Towards Intelligent Control in Generative Design

Manuel Muehlbauer 1, Andy Song 2, and Jane Burry 2

1 RMIT University, Melbourne 3000 VIC, Australia, manuel.muehlbauer@rmit.edu.au, 2 RMIT University, Melbourne 3000 VIC, Australia

Abstract. This position paper proposes and defines the nature of a framework, which explores ways of integrating control system (CS) with machine intelligence for generative design (GD). This paper elaborates about the implications of and the potential for impact on GD. The framework described in this work can be used as an active tool to drive design processes and support decision making process in early stages of architectural design. This type of system can be either automated in nature or adaptive to regular user input as part of interactive design mechanisms. The module of CS in the framework would allow additional guidance during design and therefore reduce the need of manual input to enable a semi-automated design practice for lengthy generative processes. This study on GD reveals emergent properties of the framework, for example the introduction of intelligent control allows guidance of GD to meet specified performance criteria and intended aesthetic expressions with reduced need for user interaction.

Keywords: Semi-Automated Design, Evolutionary Architecture, Generative Design, Architectural Optimisation, Artificial Intelligence

1 Introduction

As technological progress accelerates, there is an unprecedented growth in information [12] that provides opportunity for the establishment of contemporary design paradigms, generative design being one of them. While the impact of emergent innovative technology on design processes has a significant history in architecture [21], [34], [61], Mitchell predicted [41] that the extension of the use of computational design systems will change the way we practice architecture in radical ways and increase in computational power will transform the industry. In brief, the presented position paper elaborates on one trajectory of technological disruption that possesses the potential to transform the nature of GD to utilize it as support tool for decision making in architectural design.

Machine intelligence can be applied to architectural design in one of the following categories: (1) analytic processes feeding into predictive analysis to learn from realized designs for the design of future instances of a similar typology or design case, (2) intelligent building control, coined smart buildings, adding control features to architectural technology or kinetic features in buildings, or (3) intelligent control of
generative processes, usually in form of optimization of potential design solutions towards specified performance criteria – coined morphogenesis in architecture [40], [50]. Because of the limitation of architectural optimization on the variation of decision variables, the proposed framework focusses on the integration of artificial intelligence technology for grammatical evolution of design solutions to increase the emergent design potential during architectural optimization.

In the same way, generative structural design of shell structures [5], [48] and structural nodes [13], [24], [45], [47], [62] may be improved. Form-finding for optimal structural solutions is addressed by two main streams: discrete structural optimization [7], [18] and continuous structural optimization [17], [64]. Even interactive structural optimization processes were explored in the context of parametric design environments [14], [27], providing invaluable insights on the limitations of parametric design and interactive exploration of the associated design space. While an exploration of these areas would be promising for application of the framework, it is beyond the scope of the provided argument to extend on the considerations related.

In this paper, we describe the framework for the development of a CS for GD based on machine intelligence and interactive evolutionary computation [39], [58]. The key aspects discussed are:

1. the intelligent design framework for the integration of a novel interactive strategy to enhance semi-automated design generation
2. the integration of a preference-based fitness function in the evaluation process of the search process in the context of early design in architecture
3. the potential improvement of decision making in GD based on the guidance procedure that allows the designer to navigate the optimisation process

2 Background

For a review and comparison of other suitable generative algorithms that could be used as module in the proposed intelligent design framework, refer to the work of Singh & Gu [54]. Furthermore, the referenced work provides evidence on the use of a combination of different GD methods. Subsequently, different GD methods could be applied to advance the design to the next design state with different contextual specification.

The basis of the argument is the necessity to address designer’s intent in automated design processes. Therefore, the GD process described by Mitchell [41] and presented in Fig. 1 is used as a foundation for the reasoning. This computational design process incorporates the cyclical nature of design tasks and differentiates essential activities during the design process. To summarize this approach, the design problem is analysed and structured to the current knowledge of the design process before a data structure is developed and implemented as a representation of the design problem.
Performance criteria, decision variables, constraints, parameters, geometric attributes and topological relationships are defined by the design case and specified for the implementation.

In the next step, the generative process is initialized and evaluated based on performance criteria. The choice of those criteria frames the decision-making process and reflects the design goals. Thus, the simulation methods used to evaluate the design and the decision variables are to be defined in the context of the desired trade-off. At this stage human and computational resources available for the GD need to be considered, especially when using expensive fitness evaluation. One key contribution of the intelligent control module outlined, is the reduction of the user and simulation effort during lengthy processes, while providing maximum flexibility for the designer to provide additional non-solution input.

2.1 Structuring Design Problem

Structuring design decisions is a crucial task during design processes. As tools frame design processes during computational processes, the choice of representation is crucial for the outcome of the process. The impact of the choice of representation used to describe design spaces is extensive and defines the set of phenomena we are interested in [15]. At this step, architectural expertise is required to define the hierarchy of functions, variables and parameters for a specific design case [2]. Based on the suggestion of Akin & Dave [2] to rethink those processes of representation construction as part of a control system, some aspects of the hierarchical assembly can be constituted based on the flexible representation of genetic programming.

It has been suggested that rule-based systems are an efficient way to encode architectural shapes [10]. These generative systems exhibit a range of degrees of emergence and volatility, depending on the set of rules or behaviours chosen to define the bottom-up process. Prior to the introduction of CS, the use of grammatical rules [6] and the generation of those rules [35] needed to be fully automated to efficiently exploit the emergent potential of GD.
The context of shape generation in architecture is broad with methods ranging from shape grammar [23], genetic algorithm [38], genetic programming [11], [59] and grammatical programming [10] to grammatical evolution [6], [44]. Even if all those methods allow for the introduction of intelligent control, the presented framework focusses on the use of genetic programming and shape grammars.

The definition of performance criteria is dependent on the design task and specific to the application case. Three main decisions need to be taken:

1. Specification of the criteria used for the performance evaluation
2. Definition of constraints used as thresholds of the criteria’s domains
3. Goal weighting for the performance evaluation, specifying the relative importance between the criteria during the trade-off process

As design problems are wicked or ill-defined in nature, the result of the structuring process might be a preliminary trajectory of investigation that is used to generate knowledge about the design problem. In this case, the process would result in a more refined formulation of the representation in the following design state. In other cases, the optimisation process already leads to a computational model representing a feasible design solution.

### 2.2 Defining Representations

The qualities integrated in the representation of architectural design are of either geometric, topological or contextual nature [2]. Those aspects of the architectural problem need to be specified and encapsulated in the representation to define the design space. The representation is dependent on the chosen process of shape generation. This research paper focuses on the discussion of shape grammars and grammatical evolution [44], [46], because the multi-facetted nature of design problems suggests their efficiency in use for performance-driven design to define architectural shape, building features and plan layouts.

Grammar-based models of the representation of architectural shapes [1], [43] were explored in depth [19], [28], [57], [63] and the related challenges reported in other research in this area [29], [55] will be addressed by the proposed intelligent design framework. The lack of efficient control of shape grammar systems during interactive design processes is addressed through the introduction of a guidance mechanism controlling the periods of automation in broad interactive approaches will increase the efficiency of GD and reduce human effort.

During the control of GD, interaction between the artificial intelligence and human designer is facilitated through the common evaluation of the genotypic (representation and grammar rules) and phenotypic (interactive, multi-criteria optimization) expression developed during the framework development. Further insights, gained by reflection on the relationship between architectural genotype and phenotype [42] will inform the ongoing speculation promoted through the development of the proposed framework.
The combination of the use of flexible representation based on genetic programming for automatic shape generation [44] and control of the grammatical parameters and rules as described by Byrne [6] and Lee [35] are basis for the conceptualization of the presented framework for integration of intelligent control [65]. It will contribute through the introduction of additional strategic potential for support of decision making in early design stages and a novel mode of interactivity in GD using non-solution input as a reference for the guidance mechanism contributing to the fitness evaluation, which will be discussed in section 4.

2.3 Performance Evaluation

Numerous authors have argued that integration of simulation and optimization in architectural design generate novel designs, while simultaneously leading to significant performance improvements [3], [4], [9], [53].

The integration of performance simulation and multi-criteria optimization is facilitated based on the modular nature of the intelligent design framework. A variety of possible combinations of criteria could be introduced and some will be explored in future experiments. Repeated computational experiments to review the performance of design solutions are described by Burry and Burry [8] as virtual prototyping. The presented research extends this notion that is closely associated to parametric design setups for optimization in architectural practice, towards a semi-automated design practice to overcome the limitations associated with current visual programming environments.

2.4 Interactive Design Synthesis

The main contribution of the proposed intelligent design framework for digital morphogenesis using semi-automated design is the introduction of a preference-based control system using reference input to constantly reflect the preferences and requirements of decision makers that frame architectural design processes.

As the generated shapes by themselves contain no meaning when the process is initialized based on a random population, after the first iteration already a common creation of meaning takes place inside the framework. This creation of meaning is provided by interactive evaluation and the novel input mechanism on one side and the automatic evaluation of multiple criteria on the other side. Then, constraint satisfaction and semantic structure of the solutions are tested to make sure that the solution reflects the scope of the design problem and addresses all thresholds and limitations imposed on the decision-making process. Therefore, the interactive evaluation during every generation addresses only the solutions that are suitable as solutions to reduce user effort and computational expenses during fitness evaluation.

Performance-based GD related to Morphogenesis [31], [50], [51] will benefit from the adaptation of intelligent control, extending the designer’s influence on the strategic exploration of the design space. Furthermore, the discourse about second-order cybernetics in architecture and the associated discussion about control of feedback loops through the designer, as intensively discussed by Thomas Fischer [20]
will be revived through the reasoning on concepts of intelligent control and human-computer interaction.

The described framework takes the step from GD and semi-automated design using a CS for intelligent decision support during GD, as a novel way of human-computer interaction in early design stages of architectural design. In conclusion, this position paper speculates about a design framework through the introduction of intelligent control into decision making processes based on interactive multi-criteria optimization.

3 Research Methodology

The intelligent design framework builds on critical review of GD in architecture, engineering and computer science and simultaneously evolves theoretical considerations, implementation of active design tools based on system design methodology and strategic design exploration in form of case studies to reflect on the implications of the work on creative practice. The main modules of the framework are presented in Fig. 2. The green modules are already implemented, the module marked blue is currently under development, while the red modules will be implemented as part of the intelligent design framework. The goal of the study is to reduce designer fatigue during interactive GD by introduction of a CS alongside conventional fatigue reduction methods.

![Fig. 2. Modular structure of intelligent design framework](image)

3.1 An Intelligent Design Framework for Semi-Automated Design

The intelligent design system proposed in this research could be applied in a variety of different design-related fields, e.g. engineering, interior design, landscape architecture. The validation of the design system for architecture based on the exploration of various case studies is inside the scope of the research project and will be explored in the next iteration of the process. The implementation will be discussed in section 4 of this paper.
Basis for the development of the framework is a GD that is based on parameters, and rules or behaviours to build solutions computationally during a bottom-up process. In this context, artificial selection is an invaluable asset for the introduction of expert knowledge into the design process.

A point often overlooked, is that in the period between user inputs during a broad interactive approach, the generative process is fully automated and could be extended by introducing an additional guidance mechanism. In fact, this CS evaluates aesthetic qualities of the design alongside other specified performance criteria. Therefore, it addresses the core aspects of architectural design – form and function – about subjective preferences and allows GD to evaluate design solutions towards the codes of utility and beauty constituted by Patrick Schumacher [52].

The CS is integrated into GD as part of the Designer Interaction Module as shown in Fig. 3. Here, a non-solution input is provided during initialization of the GD as reference for the controller to evaluate subjective preferences as part of the utility function. This research contributes to the architectural discourse in the context of GD with the proposal of additional guidance during lengthy design processes to minimize human effort, while increasing the amount and quality of user input to guide GD.

### 3.2 Intelligent Design System

For control of GD, an intelligent system is developed that uses either images as two-dimensional input or mesh geometries as three-dimensional input to guide the search process based on a set of features extracted from the input data.

To the end of a more direct and creative input to the design process, the reference introduced to the GD is continuously evaluated to provide additional robustness and stability. This is achieved by a constant comparison of the features extracted from input and output geometry to calculate a distance measure used as part of the fitness evaluation.
The intention is to provide an aesthetic guidance mechanism that allows the designer to introduce specific aesthetic qualities through the specification of a design reference. In addition, a multi-objective trade-off process integrates the aesthetic guidance with performance optimization towards specified criteria.

Fig. 4. Intelligent real-time CS for GD

The intelligent CS performs in a variety of calculations in real-time during the discrete time steps of the GD that works as a closed-loop system. A conceptual diagram of the CS is presented in Fig.4. In the next section of the research, we will discuss the agents, means and ends of the CS in greater detail.

3.3 System Identification Using Genetic Programming

The controller is based on a non-linear input-output model, because GD exhibits a non-linear system response that shows a discontinuous fitness landscape. Genetic programming systems as population-based methods fit well with complex fitness landscapes used to define the structure of a robust controller in a tree-based representation [37]. The application of CS in design systems can be used to integrate a reference to the performance requirements, system constraints and customer specifications of the specified design case [36]. Application to texture classification provides initial insights on possible implementation strategies [56] and considerations about aspects leading to increases in efficiency of image classification.
3.4 Learning and Knowledge-Preservation

The knowledge generated implicitly in the generative process is implicitly stored in the grammar representation used for the description of the solution. It consists of two connected aspects of the representation. On the one side the topological encoding of spatial relationships in the data structure and on the other side a feature-based encoding of the building components. Desired features can be enforced through their selection during the input process.

A mesh representation can be chosen for the development of the solution during the process, so that geometrical information about the shape and features of the design are stored in the representation directly. Therefore, any generative algorithm that can generate polygon meshes may be integrated into the framework. This link allows the use of different GD for distinctive design states and thus more creative freedom to the computational designer.

3.5 Genetic Programming

Genetic Programming is an evolutionary computing method first developed by Koza [33] that exhibits enormous flexibility in representation and can be used in a variety of tasks. It was used for shape generation [44], optimization [60] and space layout planning [26].

The symbol-based evolutionary system can be used to evolve computer programs and in this respect, evolution of controllers and classifiers are part of the systems capabilities. In the context of this research project, GP is used to implement the GD process based on shape grammar and genetic programming [44], [46].

The GP algorithm is based on an iterative, generational process that is structured in several steps. After generating an initial random population, a selection process takes place based on the evaluation of a fitness function. The selected individuals are then modified by the genetic operators of recombination and mutation. In a separate process, the best solutions of the generation are preserved by elitism. During the advancement of the generation, a new population is constructed based on the mated, mutated and preserved solutions.

A diagram of the implementation approach of the framework based on grammatical evolution of a strongly-typed grammar is shown in Fig. 5.

3.6 Implementation of Guidance Mechanism

An image upload allows the designer to upload a reference image. The features of the image are extracted using the SIFT algorithm available in the OpenCV library to identify the key points of the image.

During GD, the solutions generated are rendered as image files and the features matched to those of the uploaded image. This process allows to specify a distance measure that contributes to the fitness evaluation. It relates the reference image to the generated solution and guides the GD along the implicit aesthetics.

The necessary pre-processing of the images for feature extraction consists of resizing, blurring and omitting the colour information of the images. Through this process, the performance of the feature extraction algorithm is enhanced tremendously. The steps from source image (1) over blurring (2) and key point detection (3) to feature matching (4) are shown on an example in Fig. 6.
3.4 Learning and Knowledge-Preservation

The knowledge generated implicitly in the generative process is implicitly stored in the grammar representation used for the description of the solution. It consists of two connected aspects of the representation. On the one side the topological encoding of spatial relationships in the data structure and on the other side a feature-based encoding of the building components. Desired features can be enforced through their selection during the input process.

A mesh representation can be chosen for the development of the solution during the process, so that geometrical information about the shape and features of the design are stored in the representation directly. Therefore, any generative algorithm that can generate polygon meshes may be integrated into the framework. This link allows the use of different GD for distinctive design states and thus more creative freedom to the computational designer.

3.5 Genetic Programming

Genetic Programming is an evolutionary computing method first developed by Koza [33] that exhibits enormous flexibility in representation and can be used in a variety of tasks. It was used for shape generation [44], optimization [60] and space layout planning [26].

The symbol-based evolutionary system can be used to evolve computer programs and in this respect, evolution of controllers and classifiers are part of the system's capabilities. In the context of this research project, GP is used to implement the GD process based on shape grammar and genetic programming [44], [46].

The GP algorithm is based on an iterative, generational process that is structured in several steps. After generating an initial random population, a selection process takes place based on the evaluation of a fitness function. The selected individuals are then modified by the genetic operators of recombination and mutation. In a separate process, the best solutions of the generation are preserved by elitism. During the advancement of the generation, a new population is constructed based on the mated, mutated and preserved solutions.

A diagram of the implementation approach of the framework based on grammatical evolution of a strongly-typed grammar is shown in Fig. 5.

Fig. 5. GD based on Grammatical Evolution

Grammatical evolution is integrating the evolution of the rules or behaviours of GD into the evolutionary process based on either a string presentation or linear GP. Different combinations for the application of the rules or behaviours in a stages manner are explored during the co-evolution with the shape grammar representation, providing the data of the topology-based plan layout and the associated building features that describe a design space, usually in the scope of style or typology.

3.6 Implementation of Guidance Mechanism

An image upload allows the designer to upload a reference image. The features of the image are extracted using the SIFT algorithm available in the OpenCV library to identify the key points of the image.

During GD, the solutions generated are rendered as image files and the features matched to those of the uploaded image. This process allows to specify a distance measure that contributes to the fitness evaluation. It relates the reference image to the generated solution and guides the GD along the implicit aesthetics.

The necessary pre-processing of the images for feature extraction consists of resizing, blurring and omitting the colour information of the images. Through this process, the performance of the feature extraction algorithm is enhanced tremendously. The steps from source image (1) over blurring (2) and key point detection (3) to feature matching (4) are shown on an example in Fig. 6.
During the GD, the user selects solutions as part of the fitness evaluation after a defined number of generations to adjust the guidance mechanism. This information is used for labelling a continuously growing set of training data for a Naive Bayes classifier.

3.7 Classifier

During the last decades, machine learning experienced a growing attractiveness in the scientific discourse and many new applications were developed. In the context of architectural design and engineering the use of expert systems was superseded by the introduction of classifier technology [49], because of the learning capacity of the system.

In machine learning, the term learning describes the capability of an agent to adapt its behaviour to an environment through an increase in knowledge. Therefore, performance improvements of the system are achieved over time based on the exposure of the agent to specific environments [32].

A classifier performs in different modes. In learning mode, a training set of properly labelled data is used to provide the classifier the possibility to learn the relationship between data and the labelling approach. In the context of this research, aesthetic criteria are addressed through the constant labelling of the solutions chosen by the designer as “desired” and “not desired” to train the classifier. The collection of
the dataset for the classifier is continuously collected during the ongoing process, so that the evolution of the design can be adjusted based on the choices made during the artificial selection process.

![Diagram of Generative Design Process](image)

**Fig. 7.** Schema of a classifier integration

After successful training of the classifier, the CS is presented with the validation data set in application mode. Based on the learnt behaviour, the classifier evaluates the data and continues the learning process. The selection solutions presented for user evaluation is facilitated by the continuous update of the learnt model/classifier, so the system can be adaptive to changes hence more intelligent and requires less manual input. A schematic description of the process is given in Fig.7.

The Naïve Bayes classifier can work with comparatively small sets of training data and deal with noisy data efficiently. As the training data set might not be generated consistently in reflection of rational principles, but handled intuitively, the choice of classifier allows to accommodate for changes in subjective perception during the generation of solutions.

## 4 Framework Discussion

The focus on early stages of architectural design frames the following speculation about the implications of and the potential for designers to use the proposed framework during briefing, design and planning processes. During those decision-making processes, the design framework can be used to explore a variety of design solutions for goals with different weighting on performance criteria.

Linking to the tradition of automated design in architecture, the section is structured analogous to the seminal work of Mitchell [41] in the conclusions of his publication.

The modular framework for intelligent design for semi-automated design increases the efficiency of interactive GD through two modules. One is a similarity criterion that reduces the number of solutions that are reviewed by the designer and the other one a CS that introduces guidance based on aesthetic evaluation of the generated solutions.

As the learning process of the classifier is computationally expensive, a parallel implementation of the process might increase the efficiency of the decision support tool. In this context, the computation of expensive fitness functions could also be...
outsourced to a computer cluster to further improve GD. Extensive studies with finite elements (FE) or computer fluid dynamics (CFD) will gain more relevance, as the CS will help to reduce optimisation time by limiting the amount of solutions computed.

Keeping this scenario in mind, the proposed intelligent design framework can be used to provide guidance to these lengthy processes, which usually either prevent the feasibility of human interaction completely or accept delays based on the waiting time of the computational system, if the designer is not present.

User fatigue is one of the main concerns that are present in the current discussion about interactive GD. The proposed framework increases the efficiency of the GD and reduces the amount of manual work that is necessary for the designer to input through an increase in automation of the interactive aspect.

Another main aspect of the framework is the aesthetic guidance mechanism that extends the designer input and guidance potential during GD. The relevance of this aspect in based on the assumption that the designer using the intelligent design framework uses a rational decision making process based on criteria that can be either quantified or consistently applied during artificial selection.

5 Conclusion

In the context of architecture, multiple complex systems can be combined to generate a holistic design solution. Often a variety of experts participate in the development of sub-systems that contribute to the building performance. The modularity of the GD process that can be achieved based on a unified representation can facilitate a multi-level GD process that could be used to generate solutions for specific sub-systems in the context of a larger model.

Because of increasing costs and rising awareness for the climatic challenges of our times, performance requirements for buildings are constantly improving. Therefore, optimization processes will become mainstream technology in future design processes. Even if some performance criteria can be automated, the unique knowledge of architects about the philosophical, social and cultural aspects of design will not be automated. Semi-automated design processes using intelligent technology to provide means for interaction for designers during GD, integrate the extensive knowledge of the architect with the computational capacity of automated systems to evaluate complex trade-offs between conflicting criteria.

In early design stages, the specification of fabrication constraints, case-specific and site-specific requirements and stakeholder preferences are still in development. Thus, designer input is necessary to dynamically add those specifications using the designer’s intuition. The proposed intelligent CS allows the externalization of some of these aspects through the specification of a reference input that incorporates an aesthetic expression, specific features and qualities in its representation.

In GD, the acceptance of generated solution by the designer is the main criterion for the application of optimization results. Additional input to the GD increases the potential acceptance, as the decision-making process is on side of the designer, while the GD supports the decision making through the provision of strategic data, the
potential to explore different combinations of goals and criteria and the definition of a wide range of constraints on the representation, grammar and parameters.

6 Future Research

Future work will explore the application of a mesh representation for the generative process to store geometric features directly in the representation that is evolved during the grammatical evolution. This approach creates the possibility to use the output of one GD process as input for another one that modifies the mesh representation again. As an implicit knowledge storage mechanism, design decisions made during the development of one design stage can be transferred directly to the process used during the following design state.

The presented intelligent design framework extends strategically the application potential for interactive features of GD for architectural design. To these ends, a mechanism for guidance by aesthetic evaluation is integrated into the semi-automated search process for case-specific [22] and site-specific design. Multiple ways of implementation for a CS can be used to improve the efficiency of GD. While the presented framework builds on classifier technology with a medium level of complexity, less complex approaches like model-based and data-driven control mechanisms may achieve different results that might be more desirable in the context of specific design cases.

The proposed speculative framework can be applied in a variety of computational design methodologies, from diagrammatic approaches e.g. in space layout planning [26], [30] or activity representation [16] to three-dimensional shape generators [44]. In parametric design, the application of the framework might increase the efficiency of recently developed generative tools for shape generation, e.g. the genetic programming-based active tool implemented by John Harding [25].

In conclusion, the proposed intelligent design framework reveals potential for better performance of interactive design processes using GD by increasing the efficiency of user input. The main contribution of the framework is the use of non-solution input during the initialization of the GD process and the intelligent control of the GD based on feature extraction of the reference provided during this input.

References

20. Fischer, T.: Designing (tools (for designing (tools (for …))))), RMIT University, Melbourne (2008)
36. Li, Y.: Performance-based control system design automation via evolutionary computing (2001)


