Towards Rule-based Knowledge-Based Systems for Synthetic Biology

Ernst Oberortner, Audrey Lewis, Douglas Densmore
Department of Electrical and Computer Engineering
Boston University
{ernstl,alewis13,dougd}@bu.edu

ABSTRACT

The design of novel synthetic biological systems involves various design decisions. Biologists mine knowledge usually from the literature and utilize the acquired knowledge in the decision making process, making it hard to reenact the design rationale of functioning biological systems. Moreover, expert knowledge gets lost. We introduce the applicability of Knowledge-Based Systems (KBS), enabling to electronically share, exchange, and communicate design rationale formulated as rules. The ultimate goal is to computationally support biologists in the decision making process and to understand and learn how and why biological systems function or fail to function.

1. MOTIVATION

The design paradigm of synthetic biological systems proposes to recursively decompose a system under design into encapsulated sub-components that either perform basic biological functions ("parts") or human-engineered functions ("devices") [5]. To responsibly engineer robust, reliable, and secure biological systems, parts and devices must then be composed according to biological "rules". Rules specify conditions, characteristics, and interactions of the parts' and devices' function in isolation and when composed together. We infer that design-specific rules encapsulate knowledge on the functioning of synthetic biological systems.

When iteratively composing parts and devices into complex systems, biologists face various design decisions. Specifically:

- The **Selection** design decision focuses on selecting appropriate parts, devices, or sub-systems based on desired characteristics. To provide some examples: What type must the parts have? Should only "strong" promoters be selected? How do the selected promoters influence the expression rate of the selected coding sequences?
- The **Arrangement** design decision deals with an appropriate ordering of a system's parts, devices, and sub-systems. To provide some examples: Does every gene require an upstream promoter? What orientation must the components have? When and where should desired regulatory interactions occur?

For each design decision it is hard to choose among appropriate alternatives with no or little knowledge on the biological function. Currently, biologists mine the literature in the decision making process or follow trial-and-error and combinatorial design approaches. Design specifications of biological systems and their building-blocks must be enriched with rules, enabling to infer and reason biological knowledge based on existing knowledge. Such approach will computationally support synthetic biologists in the decision making processes of forward engineering novel systems. Also, an automated and standardized knowledge transfer will facilitate learning and understanding of the functioning and language of biology.

2. BACKGROUND ON KBS

Knowledge-Based Systems (KBS) are clearly separated into a Knowledge-Base (KB) and an Inference Engine (IE). Also, a KBS usually provides interfaces for humans and machines, allowing them to interact with the KBS [2].

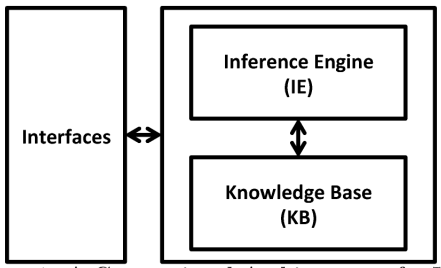

Figure 1: A Conventional Architecture of a KBS

The KB of a rule-based KBS contains various types of knowledge. *Facts* are problem-specific and immutable pieces of information and originate, for example, from experiments, observations, or novel discoveries. Depending on the offered interfaces, facts can either be inserted programmatically or manually, as well as by the IE. *Rules* can encapsulate general knowledge as well as specify constraints about valid relations among the facts. Rules are usually specified in the following form:

IF <CONDITION> THEN <CONCLUSION>

Conclusions are implications of conditions and can utilized to nest rules and to perform appropriate actions if the conditions are satisfied.

Depending on the application of the KBS, the IE performs various tasks. For example, an IE asserts the conditions of the rules on the facts in the KB and performs the rules' conclusions or actions. By specifying appropriate conditions, conclusions, and actions, an IE can infer novel facts or rules automatically, for example, by machine learning algorithms.

A prominent subset KBS are so-called *Expert Systems*, in which the knowledge — either facts or rules — stems from domain experts, such as synthetic biologists. Usually expert systems also provide additional interfaces for the domain experts.

Depending on the requirements and application, a KBS consists of additional components. For example, an "Explanation" component can describe how certain conclusions were deduced or why certain actions were performed.

3. CASE STUDY

Now, we demonstrate the applicability of a KBS on a case study designing a genetic AND gate [1]. Currently, we develop *Sparrow*, a rule-based KBS for synthetic biology offering, at the time of this writing, a primitive set of interfaces that can be accessed programmatically. *Sparrow* is build upon the JBoss Drools rule engine[1], enabling rapid prototyping of synthetic biology designs and their associated rules.

First, we manually specify a library of 13 parts in Eugene [3] and insert them as *facts* into the Sparrow KB. We also specify relationships among four orthogonal repressor-promoter pairs as presented in [7]. For example, the $AmeR$ repressor gene is orthogonal to the $pAmeR$ promoter.

Next, we select appropriate parts from the library. For example, the Sparrow query TYPE=="PROMOTER" returns all promoter parts. According to [7], an AND gate consists of 17 parts arranged in a specific order. We specify appropriate positioning constraints in Eugene. Based on the number of parts in the library, Eugene returns $12,288$ AND gates, which we store as *facts* into the Sparrow KB.

However, most of the $12,288$ AND gates contain common failure modes of genetic circuits [4]. Hence, we engineered knowledge from [4], formulated two common failure modes as rules, and stored them into the Sparrow KB. We have chosen the "Orthogonality" and "Recombination" failure modes since both encapsulate knowledge to support the Selection and Arrangement decision making in designing genetic circuits. We formulate the rules as follows:

$$\begin{gathered}\text{IF } R1 \text{ is not orthogonal to } pR1 \wedge \\ R2 \text{ is not orthogonal to } pR2 \wedge \\ R3 \text{ is not orthogonal to } pR4 \wedge \\ R1 \text{ is not equal to } R2 \wedge \\ R2 \text{ is not equal to } R3 \wedge \\ R3 \text{ is not equal to } R1 \\ \text{THEN The design is valid}\end{gathered}$$

The first three conditions formulate the "Orthogonality" failure mode and the last three conditions specify the "Recombination" failure mode. All conditions are composed using the logical AND ($\wedge$) operator. In this case study, we conclude that an AND gate design is valid if it complies with those five specified conditions. Based on the specified rule, the IE of the Sparrow KBS finds 144 rule-compliant AND gate designs.

As demonstrated, six conditions — engineered from the literature [4] — prune the $12,288$ AND gates to 144 AND gates, also enabling to computationally exchange, communicate, and share the formulated rule.

[1] `https://www.jboss.org/drools/`

4. CONCLUSION

Galdzicki et al. [6] utilize semantic web technologies to develop a KB of standard biological parts, enabling efficient querying of biological parts based on characteristics, such as type, DNA sequence, or orientation.

One drawback of a KBS for synthetic biology lies in the development and maintenance costs, such as the initial insertion of biological knowledge. Furthermore, the probability that a KBS will provide wrong answers because of faulty inferences and reasons will never disappear. A KBS is most powerful when its KB, IE, and interfaces are tailored for a very specific application domain or biological organisms in synthetic biology. Here, we demonstrated how to engineer rules from the literature. However, knowledge must be engineered, represented, and specified in close collaboration with domain experts.

Benefits of a synthetic biology KBS lie in a standardized, automated knowledge transfer of how and why synthetic biological systems work or fail under a specific set of conditions. A KBS can then be further utilized to *explain* how and when synthetic biological systems function and behave as programmed. Explanations can then be further utilized to *learn* common design principles to design and build robust, reliable, and secure synthetic biological systems. Learned design principles can then then be further utilized for *education*. We believe that Knowledge-based Systems are crucial in engineering functioning, reliable, and secure synthetic biological systems, will help to save time and money, and will enable the automate and standardize synthetic biology engineering processes.

Acknowledgements

The work was funded under NSF grant 1147158 and by the Agilent Technologies' Applications and Core Technology University Research (ACT-UR) program.

5. REFERENCES

[1] Anderson, J. C., Voigt, C., and Arkin, A. P. Environmental signal integration by a modular AND gate. *Molecular systems biology 3*, 133 (Jan. 2007), 133.

[2] Beierle, C., and Isberner, G. K. *Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen*. Vieweg, 2006.

[3] Bilitchenko, L., Liu, A., Cheung, S., Weeding, E., Xia, B., Leguia, M., Anderson, J. C., and Densmore, D. Eugene: A domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS ONE 6*, 4 (04 2011), e18882.

[4] Brophy, J. A. N., and Voigt, C. A. Principles of genetic circuit design. *Nature Methods 11*, 5 (Apr. 2014), 508–520.

[5] Endy, D. Foundations for engineering biology. *Nature 438*, 7067 (2005), 449–453.

[6] Galdzicki, M., Rodriguez, C., Chandran, D., Sauro, H. M., and Gennari, J. H. Standard biological parts knowledgebase. *PLoS ONE 6*, 2 (02 2011).

[7] Stanton, B. C., Nielsen, A. A. K., Tamsir, A., Clancy, K., Peterson, T., and Voigt, C. A. Genomic mining of prokaryotic repressors for orthogonal logic gates. *Nat Chem Biol 10*, 2 (Feb. 2014), 99–105.