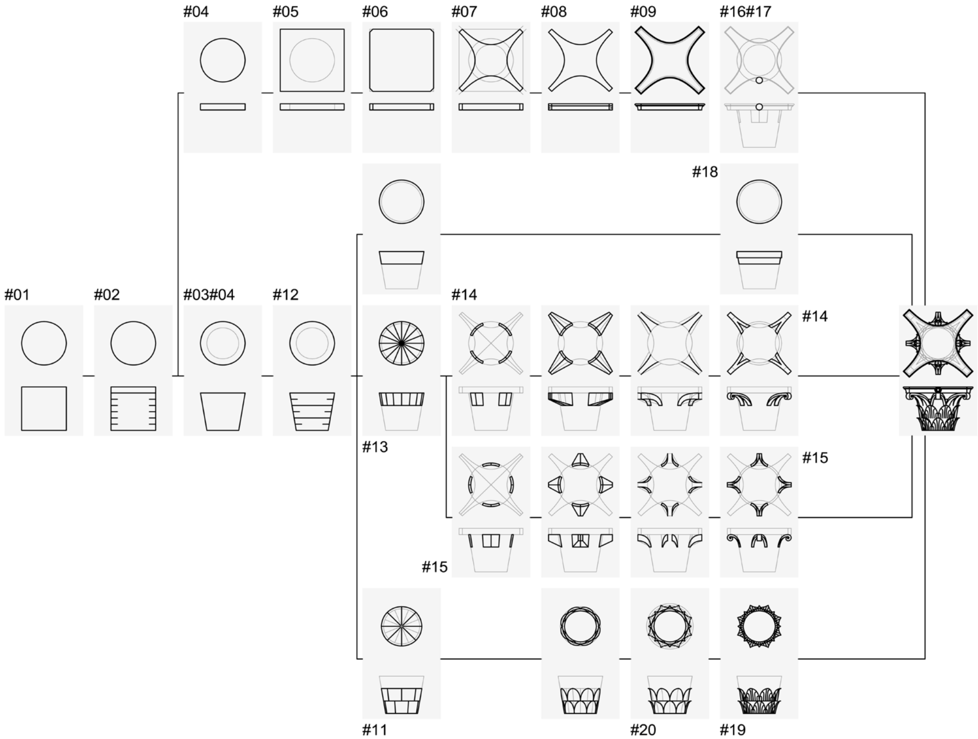

Figure 1
Hierarchical modelling scheme of the Corinthian capital (Castro e Costa, 2012).

gested that the elements of the column system could be organized into hierarchical tree structures, which are represented in the modelling schemes (Figure 1) (Castro e Costa, 2012, p. 50).

Filling in the gaps

However, Alberti is not as diligent prescribing shape rules as he is prescribing proportion and subdivision rules, as they are insufficient to determine the exact shapes of every element of the column system. Therefore, it was necessary to look for solutions outside of the treatise, namely illustrations of later editions of De re aedificatoria, and observation of built examples. Illustrations of other treatises were also consulted, but only aspects not conflicting with Alberti's rules were taken into account. Analysing these sources, it was possible to fill in the gaps, and to model the whole of the parts of the column system.

The Corinthian capital computational model poses as a paradigmatic example of this approach. The formal complexity of some of its elements, like the acanthus leaves or the sprouting stalks (Figure 2), challenged both modelling skills and knowledge

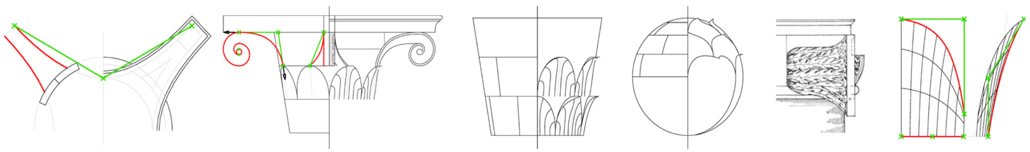

Figure 2
Filling in the gaps for the Corinthian capital: sprouting stalks and acanthus leaves.

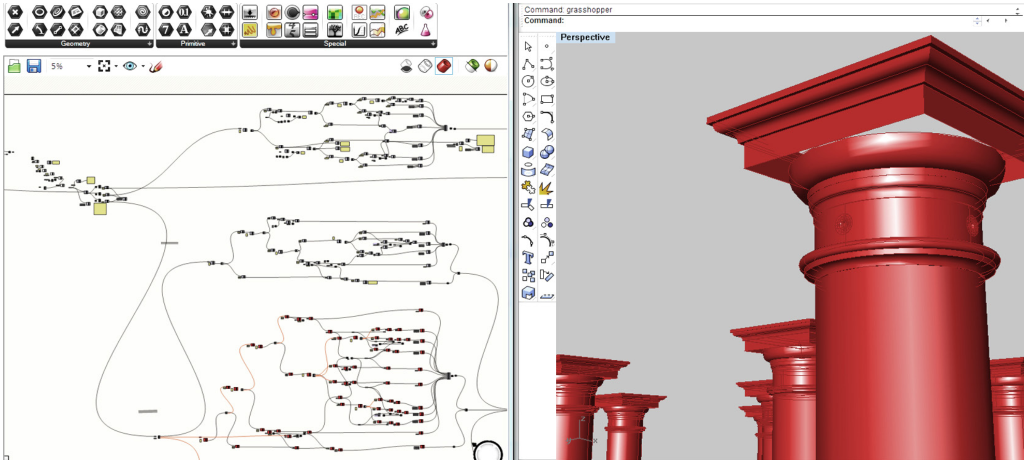

Figure 3
Computational and digital models (Grasshopper and Rhino) of the Albertian column system.

of geometry. Observation of elements both drawn and sculpted was essential for the understanding of such geometries, whose modelling is still being improved.

IMPLEMENTATION OF THE COMPUTATIONAL MODEL

Framework

The computational model was implemented into a computer program through the use of Grasshopper (GH), a visual programming language that interacts with NURBS modelling software, Rhinoceros (Rhino). In GH, programs can be visually developed as instruction sequences that automatically generate digital 3D models in Rhino according to input parameters. The computational model was implemented by translating Alberti's instructions into GH instructions, called components, and feeding Alberti's parameters into these components, creating a computer program that generates any column according to Alberti's canon (Figure 3).

Optimization

As stated before, designing and implementing the computational model were not sequential but intertwined tasks. Each design iteration was tested through a subsequent implementation iteration, which provided clues on how to improve the model. These clues were then fed back into the model in the next design iteration, and so on. The first implementation iteration took place after the analysis of the treatise, and so the systematization process had been partially completed, allowing for a considerable simplification of a significant part of the program (Castro e Costa, 2012).

However, the translation of some of Alberti's instructions into GH components cannot be done directly, namely those concerning the shape of complex column elements, such as the acanthus leaves. On the other hand, even the modelling of simpler elements did nevertheless imply a large number of operations, leading to complex and heavy programs. These GH compositions took a long time to process changes in some parameters, and thus compromising the performance of the models themselves.

Classes

Implementation of the first computational models suggested the need of and potential for optimizing the system's design. On one hand, it became clear that most elements of the column system share common properties, allowing interpreting them as topologically similar entities. That pointed to a pos-

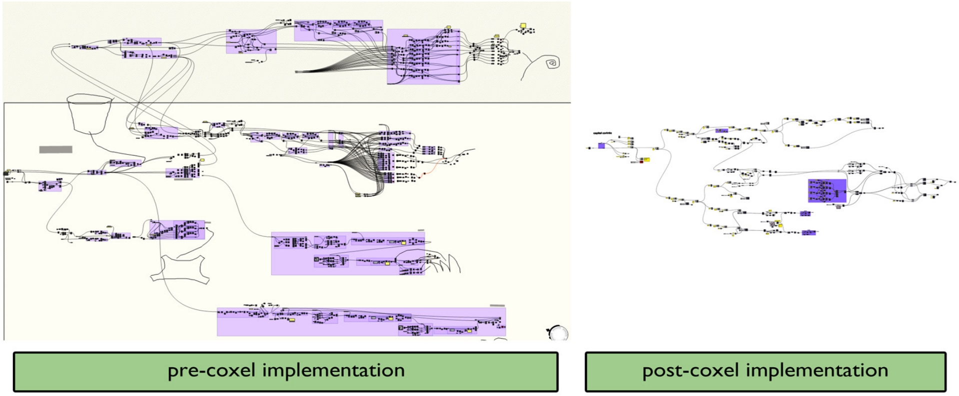

Figure 4
Class implementation as means of simplifying the system.

sible implementation as computational classes. On the other hand, it often happened that the same groups of Grasshopper commands were used repeatedly, pointing to the use of subroutines (Scott, 2006).

Since GH is not prepared for implementing subroutines, the need arose to find a scripting language that would. The selected language was VB.NET, an object-oriented language for which GH has built-in scripting tools, and so a custom class was created, and named *coxel* (COlumn ELement). Implementation of this class allowed both defining relationships more intuitively, as well as reducing the amount of code for the computational models (Figure 4), thus further simplifying the system (Castro e Costa, 2012).

However, there was a downside: since the *coxel* is a new, custom-made type of object, GH had to be informed about how to generate its shape, in other words, how to render it. It was then necessary to gain a deeper understanding of how shape and geometry is represented both conceptually and computationally, since rendering of the elements had to be programmed directly, using RhinoCommon. Although the results were achieved, allowing for the *coxel* approach to be implemented, there is still room for improvement.

DIGITAL MATERIALIZATION

Rapid prototyping and digital fabrication

As stated before, digital models of the column can be generated using the computational model. These digital models can in turn be used to produce physical models of the column elements through digital materialization technologies, comprising both rapid prototyping and digital fabrication (Pupo, et al., 2009). Different technologies were tested, aiming at two main goals: a) to determine the qualities of the previously modelled geometry, and b) to assess the suitability of each technology for producing the different elements of the column system, according to parameters such as shape or size. For the making of the physical models, three materialization technologies were tested: a) Fused Deposition Modeling (FDM), b) Three-Dimensional Printing (3DP), and c) CNC milling.

Additive technologies

As additive materialization technologies, FDM and 3DP proved to be more suitable for smaller models that require more detail. Several smaller models were produced using FDM, namely some test models, and a collection of small scale models of all the elements of the column system is currently under production (Figure 6). This miniature collection will

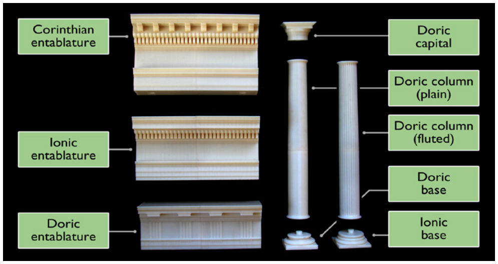

Figure 6
Collection of miniature FDM produced models of the column system.

feature in the exhibition as an interactive installation, demonstrating the combinatorial nature of Alberti's column system (Coutinho, et al., 2011). In fact, the collection was developed so that elements of different types can be combined. For example, a column with a Ionic capital may feature a Doric base, and be topped by a Corinthian entablature.

Compared with FDM, Three-Dimensional Printing (3DP) technology features a higher modelling resolution, allowing for the production of more detailed objects, rendering it especially suitable for models that feature geometry curved in more than one direction. The Corinthian capital is an example of such geometry. Figure 7 shows instances of the Corinthian capital produced through each of the two technologies. The FDM model is less developed in terms of shape, and was printed at half the scale of the 3DP model. Nevertheless, the problems caused by the anisotropy inherent to the FDM printing process (Ahn, et al., 2002) were detected in the first case, and the computational model was edited in order to correct the problem.

CNC milling

Production of physical models featured the fabrication of a full-sized Corinthian capital in stone through a subtractive fabrication technology, namely a 6-axis CNC milling machine. The success of this experiment resulted from the collaboration with a Portuguese stone-cutting company, which provided a good insight on the problems that arise within an industrial context.

One of the main limitations in CNC milling concerns the existence of inaccessible regions. These are regions in the model that will not be accessible by the milling tool like, for example the inside of a hollow sphere. This is a process-related, unavoidable limitation (Lennings, 1997). During production of the Corinthian capital, this problem was identified

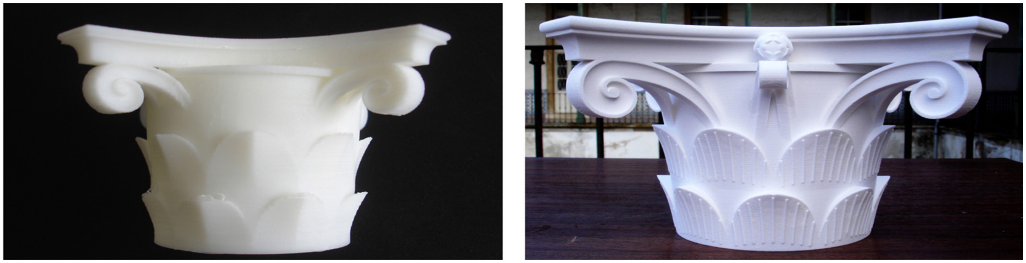

Figure 7
Corinthian capital models produced through additive prototyping technologies.

Figure 8
Five production phases of the Corinthian capital.

Figure 9
Final capital stone model.

but not addressed. However, it is planned to be minimized in a future model, both at production and design stage.

Production of the stone capital comprised 5 different phases (Figure 8). Each phase featured an increased level of detail from the previous one, derived from the tool used in each phase. In total, production took 15 hours.

Usually, milled stone models go through post-production phase, in which they are manually polished, and eventual production artifacts are corrected, also by hand. However, since one of the objectives of milling the capital was to evaluate the suitability of CNC milling technology, namely in terms of surface detail (Figure 9), post-production was skipped. Therefore, it is possible to observe the artifacts derived from the existence of inaccessible areas.

The production of this model is an example of the potential of establishing collaborations between academic research and industrial activities. Following the production of the Corinthian capital, further production is planned, as well as further development of the computational model according to the needs of the stone-transforming company. The ultimate goal is to add to the company's and competitiveness, increasing productivity through innovation.

CONCLUSION

Regarding computational modelling, the implementation of the parametric generative model followed

a thorough analysis and an effort to better understand and organize relationships between the parameters to be implemented, as well as to represent shape, in this case the shapes of Alberti's column elements. Future research will focus on further optimization of the relationship system, as well as on development of a user interface. The achieved results are expected to aid in the development of the column system shape grammar.

In terms of physical modelling, fabrication of the Corinthian capital poses as a case study of a successful collaboration between an academic research project and an industrial company. This collaboration is expected to be extended with the production of a composite capital, larger than the capital presented above and with a different kind of stone.

ACKNOWLEDGEMENTS

This work is funded by FEDER Grants through COMPETE – Programa Operacional Factores de Competitividade and by National Grants through FCT – Fundação para a Ciência e a Tecnologia, as part of the "Digital Alberti" project (PTDC/AUR-AQI/108274/2008 – FCOMP-01-0124-FEDER-008842). The project is hosted by CES at the University of Coimbra and coordinated by Mário Krüger. Filipe Coutinho is funded by FCT with PhD grant SFRH/BD/66029/2009.

REFERENCE

Ahn, SH et al., 2002. Anisotropic material properties of fused deposition modeling ABS. *Rapid Prototyping Journal*, Volume 8 issue 4, pp. 248 - 257.

Alberti, LB, 1988. *On the Art of Building in Ten Books. Trans. Engl. of De re aedificatoria*. (J. Rykwert, N. Leach, & R. Tavernor, Trans.) Cambridge (Massachusetts): MIT Press.

Alberti, LB, 2011. *Da arte edificatória*. (A. E. Santo, & M. J. Krüger, Trans.) Lisbon: Calouste Gulbenkian Foundation.

Castro e Costa, E 2012. *Modelação computacional e materialização digital de elementos clássicos de arquitectura: Sistematizando a coluna de Alberti*. Dissertação de mestrado. Lisboa, Portugal: Faculdade de Arquitectura da Universidade Técnica de Lisboa.

Castro e Costa, E and Coutinho, F and Duarte, JP and Krüger, M 2011. *The Corinthian Capital according to Alberti: Generative modelling and digital prototyping of classical architectural elements*. Leiria, Portugal, s.n., pp. 255-262.

Coutinho, F Castro e Costa, E., Duarte, J. P. and Krüger, M., 2011. *A computational interpretation of De Re Aedificatoria: translating Alberti's column system into a shape grammar*. Ljubljana, Slovenia, s.n., pp. 788-798.

Duarte, JP, Celani, G. and Pupo, R., 2011. Inserting computational technologies in architectural curricula. In: *Computational Design Methods and Technologies: Applications in CAD, CAM and CAE Education*. Hershey, PA, USA: IGI Global Publishing.

Lennings, L 1997. *Selecting Either Layered Manufacturing or CNC Machining* to. s.l., Rapid Prototyping Association of SME.

Pupo, R and Celani, G and Duarte, J. P., 2009. *Digital materialization for architecture: definitions and techniques*. s.l., s.n., pp. 439-442.

Scott, ML 2006. *Programming Language Pragmatics*. s.l.:Elsevier.