Computational Synthesis

From low level building blocks to high level functionality

Computational Synthesis research seeks formal algorimic procedures that combine low level building blocks or features to achieve given arbitrary high-level functionality. The main challenge is scaling synthesis algorithms so that they can achieve complex functionalities, and the paths of investigation deal with automatic composition of building blocks into useful modules, automatic abstraction of module functionality, and automatic hierarchical reuse of modules. This collection of papers brings together research presented and discussed at the 2003 AAAI spring symposium on Computational Synthesis. The symposium attempted to focus on domain-independent methods that address modularity, regularity, hierarchy and abstraction in synthesis. Recently there has been a surge of interest in these fundamental issues from three directions: AI researchers interested in scaling discovery processes, engineers interested in fully automated design, and, not the least, biologists interested in the origin of complexity. In organizing this symposium we tried to bring together researchers from these diverse fields to exchange ideas about common fundamental questions.

Design invention is perhaps one of the greatest feats of human intelligence, and it is not surprising that it would provoke a multidisciplinary challenge across engineering and AI. As an example, imagine a construction set at your disposal: Links, wheels, wires, actuators, sensors and logic are your atomic building blocks, and you must find a way to put them together to achieve a given high-level functionality: A machine that can move itself, say. You know the physics of the individual components' behaviors; you know the repertoire of pieces available, and you know how they are allowed to connect. But how do you determine the combination that gives you the desired functionality? While the example above is confined to electromechanics, similar synthesis challenges occur in almost all engineering disciplines: Circuits, software, structures, robotics, architecture, control, MEMS, and biology – to name a few.

Many current design automation paradigms focus on specific domains, make use of elaborate domain knowledge and domain-specific algorithms, and are typically limited to a single level of optimization. This symposium emphasizes universal methods that address open-ended conceptual synthesis in a largely unconstrained design space. Are there fundamental, computational properties of design synthesis that cut across engineering fields? What is needed to allow synthesis algorithms scale to complex tasks? Can a computer ultimately augment or replace human invention?

The question of automatic design synthesis has had a long tradition in AI. There have been several demonstrations of systematic synthesis systems using case-based reasoning and rule-based design. At the intersection of Artificial Intelligence and Engineering, much work has been done in providing a representational foundation for engineering knowledge. Ideas relevant to describing the architecture of a component can be found in the framework of *functional representation* research. The knowledge-based approach is useful when the problem domain is specific, and when concise action/consequence knowledge is available. It is less appropriate when knowledge is not available or is too broad, or when we seek unforeseen solutions which are not direct consequence of the rules we key-in. Also, not only do we need to find a way to collect and represent all this knowledge (and this knowledge will be outdated by the time we complete, if ever), but the history of AI has presented us with ample evidence that rule-based systems are difficult to scale. In this symposium we seek to bring together many of these existing ideas with new approaches that address scalability and reduced dependence on domain knowledge.

Scope of topics

This symposium considered submissions that covered all of the different levels of computational synthesis methods, starting with the low-level building block manipulation, up to high-level conceptual reasoning. We especially sought general methods that are domain independent (but might be demonstrated in a particular domain), that explicitly address the question of scalability: Reaching high-level tasks from low-level building blocks.

There are a number of new ideas addressing these questions, which are being investigates across a wide scope of science and engineering fields. The question of efficient synthesis of complex forms from elementary building blocks is also a fundamental question in life sciences. There are several key common issues:

A. **Automatic module discovery:** Methods for aggregating low-level building blocks into higher-level modules, or decomposing an ad-hoc solution into useful sub-modules, so that these modules, in turn, can serve as building blocks for higher-level search. Automatic module discovery has been a focus in AI, design, and biology. New approaches to discovery of modularity include models of symbiosis and co-evolution, game theory, cooperation between agents, and pattern recognition.

B. **Regularity and hierarchy:** Another key ingredient is the question of efficient composition of building blocks into large complexes. Regular structures, hierarchical structures, symmetry, duplication, self-similarity and minimization of information content are important aspects of making large complexes constructible, manageable and the construction process scalable.

C. **Automatic abstraction and encapsulation:** For synthesis processes to scale, it is necessary to be able to abstract and encapsulate lower level modules so that the overall number of free parameters remains constant as we reach higher levels. The abstraction problem is a long-standing question in AI, but in context of synthesis it takes a better-defined form. Dimensionality reduction, linearization, hidden parameterization, are possible approaches. While automatic abstraction is of less interest when a design is embedded in reality (as in nature), it is crucial when simulations are involved, and has been a key to the development of many design automation areas such as VLSI.

Other topics of interest included:

1. Models of bottom-up composition and top-down decomposition
2. Scalability of composition processes to high complexities
3. Automatic identification and composition of useful modules
4. Regularity and hierarchy in composition
5. Automatic abstraction and encapsulation of modules
6. Efficient and adaptive representations of design space
7. Solution-neutral goal specification
8. Stochastic, game-theoretic and evolutionary design processes
9. Machine learning in design automation

10. Computational synthesis as models for discovery in nature and engineering
11. No free lunch: What can we trade to get open-ended design

Other similar meetings in the past

No prior meeting has been dedicated to fundamental questions automatic synthesis. There are several conferences on design automation (e.g. ASME Design automation (mechanical), IEEE Design automation for VLSI), but these are typically domain specific and do not systematically address universal issues. There are also annual conferences on evolutionary computation; however here we wish to examine a wider scope of ideas for synthesis, including pattern recognition, agent cooperation, and search. Finally, meetings on biological complexity do not address applicability to engineering design. No meeting that we know of focus on generic synthesis using modularity, hierarchy, regularity and abstraction.

Target audience and symposium goals

The symposium had two primary objectives: The first is the dissemination of research results from the small but growing community of researchers working in this area. The second is to establish a research agenda for this emerging research field, and frame the critical questions. Unfortunately, many of the researchers in synthesis belong to separate domain-specific communities. Despite its universal nature and the strong correlation in the type of problems encountered, interaction is minimal. We hope that by bringing together researchers from diverse fields we will be able to bridge some of these efforts.

The organizers

Hod Lipson (Cornell University)
Erik K. Antonsson (California Institute of Technology)
John R. Koza (Stanford University)