The Synthetic Biology Open Language a data exchange standard for biological engineering

Michal Galdzicki

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2012

Reading Committee:

John H. Gennari, Chair

Herbert M. Sauro

Daniel L. Cook

Program Authorized to Offer Degree:

Biomedical and Health Informatics

University of Washington

Abstract

The Synthetic Biology Open Language a data exchange standard for biological engineering

Michal Galdzicki

Chair of the Supervisory Committee:
John H. Gennari
Department of Biomedical Informatics and Medical Education

Synthetic biology is the emerging research and engineering field concerned with the design and construction of new biological functions and systems. Synthetic biologists are engineering organisms to solve outstanding problems in medicine, bio-energy, environmental health, and nutrition. Their goal is to improve the biological engineering process by applying standardization, decoupling, and abstraction. To more efficiently engineer gene circuits synthetic biologists need software tools that support standardized data exchange.

For my dissertation research I led the development and deployment of the Synthetic Biology Open Language (SBOL). In this dissertation, I present the SBOL community, the specification, and demonstrations of its use. The SBOL community is supported by stakeholders from the synthetic biology software community. The SBOL Core specifies the vocabulary, data model, and format to define the standard. I describe SBOL Core as a common representation for synthetic biology designs capable of describing theoretical DNA component designs; annotated DNA sequence; and collections of components. To aid the exchange synthetic biological designs among software tools I explain the software libraries which support the implementation of SBOL. Then, I illustrate the recognition of its value and acceptance by the stakeholders through the deployment of the technology at collaborating sites. Finally, I show how the choice of Semantic Web technology to facilitate the information exchange between software can also be used for information retrieval to improve the selection of DNA components in new designs. Through this work I contribute to the development of informatics standards a computational infrastructure to enable a rapid biological engineering process for biotechnology.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my wife Sarah and my friends from the Biomedical and Health Informatics 2007 cohort Alicia, Becky, Blaine, Casey, CJ, and Mabel for their support throughout my entire time in the PhD program.

Thank you to my mentors and collaborators Professors Herbert Sauro, Daniel Cook, and John Gennari, who supported my ideas and encouraged me to pursue this research.

A special thank you to Georg Seelig, who served as the Graduate School Representative on my Supervisory Committee.

I am also indebted to the SBOL Developers for their contribution in the SBOL discussions: Eduardo Abeliuk (Teselagen), Laura Adam (Virginia Bioinformatics Institute), Aaron Adler (BBN Technologies), J. Christopher Anderson (University of California, Berkeley), David A. Ball (Virginia Bioinformatics Institute), Jacob Beal (BBN Technologies), Swapnil Bhatia (Boston University), Michael Bissell (Amyris, Inc.), Matthieu Bultelle (Imperial College London), Yizhi Cai (Johns Hopkins University), Deepak Chandran (University of Washington), Joanna Chen (Lawrence Berkeley National Lab), Kevin Clancy (Life Technologies), Kendall G. Clark (Clark & Parsia, LLC.), Daniel Cook (University of Washington), Wilbert Copeland (University of Washington), Douglas Densmore (Boston University), Omri A. Drory (Genome Compiler, corp.), Drew Endy (BIOFAB and Stanford University), John H Gennari (University of Washington), Raik Gruenberg (IRIC, University of Montreal), Jennifer Hallinan (Newcastle University), Timothy Ham (Joint BioEnergy Institute), Nathan J. Hillson (Lawrence Berkeley National Lab), Cassie Huang (Boston University), Jeffrey D. Johnson (University of Washington), Marc Juul Christoffersen (BIOFAB), Kyung H. Kim (University of Washington), Richard Kitney (Imperial College London), Allan Kuchinsky (Agilent Technologies), Sung Won Lim (Genspace), Matthew W. Lux (Virginia Bioinformatics Institute), Curtis Madsen (University of Utah), Akshay Maheshwari (BIOFAB), Goksel Misirli (Newcastle University), Barry Moore (University of Utah), Chris J. Myers (University of Utah), Josh Natarajan

(Autodesk Research), Ernst Oberortner (Boston University), Carlos Olguin (Autodesk Research), Jean Peccoud (Virginia Bioinformatics Institute), Josh Perfetto (Cofactor Bio, LLC.), Hector Plahar (Joint BioEnergy Institute), Darren Platt (Amyris, Inc.), Matthew Pocock (Newcastle University), Jackie Quinn (Harvard University), Sridhar Ranganathan (Life Technologies), Cesar A. Rodriguez (Genome Compiler, corp.), Nicholas Roehner (University of Utah), Vincent Rouilly (University of Basel), Herbert M. Sauro (University of Washington), Evren Sirin (Clark & Parsia, LLC.), Trevor F. Smith (Agilent Technologies), Lucian P. Smith (University of Washington), Guy-Bart Stan (Imperial College London), Jason Stevens (University of Utah), Vinod Tek (Imperial College London), Alan Villalobos (DNA 2.0, Inc.), Mandy Wilson (Virginia Bioinformatics Institute), Chris Winstead (Utah State University), Anil Wipat (Newcastle University), and Fusun Yaman Sirin (BBN Technologies).

DEDICATION

To my grandparents:

Witold Korkus
Dr. Bożena Korkus

Alicja Gałdzicka
Docent Dr. Zygmunt Gałdzicki, Wrocław University of Technology

Chapter 1. INTRODUCTION

"*How the hell do some rocks become a toaster?*"
The Toaster Project, Thomas Thwaites, 2011

Thwaites, inspired by a quote from Douglas Adams' *Mostly Harmless - The fifth book in the increasingly inaccurately named Hitchhikers Trilogy*, set out to build a cheap toaster from scratch. Through the analysis of the quest he reveals the staggering complexity of knowledge that civilization has created to make the cheap plastic goods we rely on today. His epic journey to mine the ore and find the methods for the building process began with 157 parts and 404 sub-sub-parts of the $6 appliance he bought to start the project [1]. Much like the character in the Adams' book, without the rest of society he fails, repeatedly.

In contrast to the marvels of modern engineering, the biotechnology industry has been stuck. Human insulin produced by recombinant DNA technology was the first commercial health care product derived from biotechnology [2]. Since this landmark triumph the technology has helped scientists better understand how cells work and add new functions to living cells [3]. However, progress in realizing this tremendous potential for health and medicine in the world has been slow. On average, new drugs take 10-15 years and $802 million to develop because the tools available to scientists have not changed significantly since the 1970s [4]. A radically more efficient genetic engineering process can improve this outlook.

What is needed to move biotech forward? The promise made to the general public is that biotechnology will have a revolutionary impact in improving the human condition. Compared to feats of electrical engineering, genetic engineering has not improved in efficiency since human insulin was cloned and produced in bacteria by Genentech. Admittedly, cells are far more complex than toasters, but today, biological engineers are not routinely developing consumer goods using 404 engineered DNA components. The capacity of current technology used to synthesize and assemble DNA molecules [5] [6] far surpasses the current practice of forward engineering. The limiting factor is the understanding science has of living organisms. The current state of biological engineering knowledge is not sufficient to design and build even the simplest cells at will.

In this dissertation, I present the creation and deployment of a data exchange standard for the community of synthetic biology researchers. The demand for data standardization arises from the need to collaborate in order to engineer ever more complex bio-molecular circuits. This need is posited on the experience that once we move from individual examples, with often neatly hidden complexity, the number of components and their relationships grows astronomically large.

1.1 Motivation for Research

Synthetic biology is the emerging engineering field concerned with genetic engineering of biological systems. Synthetic biologists apply the process of specification, design, modeling, testing, and validation to make organisms easier to build. These researchers aim to create organisms which have a range of practical applications [7]. For example, in the last decade, research efforts from synthetic biology laboratories began to show promise in the clinic. Synthetic biology approaches are beginning to make their way to the bedside through the treatment and prevention of infections, vaccine development, microbiome engineering, cancer treatment, cell therapy, and regenerative medicine [8]. Such customized therapies, designed to act to specification, have great potential to improve human health. The inherent properties of biological systems, replication, evolution, and self-repair, are both challenges and opportunities to engineers. Notwithstanding challenges of biological complexity, the potential advantages provide solutions to the world's most pressing problems beyond medicine, such as in energy and the environment. Engineered organisms already produce biofuels from renewable sources and produce biological materials such as spider-silk [7].

However, the early examples of synthetic biology applications do not yet fulfill the promise of modern engineering. Synthetic biologists have only begun to learn how to apply the engineering principles of standardization, decoupling and abstraction [9]. More recently, researchers in the field began developing computational tools that support the application of the engineering principles. These tools aim to implement a formal process of specification, design, modeling and analysis, construction, and experimental testing to make organisms [10]. Endy [9] called for standards that define, describe, and characterize operation of parts and the overall system, as well as legal standards which aid the sharing and re-use of the standardized parts. As he suggests, new technology is needed to allow computer applications to work with previously described data and information. Standards that define the structure and meaning of that data are needed to formalize and enable their practical use. Such standards are necessary to realize the computer-aided design process and bring biological engineering to a modern engineering practice.

Arkin [11] opines that standards for formal sharing of information and materials are critical to increasing efficiency in synthetic biology. To date standardization has been a very active research direction in synthetic biology [12-15]. Standards improve communication, compatibility, interchangeability, reproducibility, effective use, fitness for use, safety, quality assurance, and ultimately consumer and environmental protection [13]. Muller et al [13] provide a broad review of standards efforts.

In this dissertation I consider how we can create standard representations which are useful and sustainable, in the face of the evolution and development of a new discipline of biomedical science. Standard representations provide value to the participants through the derived value of cooperation. This value becomes extraordinary at large scales. The benefit of cooperation is demonstrated by standards in technology, even toasters that we use today. Simply put the sum is greater than its parts. On a social level the cooperation is a synchronization of human behavior. It is cooperation when we can depend on it based on a previously arranged agreement. Just like any synchronization this depends on communication between the participants. For example, Hypertext Transfer Protocol (HTTP) [16], now a very common standard, is useful for data communication over computer networks. The result, a standard, enables those who participate in the agreement to achieve greater efficiency through the coordination of their work.

My proposed solution is based on a call for the need to standardize data in life sciences research [17]. This call was heard within the field of bioinformatics when the research focus moved beyond the one or a few genes to thousands of genes. Bioinformatics has concerned itself with the understanding, analysis, and management of life sciences information for decades [18]. Lessons from success and failures within this related field can serve to inform efforts in synthetic biology. While the goal of synthetic biology is the design and implementation of new biological systems the physical substrate with which we are concerned is the same as the other life sciences. The ability to manipulate and interrogate molecular level components, especially DNA, is what enables synthetic biologists to realize biological circuit designs. Software tools which aid in planning, performing molecular techniques, and interpreting the results require a diversity of up-to-date information. Standardized data from multiple sources and the capability to manipulate those data structures allows for an improvement in the efficiency of research and, in some cases, offers new possibilities. If computational analyses are to generate meaningful results, established conventions for naming and describing biological objects in terms relevant to the goal of the analyses are critical. Additionally, the information provided will need to be provided in a format which can be parsed computationally; its descriptions will have to conform to a structure, and a constrained terminology. Once it is possible to interpret these vast information resources computationally, the novel insights gained can be leveraged to improve new designs. These questions of how to manage an ever increasing body of knowledge about biological systems remain unanswered by either field.

1.1.1 *Information needs throughout the biological engineering research enterprise*

The current challenge for synthetic biology is the diversity of data produced by the many scientists working on different aspects of the field. The data created at each stage of the engineering process is valuable within a single project, across collaborators, and in aggregate form for future ventures. Can we create a web of synthetic biology information that is a collective memory which enables collective learning for future biological engineers? The answer begins with the agreement of the community to use a common representation for minimally useful information, a representation of a core data model. However, the model must be flexible enough so that new information can be added over time. To make progress in the field of synthetic biology, scientists must be able to communicate their research findings. A solution is needed to help them to better understand each other despite their diverse academic backgrounds.

Research in biological design is at the center of synthetic biology. I describe the current state of the work in the field in Chapter 2 in more detail. Today, the DNA sequence level design is the core of the engineering process for the majority of the work in the field. This is especially true for the design of synthetic genetic circuits. To create a synthetic organism the design process must necessarily specify the DNA sequence [19]. I believe this is a necessary element to realizing the vision of synthetic biology. The sequence information itself is not sufficient; the information that describes the sequence is also needed. The representation of the DNA sequence as a design element is therefore essential and at the core of synthetic biology information needs.

Synthetic biologists have begun to apply an engineering cycle approach to build synthetic organisms. At each stage they need information to design, assemble, analyze the test of the system, or to modify the design to better fulfill the design specification. Transitions between stages of the engineering cycle are also the points at which the researchers transfer information between software tools, researchers, and organizations. This information describing the design of the synthetic organism is almost always an abstraction of the DNA of that organism, often restricted to the segment being modified. Therefore, to help the synthetic biologist to transfer the information, an unambiguous language that can capture the information is needed.

To transfer the information from one researcher to another and from one software tool to another, both parties need a shared understanding to effectively communicate. Shared understanding is, "a collective way of organizing and communicating relevant knowledge, as a way of collaborating" [20]. This is critical for people to coordinate work and must be reflected in the technology that supports their work. A data exchange standard, with defined semantics, would fulfill this need [21]. The overall contribution of this dissertation is to create such a standard and to promote its use as a method for building shared understanding. However, developing just the technical capabilities of a solution to satisfy the need to exchange data is not enough. Additionally, the technical solution needs to be accepted by the community in order to succeed.

Buy-in from stakeholders from the synthetic biology community is necessary to develop the standard and for the standard to be used by that community. Stakeholder buy-in will ensure that the standard satisfies their needs. First, the stakeholders will ultimately be the people who benefit from the ability to exchange data. The synthetic biology software developers will directly interact with the standard when implementing the exchange capabilities into their own systems and the synthetic biologists who will use the software tools and information for the purpose of biological engineering. Second, the stakeholders are the people who must be engaged in the decisions to form the standard itself. It is critical to engage the stakeholders early and often in the design of the standard. For a data exchange standard this requires both the software developers and the synthetic biologists to be a part of the group which creates the standard. Therefore, in my research I actively pursued collaborations with the stakeholders. In this pursuit I led a group of stakeholders towards the definition of the standard and helped to create a community interested in synthetic biology data standardization. Below, I expand on the needs of this community represented by synthetic biology software developers and those on the interface between computational and laboratory research.

1.1.2 *Limited by lack of availability of high quality descriptions*

While biological engineers strive to capture the biophysical theory essential for predicting how a newly designed synthetic organism will behave, the current state of this knowledge is far from ideal. To facilitate the research towards this goal, specifically through the application of computational tools, the data required to engineer biological systems should be electronically accessible and interpretable. The challenge to represent this information computationally is complicated by the enormous diversity of data and modeling methods. There is a plethora of biological components, interacting physically and chemically, with implications for behavior at multiple time and spatial scales. These factors contribute to the lack of the kind of high quality quantitative descriptions of behavior needed to predicatively engineer using biological components. Until recently, no dedicated synthetic biology resource offered mathematical models of parts or systems. Resources such as, BioModels DB, Standard Virtual Parts, and SABIO-RK can serve as sources to build such a resource. However, the lack of experimentally derived model parameters has hampered progress so far [22]. A recent effort has taken the Standard Virtual Parts [23] approach and expanded it to the BacilloBricks repository at Newcastle University, a promising resource for such information. Additionally, the BIOFAB has recently made an effort to systematically characterize a large number of professionally designed parts. Efforts such as this will help to produce the necessary descriptions which can then be used to inform design of new biological systems. Going forward, this will be an important area of standardization.

1.1.3 *Current research results are not easily reproducible*

In current synthetic biology practice, to propose a practical design the engineer must know what components are available. Synthetic biology has not advanced far enough yet to enable engineers to design starting with principles. Today, there are simply not enough well characterized components. Therefore, biological engineers rely heavily on the results from published research. The DNA sequence design is at the core of the engineering process. If the field succeeds in reproducing research at the DNA sequence level, future research will be better able to better inform principles of design at higher levels of abstraction.

DNA components used to successfully build a system used in a previous publication are important starting points for new projects. However, it is very difficult to replicate the work published today. Building new systems directly from the information provided is impossible in practice without receiving additional information or even materials from the authors [19]. Repositories, such as the Standard Biological Parts Registry, play an important role; however, it lacks consistent descriptions of the sequence [24]. The standardization efforts to date are not sufficient to allow reliable re-use and therefore engineering. The ability to easily transmit the information describing these components is needed to support the engineering cycle within an individual endeavor and across institutions.

Scientific communication, such as is facilitated by science journals, is also necessary for the overall progress of the field. The transfer of published designs among academic research groups, and hopefully, industry, is necessary to foster translation of synthetic biology research into practice. Such translation requires re-use of components from previous projects, and dedicated characterization efforts. Also, cost of implementing and of transferring published designs should be minimal. The information needs of synthetic biologists span the entire field. My work begins the standardization of data exchange for synthetic biologists, but it is limited in scope to the DNA sequence level.

1.2 SCOPE

The evolution and rapid growth of the synthetic biology field poses a challenge to standardization. I undertook research to develop a data exchange for synthetic biology while the researchers in the field have yet to come to a consensus over many definitions. Synthetic biologists continue active research on some of the most basic questions for engineering biology. In the face of these changes to formulate the standard, the stakeholders had to agree to a common computational representation. These two facets of the goals of synthetic biology are at odds with each other. Therefore, the focus of my work was to leverage the community of participants, to define the technical aspects of data exchange, and to deploy the solution using an iterative strategy sensitive to the changing nature of the field.

Furthermore, my dissertation research relied on the scope of the representation of an abstraction at the DNA sequence level. I discuss the principle of abstraction in more detail in Chapter 2. I acknowledge that exchange of system design, especially dynamic simulation based on mathematical models is critical to complete vision for predicable biological engineering, as described in Section 1.1.2. However, at the core of biological design is the representation of DNA sequence. The representation at the DNA sequence level is a prerequisite for the practical application of predictive simulations to biological engineering. Thus, in my dissertation I began the development of the data exchange solution with the representation of segments of engineered DNA.

Additionally, while research in synthetic biology does take advantage of eukaryotic systems, especially *Saccharomyces*, and has even been extended to mammalian cells, there is a bias towards research using prokaryotic organisms. In this dissertation I limit the discussion of SBOL capabilities to the representation of DNA sequence for prokaryotic genetic elements. As a consequence, I leave out the potential for representation of intron-exon delineation. Furthermore, I do not consider the complexities which result from the difference in the molecular structure of DNA in eukaryotes, such as epigenetic modification, nor chromatid structures.

The semantic representation I discuss is a simple abstraction of the DNA sequence and its implications for forward engineering. Therefore, the implications for data exchange are constrained to the challenges synthetic biologists face in contrast to the reverse engineering perspective which most biological researchers take in deconstructing natural systems. While these limitations pose some risk of excluding information critical in the engineering of new organisms, I believe the SBOL representation is abstract enough to eventually be extended to include these more complex cases.

1.3 CONTRIBUTIONS

Throughout my dissertation research I worked to develop the Synthetic Biology Open Language (SBOL) a data exchange standard for synthetic biology information. SBOL is the result of a collaborative effort of the Synthetic Biology Data Exchange Group to develop standards and technologies that facilitate information exchange for synthetic biologists. In my research I led the effort to define and deploy the SBOL Core data model, a common representation for the description of synthetic biological designs.

The SBOL standard is the rough consensus of core synthetic biology concepts and their relationships and represents the semantics of synthetic biology theory and practice. In Chapter 2 I expand on the discipline of synthetic biology in more detail and build the case for a synthetic biology data exchange standard. In Chapter 3 I describe the SBOL community. The SBOL Developers are a group of stakeholders from synthetic biology. Leaning on the history of experience from the successes of the systems biology and bioinformatics standards communities, it is within this group that that we, the SBOL Developers, established the standard. When working with my collaborators we continuously refined the model and the technology to use it. I describe the technical specification of the SBOL Core data model and its serialization in Chapter 4. I developed the serialization format for the core model so that SBOL core is machine readable and interpretable. To inform and inspire this new standardization, I draw on and apply the lessons of information technology solutions proposed for the Semantic Web and created the technology to use it.

Then with the help of my collaborators I deployed the standard, in several iterations, within synthetic biology software tools and repositories to demonstrate its use. We used an open process for the evolution and standardization of data models. I describe the capabilities, process, and deployment of SBOL in Chapter 5. The process was significantly inspired by the framework for how data models in synthetic biology should be published [25]. Through this work I contributed to the synthetic biology community and I likewise received help from the same community. Together we built the standard and demonstrated its value Below, I enumerate the specific contributions of my research to the synthetic biology and to biomedical informatics.

1.3.1 *Contributions to Synthetic Biology*

This dissertation makes several contributions to the practice and theory of biological engineering. In particular it addresses standardization of information exchange efforts in the field. The work presented in this dissertation is a basis for future work in the development of standards for synthetic biology.

- The first contribution is the development of the Synthetic Biology Open Language (SBOL) Core data model. The model is an abstraction of DNA sequence designs used in forward engineering of biological systems. The model defines the DNA Component as a fundamental design element and how it can be annotated. The core model is discussed in Chapter 4 and its capabilities are detailed in Chapter 5.

- A second contribution is the SBOL serialization format. The SBOL format is available for immediate use by synthetic biologists in compliant software tools. The format is used to store SBOL as a file and transfer it between software applications. The format is described in Chapter 4 and the initial compatible tools are featured in Chapter 5.

- A third contribution is the implementation of software libraries that support the use of SBOL. This involved several cycles of development, feedback, and re-design. In the final stage the software implementation tasks were assumed by other members of the team and expanded to new programing languages. The result is a community supported free and open source set of libraries which support basic serialization and de-serialization of SBOL. The availability of these libraries simplifies the process of adoption for software developers. The software libraries are described in Chapter 5.

- A fourth contribution is the successful demonstration of use by different institutions in several scenarios. These demonstrations show the general practicality and the potential benefit of SBOL to synthetic biology practice. The demonstrations are described in Chapter 5.

- A fifth contribution is the formation of the SBOL Developers group, a community concerned with standardization in software tools for synthetic biology. This community has formed around SBOL, but has established ties which may result in future collaborations leading to additional benefits in computational solutions for synthetic biology. The community uses workshops and an e-mail mailing list to maintain relationships and continues to grow and mature as a social construct resulting from the work to develop the SBOL core model. The community around SBOL is described in Chapter 3.

- A sixth contribution is the formation of a process to standardize future data models for the purpose of biological engineering and design. The social process which led to the SBOL core data model instituted a formal process within the synthetic biology community to establish standards such as these. This involved the use of the BioBrick Foundation's Request For Comments (RFC) model based on the prior achievements of organizations such as the IETF. The success of this process for the synthetic biology domain demonstrates the strength of the community based consensus driven standardization approach for synthetic biology data models. The process and the community which enabled this contribution are described in Chapter 3 and their implications are discussed in Chapter 6.

These contributions have the potential to have a positive impact on reproducibility, sharing, standardization of data models, collaborations in software development, and policy of the broader synthetic biology community. Finally, these contributions hold the promise to lead to the ultimate goal of increasing efficiency of genetic engineering.

1.3.2 *Contributions to Biomedical Informatics*

Someone once asked me, "Is biomedical informatics just a service discipline, or is there informatics that does not play a service role?" This dissertation plays both roles. Through the service to the synthetic biology community I have both aided the biological engineers and furthered informatics research in knowledge representation of biological systems. This work aids the creation of new living systems and provides a methodology for defining and deploying a computational representation of such systems in order to transmit them among researchers. I accomplished this work by creating a knowledge representation framework that enables researchers to interrogate the computationally sorted information to derive new knowledge.

- The first contribution to the practice of informatics is the extension of knowledge representation to account for human-made living machines. The SBOL Core data model is the first formal representation for synthetic biological constructs. The representation provides capabilities for the forward engineering of biological designs in the practice of synthetic biology. The SBOL Core data model is described in Chapter 4.
- A second contribution is enables the effective use of biomedical data for problem solving facilitated by knowledge representation. The new representation is for synthetic biology, a new biomedical field, which is already affecting developments in medicine. The SBOL Core data model, a representation of DNA sequence designs, is described in Chapter 4 and its use in biological design is described in Chapter 5.
- A third contribution is the adoption of the Sequence Ontology as a controlled vocabulary for DNA components. Within SBOL the Sequence Ontology is leveraged to describe the

type of the DNA component instilling a rich but unambiguous terminology across all compliant data sources. Use of the ontology introduces the synthetic biology community to the use of an ontology brining extant informatics standards to biological engineering. The use of the Sequence Ontology is described in Chapter 4.

- A fourth contribution is a new application of the Semantic Web. The representation of the core data model as RDF, a Semantic Web standard, enables it to be used for multiple purposes, such as data exchange, information retrieval, and provides the additional capability of its re-use in unanticipated future scenarios. The use of Semantic Web standards is described in Chapter 4, the application in information retrieval is described in Chapter 5, and the implications of this choice are discussed in Chapter 6.
- A fifth contribution is the application of an iterative development strategy based on community engagement in the design, development, and deployment of a biomedical informatics solution in a specific domain field. The iterative strategy is described in Chapter 3 and Chapter 5.
- A sixth contribution is the leadership role I took in promoting the ideas and the need for a forward looking solution based on biomedical informatics theory. A key aspect of all informatics research is to lead each collaboration towards the application of informatics theory to challenges in the domain field. It would be impossible to do this research in isolation; my collaborators were crucial in the effort. To apply the best practices of the theory in standards development within a community of practitioners, I was required to take the leadership role. It is through the practice of informatics as a service that I was able to gain the trust and respect of my colleagues in synthetic biology. My role in the community is described in Chapter 3.

The contributions of my dissertation are to the synthetic biology and biomedical informatics fields. Through both the service to biological engineering and practice of knowledge representation of biological systems I helped define a new method for data exchange of biological designs. With a strong collaborative community I was able to advance the standardization of information exchange. This standard is a contribution of a foundational technology for biological engineering which could speed the process of translation of new therapies for medicine.

Chapter 2. SYNTHETIC BIOLOGY

"*The work on restriction nucleases not only permits us easily to construct recombinant DNA molecules and to analyze individual genes but also has led us into the new era of "synthetic biology" where not only existing genes are described and analyzed but also new gene arrangements can be constructed and evaluated*".
Waclaw Szybalski and Ann Skalka, 1978 [26]

With these words, congratulating Arber, Nathans and Smith for their discovery of restriction enzymes, Szybalski and Skalka heralded the direct manipulation of genetic material and trumpeted the promise of biological technology. Two decades later the term *synthetic biology* was adopted by biological engineers interested in distinguishing their principled approach to genetic engineering from technical improvements for the manipulation of DNA molecules. Their call to action is, "Make engineering of biology easier and predictable". On the premise that to investigate new designs, models of synthetic gene circuits, can be used as a tool for engineering [27], the new discipline began to develop. The ultimate goal is to apply these lessons to engineer organisms which address challenges in treatment of disease, production of renewable fuels, or improve global food distribution.

To enable the engineering and production of modern consumer appliances, a surprisingly complex product development system is in place [1]. Engineers rely on a multitude of interconnected suppliers and depend heavily on a belief that the components and their subcomponents will assemble and work as specified. Synthetic biology promises to move biotechnology forward into an era where the analogous process of engineering is possible using biology. A new era of synthetic biology will arrive when biological engineers can create new biological technology in the same manner as engineers, who construct consumer goods,

The principled engineering approach [9] to building biological systems, which I describe in Section 2.3.1, has potential. However, an engineering process for biology that allows for efficiencies based on specialization and coordination of work is not yet in place. Ultimately, to bring new synthetic biology products to market efficiently, a process enabling an interconnected commercial infrastructure is needed. First, a research infrastructure to develop the theory and process for synthetic biology will be necessary.

In this chapter I expand on the discipline of synthetic biology in more detail and prompt the technology used to build a synthetic biology data exchange standard. I describe synthetic biology in Section 2.1 and its applications in health and medicine in Section 2.2. Then, in Section 2.3 I describe research in biological design, which is the driving approach to engineering biology. I discuss the engineering principles for synthetic biology in more detail in Section 2.3.1. In Section 2.4 I introduce the computational tools which support the engineering process and could benefit from data exchange capabilities. In Section 2.5 I introduce the ongoing efforts of synthetic biology organizations to facilitate an open framework for information sharing. Finally, in Section 2.6 I introduce the Semantic Web and the open philosophy common to the synthetic biology community and the Linked Open Data (LOD) community. I conclude by discussing the potential of the technology as a solution for data exchange in context of the evolving synthetic biology field.

2.1 Synthetic Biology: Engineering and Research

Synthetic biology is a discipline that connects investigative biological research to the constructive practice of engineering. The goal of synthetic biology research is not only to create more sophisticated systems but to create them using a more efficient process. This new approach to biological engineering could create new capabilities and efficient solutions. Manipulation of genetic material allows synthetic biologists to both investigate and lean on biology for inspiration for solutions. For example, some synthetic biologists create unnatural molecules to reproduce natural biological behaviors, while others use natural biological parts to make systems not found in nature [28]. Therefore, synthetic biologists not only mimic living systems, but also by copying and modifying solutions which exist in nature; they engineer these living systems. As I discussed in Chapter 1 and Section 2.4, these abilities hold remarkable promise to solve outstanding problems in medicine, bio-energy, environmental health, and nutrition. The solutions for medicine may come from the creation of new therapies or from the investigation of biological mechanisms of disease. When realized, synthetic biology can have a tremendously positive impact in improving human health.

Synthetic biologists rely on knowledge from molecular and systems biology to engineer the new biological organisms. These researchers require extensive knowledge about specification, design, modeling, assembly, testing, and validation of the genetic components which form the basis of new ‘living machines’. Substantial progress began a decade ago with the design and creation of two synthetic gene networks, an oscillator [29], and a bi-stable switch [30]. The design and construction in *Escherichia coli* of synthetic gene networks to implement a particular function was inspired by electrical engineering. This stimulated researchers to perfect engineering of genetic parts, devices, and modules and to establish ‘design principles’ which govern such systems and can be applied to build new ones. The goal is to build models which capture the behavior of such systems to enable predictive simulations, which in turn, will be used to support the design of novel systems.

The work by Elowitz, et al. [29] and Gardner, et al. [30] spurred the current focus on building synthetic gene regulatory networks (GRNs) in synthetic biology research. This focus is driven by the technical abilities to manipulate DNA sequence composition to control the production of cellular components. In particular, computational tools which utilize the current understanding of GRNs give the synthetic biologist the ability to keep track of genetic constructs, make design decisions, and model system behavior. Unfortunately, well-organized information about the multitude of biological components used for engineering of GRNs, known as standardized biological parts, is not currently accessible within these software programs. When synthetic biologists use these software tools, such as DNA sequence editors (i.e. A Plasmid Editor (ApE), Laboratory Information Management Systems (i.e. sample management spreadsheets), and CAD software (i.e. Tinker Cell), they locally store knowledge. This knowledge stored for one purpose can be valuable when reused in later project stages and new ventures. However, the computational tools use disparate encodings, and are thus not interoperable. Such tools provide limited access to this knowledge for engineering, slowing the invention of biological solutions to demanding challenges in human health.

2.2 Applications in Health and Medicine

The new field of synthetic biology shows promise in creating solutions to the most significant challenges in human health. Synthetic biologists hope to apply sophisticated biotechnology to problems ranging from public health to medicine. For example, re-engineered whole cells are used as biosensors [31] to detect arsenic in drinking water [32] researchers intend to create inexpensive environmental monitoring technology for resource poor environments. Others, working towards enhancing nutrition were able to increase production efficiency of the antioxidant nutrient lycopene [33]. In a separate effort synthetic biologists engineered a bacteriophage to overcome difficulties in the removal of bacterial biofilm resistant to antimicrobial treatments [34]. Additionally, projects which are still in early stages of research aim to bring solutions to the bedside using new organisms engineered to target tumors [35], generate electricity [36] to potentially power implantable medical devices [37] [38], and to study aging of cells [39].

Most notably, a pioneering effort in synthetic biology to produce artemisinin, a constituent of the anti-malarial artemisinin-based combination therapy, led by Professor Jay Keasling of University of California, Berkeley, is nearing production. To reduce the cost of producing semi-synthetic artemisinin, researchers took the metabolic pathway from *Artemisia annua* (sweet wormwood), from which artemisinin is traditionally extracted, and re-built it in *Saccharomyces cerevisiae* (yeast) to convert inexpensive starting materials into artemisinic acid, the precursor of artemisinin [40]. Production of artemisinic acid in yeast makes it amenable to industrial scale production; thus, hundreds of millions of people threatened by multi-drug-resistant strains of the malaria parasite *Plasmodium falciparum* stand to benefit from the success of this synthetic biology effort.

Today, the majority of synthetic biology systems engineering research focuses on prokaryotes and simple eukaryotes. The most exciting potential for engineering biological systems lies in controlling processes within mammalian cells. Engineering mammalian cells possesses the potential of providing novel strategies in gene- and cell-based therapies [41] [42], as well as tissue engineering [43] [44].

In the next three sections I describe the background material which motivates the creation of a data exchange solution. I establish a perceived gap in the current information technology infrastructure which supports the engineering process in synthetic biology. I find that there exists a strong community philosophy of openness and information exchange as part of the effort to cope with the large complexity and improve efficiency of biological system design. I then provide a foundation for the information technology solution informed by prior work towards the Semantic Web.

2.3 BIOLOGICAL DESIGN RESEARCH

Synthetic Biology research has grown tremendously within recent years with a focus on solutions to practical problems. This focus resulted in well-known examples such as Artemisinin production [40]. However, the re-engineering of the biosynthesis pathway to do so was minimally predictable, at best. Some significant progress towards predictable design continues to be made, notably the design of a robust tunable oscillator by Jeff Hasty's group [45], based on a modeled architecture [46]. However, the design of synthetic transcriptional circuits has yet to reach the predictability of enzyme-free nucleic acid circuit design [47], which permits scalable designs as demonstrated in DNA computing [48]. This ability to engineer biological substrates, especially nucleic acids, suggests that the same possibility for synthetic gene circuits. Attaining this goal requires continued research in gene circuit design, but most importantly it necessitates the development of a rigorous process for engineering biological systems.

Synthetic biologists have begun to apply an engineering cycle approach to build synthetic organisms. Their goal is to rationally engineer complex biological systems using cellular components encoded as DNA sequence. Understanding this process informs the information needs of synthetic biologists. The stages of the cycle are design, assembly, testing, and re-design. Progress of work at each stage is impaired by challenges stemming from biological complexity, limitations of physical processes, and gaps in knowledge. To overcome these challenges there is a cleat need for computational tools that aid the process based on the principled biological engineering theory [49].

2.3.1 *Engineering Principles*

Synthetic biologists draw their inspiration for the engineering process from other engineering disciplines where the principles of *standardization*, *decoupling,* and *abstraction* had a significant impact. The success of scalable and efficient electronic circuit design is an excellent example of the application of these principles. Adoption of the engineering principles leads to a coordination of work through the effective reuse of components and knowledge from previous efforts. Adopting these engineering principles to biological system design should allow these researchers to use previously created solutions and apply them to solve novel challenges. These principles provide both the high-level vision and a practical framework to address the challenges posed by biological complexity and variation during the construction and characterization of synthetic systems [9, 50].

Standardization—the agreement to use uniform specifications, criteria, methods, or processes to allow for interoperability of components—enhances the ability to reuse components and to predict their behavior in combination. Standards also offer the expectation of consistency across information and data of a standardized type.

Decoupling—the separation of complicated concerns into simpler ones according to a hierarchical perspective of a system's organization—allows for problems to be worked on independently and eventually to be combined providing an integrated solution. Decoupling, therefore, provides assurance that distinct information is relevant to a particular context and can be ignored in others, resulting in clearer boundaries for a modular organization of information and systems.

Abstraction—the organization of biological function information across levels of complexity using abstraction hierarchies—hides the complexity of low level components by referencing functional outcome in generalized terms without specifying a mechanism's details. Abstraction thus provides an explicit knowledge framework by which to organize and then perceive the complexity of an engineered biological system design.

These three properties form the theoretical basis of synthetic biology engineering practice; therefore they represent the high level requirements for synthetic biology data exchange. I consider abstraction, decoupling, and standardization as the guiding principles for the design of the Synthetic Biology Open Language (SBOL). To incorporate this theoretical framework into SBOL design I used a community based standardization approach and a Semantic Web architecture to provide the necessary computational methodology. This set of tools is designed for use in and on the Web, a highly-standardized hypertextual publishing system, facilitating robust exchange information across computer networks. Specifically, Semantic Web technologies include declarative, decoupled, and abstract knowledge representation languages and the *standardized* technologies built to model and manage abstraction hierarchies.

Applying abstraction is a powerful tool in overcoming challenges in design due to complexity. Abstraction helps to reduce such problems to essential conceptualizations of relevant facts. Within the paradigm for complexity engineering, there is also the eventual goal to realize the design or to include further details previously hidden.

In the SBOL specification, we consider DNA regions as abstract elements of design for synthetic gene circuits. I provide a concise definition of these DNA components in Chapter 4. These regions often correspond to DNA segments used in synthetic biology for the design synthetic gene circuits. They are often genes and regulatory DNA regions. As a consequence of the technical abilities for manipulating DNA sequence and the limited availability of actionable and reliable higher level knowledge about biological circuits the current state of the art necessitates biological engineering projects to begin at the DNA level. Furthermore, the reduced price and bulk capacity to synthesize *de novo* high-fidelity segments of custom DNA sequence [51] demarcates building and designing at the DNA level. Complex processes such as biosynthesis, signaling, cell division, and cell death are performed by combinations of proteins encoded by genes and are controlled by regulation of transcription or translation. Thus, the common experimental paradigm in synthetic biology is the modification of genetic regulatory networks which resemble electrical circuit connections in terms of network design [52]. The basic elements of these “gene circuits” can then be thought of as modular [53] and are representative of a class of elementary behaviors [54]. It is these biological parts [55] which are commonly used as building blocks for engineering [56]. Such “Gene circuit” designs are paving the road in synthetic biology by providing the fundamental abstractions needed to implement effective and reliable complex system behaviors.

A significant part of the synthetic biology community has adopted a basic building block for the construction of biological systems in living cells, *the biological part*. Shetty, et al. [14] define a biological part to be a natural nucleic acid sequence that encodes a biological function, and a standard biological part to be a biological part that has been conformed to technical standards. The BioBrick [57] standard for physical assembly of biological parts specifies physically interchangeable parts as defined by the composition of their sequence [58]. BioBricks are defined by Assembly Standard 10 [59] as circular vectors of double stranded DNA containing the component regulatory sequence, flanked on the upstream end by EcoRI and XbaI restriction sites, and on the downstream end by SpeI and PstI restriction sites, with no other occurrences of these and other specified restriction sites [59]. The use of a specified cloning protocol with the appropriate endonucleases allows restriction sites to serve as junctions where new biological parts can be inserted, joined, or extracted [60], [14]. This standardized procedure has been widely adopted by the community as it allows for the flexible arrangement of biological parts. In current practice this or the equivalent assembly process is strongly tied to design. The expectation is that in the future a sounder *decoupling* of assembly and design will allow for more flexibility.

The interpretation of genetic regulatory networks as modules composed of parts has strong implications for how the knowledge representation framework should be structured. Furthermore, the rich description of genetic regions provides a key motivation for further standardization of the structure and poses the opportunity to provide use of such information based on the interpretation of the data model structure. Increasing the computational access and therefore broadening the availability of the necessary substrates of this field is crucial in enabling the coordination of work and specialization of expertise.

A rapid biological engineering process for biotechnology solutions proceeding from design, to prediction, to implementation, to testing, and redesign has not yet been fully realized. The establishment of such an engineering practice to effectively control biological processes would have a tremendous benefit toward economically developing medical biotechnology and to even extend synthetic biology to tissue engineering [61]. These advances would allow the use of processes such as self-replication to perform massive parallelization; utilize self-repair to derive robust systems; and co-opt adaptability to increase responsiveness. Biomedical engineers will be able to use these inherent capabilities in living systems if the grand vision of synthetic biology is realized. A significant hindrance in progress of research towards this vision is the availability of reliable information at each stage of the engineering process.

2.4 Computational Tools

To facilitate the process of development, synthetic biologists apply principles of engineering (i.e. standardization, abstraction, and decoupling) to specify the design, assembly, and validation of new biological systems [9]. In other engineering fields, such as mechanical, electrical, and computer engineering, these principles have led to the highly successful methods used today to build robust and complex products. The multiple scales, diversity, and dynamics inherent to biological systems and materials necessitate the use of computational methods to help manage this complexity. Synthetic biologists need software tools that support the engineering process of biological systems [62].

When working towards a goal of limited complexity design, assembly planning, and construction may be achievable by a ‘manual’ process. Today, it is likely that most of that ‘manual’ process will be performed by using some computer support. If synthetic biologists are to move beyond designs limited to 5-20 genes [44], significant computational assistance is necessary. To allow computer applications to work with previously described data and information, there need to be standards which define the structure and meaning of that data.

Furthermore, with each endeavor, the synthetic biology research community gains valuable knowledge, often useful in repeating the success in new domains. Researchers learn lessons from their experiences apply them to new projects and develop best practices reflected in a continually growing understanding of biological systems engineering. Therefore, accumulating a *store of knowledge* to inform future endeavors is an integral part of synthetic biology practice. To reap the benefits of such efforts, specifically to make building future living machines more efficient; new knowledge should be accessible at each stage of the engineering process. To create access to such knowledge, synthetic biologists need a data exchange solution which not only enables exchange between similar stages of the work, but also helps transition between stages of the engineering cycle.

Computer-aided design (CAD) tools have revolutionized design for print media and design in product manufacturing, especially the design of integrated electronic circuits by allowing the planning of a product based on abstract virtual objects. Synthetic biologists’ core focus is the DNA sequence which encodes biochemical systems of interest. The tools they use are molecular techniques which manipulate the DNA sequence and mathematical models which predict their behavior. Both require software which reads and then helps the researcher interpret the sequence or model. The challenge is to facilitate the work process of engineering biological circuits in a unified computational framework without limiting the ability of these researchers to apply the latest tools available.

In the electrical engineering field such approaches have led to wild success, far exceeding the efforts in the life sciences to date. For example, VLSI CAD applications use object models to distinguish between design objects with a common interface but different implementations. One example, of a standard in electrical engineering is the Electronic Design Interchange Format

(EDIF) [63]. Motivated by prior success in both life sciences and electrical engineering, synthetic biologists are attempting to create the analogous infrastructure for engineering biological systems.

Computational tools, which aim to support the biological engineering at each stage of the cycle, are in development. These design tools require computational access to a library of parts, specifically the ability to query such a library. For example, TinkerCell is a visual planning and analysis tool for synthetic biological networks [64]. TinkerCell supports *abstraction* of biological components as virtual objects and can be used to simulate the dynamics of a designed system to predict its potential behavior. Once a system design with desired behavior is created, matching the design to a set of known biological components poses a challenge. Currently this process of finding reasonable components is a manual process restricted by the researchers' own knowledge of the existence and availability of putative components. The first goal of SBOL is to provide this information in a standardized and computationally accessible form. Software such as TinkerCell could retrieve components matching the design criteria will improve the ability of refining such designs to create realistic quality synthetic biology system designs. In Chapter 5 I describe WikiDust, a plugin query interface for TinkerCell. Additionally, in Chapter 5, I describe a demonstration during which Nicholas Roehner and Prof. Chris Myers imported SBOL DNA components into the iBiosim design tool and exported a completed design.

The next step, the assembly planning stage, realizes the abstract design in terms of available real parts. This stage is focused on the interpretation of physical compatibility standards to plan the construction of a proposed device. To aid in this planning process some guidelines and utilities have been developed [65]; however, the main computational tool support at this stage is provided by software designed for DNA sequence editing, such as A Plasmid Editor (ApE) [66] and Vector NTI [67]. While these tools are capable of reading and writing the GenBank sequence *format [68],* the *de facto* nucleotide sequence standard, synthetic biology DNA constructs are not often submitted to GenBank® [19], the NIH genetic sequence database [69]. Tools, such as Clotho [70], provide an infrastructure for exchanging the information about synthetic genetic constructs which takes into account the assembly standards and their implications for assembly planning [71]. In Chapter 5 describe a demonstration involving the Clotho framework reading, validating, and exporting an example design using SBOL. Additionally, another collaborator Dr. Nathan Hillson recently implemented SBOL import and export capabilities in j5, the Joint BioEnergy Institute's DNA assembly automation software tool [72].

Finally, the design plan is executed using the assembly plan for construction, followed by transformation into or integration with the host cell strain and by the validation of the desired behavior. This process of design through testing is iterated until the outcome meets the specifications of the design. Throughout the repetition of the engineering cycle, a laboratory information management system (LIMS), aids in keeping track of the genetic constructs, strains, and validation results. In practice this function is performed by a locally maintained spreadsheet, such as Microsoft Excel™ [73]. Research to develop a computational tool set to capture and re-use of knowledge from the iterative design, assembly, and experimental validation is the topic of my ongoing collaborative research led by Prof. John Gennari with Clark & Parsia, LLC.

New computational tools, based on the *common information infrastructure* that SBOL provides will be better able to link the information used at the three stages of the engineering process. These tools will provide computational access to information about standardized, decoupled, and abstract biological parts. The goal of supporting the application of the engineering principles is to greatly accelerate and reduce the cost of the engineering process. Furthermore, if these benefits are further substantiated and then become common practice throughout the broader synthetic biology community, the advantage for biotechnology and pharmaceutical research will be enormously valuable.

2.5 Framework for Information Sharing

Strengthening the infrastructure for data sharing alerts scientists to new areas of research, maintains the integrity of science by exposing claims of truth to the potential of validation, and exposes the next generation of scientists to current ideas [74]. The openness of information sharing in synthetic biology is one of the cornerstones driving innovation. For example, in the judging criteria of the International Genetically Engineered Machines (iGEM) competition, the annual undergraduate student event known for a high level of innovation, *sharing* the work product with the community is valued highly [75]. To foster long term growth of the discipline, it is desirable to create the infrastructure for information exchange capable of both serving the local endeavor and broader community so that it can be exploited with ease by many different competing interests and purposes. And, as it turns out, Semantic Web technology, the most appropriate information technology area for information sharing, is also proving useful for information analysis and integration as well. Web based publishing and peer-to-peer exchange can be both supported by Semantic Web technology. Therefore, this technology can enable sharing of semantically rich data, standardized by a formal data model. As well as provide additional benefits such as information retrieval, validation, and to provide the technical infrastructure to support the development of model extensions.

Adoption of technical standards serves as an incentive for participating in a community and can encourage sharing. Conforming to a community standard allows integration of others' work into your own (and *vice versa*), as well as simplifies contributing back by reducing the burden of useful description to attesting to compliance with the standard.

For example, the of the success BioBrick™ assembly standard is especially visible in the context of the International Genetically Engineered Machines (iGEM) competition (igem.org) [76], as evidenced by the growing number of biological parts in the Registry of Standard Biological Parts. The Registry provides services to store and distribute plasmid DNA that conforms to the BioBrick™ assembly specifications and provides some descriptive information, i.e., a physical store and distribution point for biological parts. The Registry website is a publicly available source of information about 18,400 BioBrick™ parts [77]. The website is partially designed as a wiki, and therefore Registry users can edit its content directly. Even though, registry staff members also curate this resource, the voluntarily added information varies between entries. As open science resource the Registry is a unique and rich resource for the synthetic biology community. However, as it is a portal intended for human users, it has limited information retrieval capabilities of parts for a new design. In Chapter 5 I describe the Registry to SBOL convert, and new information retrieval capabilities I developed using SBOL data from the Registry.

In addition to the Registry of Standard Biological Parts there are new notable efforts addressing the need to manage information about biological parts. The Joint Bio Energy Institute Inventory for Composable Elements (JBEI-ICE) registry provides a web based platform as well as a graphical sequence annotator [78]. The free and open source software and this new public repository hosted at JBEI support the sharing of synthetic biological parts. Recently, the Dr. Timothy Ham added support for SBOL import and export in to the platform.

Furthermore, there are also efforts to store quantitative information that describe synthetic biological parts. The Repository of Standard Virtual Parts [23] provides model modules which can be found in the BacilloBricks repository. Additionally, the BIOFAB (biofab.org), a facility for the fabrication and functional characterization of standard biological parts on a large scale [12, 79]. Whilst immensely valuable in the pursuit of predicable biological design, it will in the near future generate immense amounts of quantitative information as part of its effort. A standard electronic form of such information would allow synthetic biologists to effectively exploit the data within computational tools. These systems, just like the design tools I mentioned earlier, will benefit greatly from a standardized information sharing framework. To make this data available to synthetic biologists there is now a need to standardize the electronic form of the knowledge about biological components.

2.6 SEMANTIC WEB

The Semantic Web is a vision and a research program that created new technology and best practices for information sharing on the web. It is inspired by and takes advantage of Artificial Intelligence theory, specifically, knowledge representation (KR) research. Berners-Lee [80] defines it as, "an extension of the current web in which information is given well defined meaning better enabling computers and people to work in collaboration." Otherwise known as the Web of data, it builds on the document Web by providing the tools necessary to make data, the kind found in databases, available for use on the web. On the document Web machines display content, on the Semantic Web they are able to interpret and make use of the data for specific purposes.

KR is a formal approach to represent knowledge in symbolic form, for example, to enable logical inferencing. The formal underpinnings of the Semantic Web are based on the theories developed in this research. For example, a semantic network, where the relations between two concepts carry meaning, and therefore state facts which can be interpreted as true in reasoning, is a form of knowledge representation. This methodology of assigning meaning to links thereby representing the semantics of the relationship between two concepts is central to formally representing data on the Web. The technologies developed by KR researchers were direct antecedents to the XML based technologies used for the Semantic Web.

2.6.1 *Semantic Web Standards*

The Semantic Web relies on information standards and tools to enable people to create data stores on the Web, build vocabularies, and write rules for handling data [81]. The Resource Description Framework (RDF), Web Ontology Language (OWL), and SPARQL Protocol and RDF Query Language (SPARQL) are standards used to represent information in a common form so that it can be interpreted, retrieved, and re-used on the Web. These standards were developed by the members of the World Wide Web Consortium (W3C), an organization which helps to coordinate development of open standards and to maintain the centralized documentation resources.

RDF is a standard model for data interchange. To enable data exchange, RDF builds on the familiar Web URI standard to extend URI Web links to uniquely identify things (i.e resources, such as data) and relationships between those things. Its resource - link to - resource structure is referred to as a *triple*. Taken as a set, RDF *triples* can be represented as a directed, labeled graph, where the resources are the nodes in the graph and links are the edges. This graph form allows the data to be visualized and interpreted at the semantic level, for example, to follow the links, query the data, or to merge it with another RDF data set. Data represented in RDF can then be exposed, written out and published on the Web in a standard format RDF/XML, a standardized syntax for the serialization of the RDF model in XML. RDF compliant software can then read-in the information and therefore enable it to be shared across different applications. Furthermore, since the standard RDF model is shared across the two applications, the data can be interpreted by the application without the need for a distinct format for each application. However, for two applications to interpret the same meaning of a data element, they have to leverage link names (i.e. typed links) to interpret the meaning of the data. This meaning can be represented using OWL.

OWL is a knowledge representation language to specify an ontology, a formal representation of a domain, in terms of its concepts and their relationships. An ontology specifies the semantics, the meaning of the concepts, such as the vocabulary used in a specific domain. The language is designed to represent rich and complex knowledge about the concepts in the domain. It allows the ontology to specify logical statements about the data so that automated reasoning programs can interpret them. For example, such programs can verify the logical consistency of that knowledge or infer new knowledge not explicitly specified in the ontology.

Ontologies written in OWL provide the shared definitions of the domain concepts across applications.

SPARQL is a query language for RDF databases. It enables retrieval and manipulation of the data stored in such a database. For example, an application can specify a request for information to retrieve written in SPARQL and the database will return the information which matches the request. Such an application can provide access to specific information stored in a large collection of data elements. Software implementations of SPARQL use SPARQL Protocol to provide ways to convey queries to query processing service and return results. The HTTP requests are often used to make such query requests. Since, SPARQL can be used to specify that the results be returned in RDF format, the language can be used to transform one RDF representation into another. The three standards, RDF, OWL, and SPARQL are the basis of the Semantic Web architecture. The W3C maintains the documentation about additional Semantic Web standards which serve other needs and are compatible with RDF [82]. Software tools, such as Protégé [83], RDFlib [84] and Sesame [85], implement these standards are, are available for free, and provide the technical foundation needed to apply Semantic Web solutions to data exchange challenges.

The Semantic Web ultimately benefits people by enabling data exchange on a broad scale through the use of computer networks and the software tools which support its standards. It is an extension of the existing Web; it uses the same HTTP and URI infrastructure to define standards for data. Ontologies, written in OWL, provide well defined shared meaning across applications and RDF provides the common data model. A standard serialization of the data to RDF/XML allows any compliant software to read it and take advantage of the ontology to interpret it. Such data can then be used for automation, integration, and re-use of data across different applications. Furthermore, the standards facilitate the data to be processed, transformed, assembled, and acted on in useful ways.

2.6.2 *Linked Open Data*

In 2006, a Linked Data community began to form with the goal of promoting the application of Semantic Web standards and enabling the benefits of the technology more widely. Linked Data is the idea that data should be published in a structured format and it should be interlinked so as to become more useful. Berners-Lee [86] proposed four principles for Linked Data. The principles are a set of best practices for publishing and connecting structured data to create a Web of Data using Semantic Web technology. He proposed:

1. Use URIs as names of things.
2. Use HTTP URIs so that people can look-up those names.
3. When someone looks-up a URI provide useful information using standards (RDF, SPARQL).
4. Include links to other URIs, so that they can discover more things. [86]

A community project formed around the idea of Linked Data, which embraced the additional requirement of openness. The W3C Semantic Web Education and Outreach (SWEO) Linking Open Data (LOD) community project set out to extend the Web with a data commons by publishing various open data sets as RDF on the Web [LOD [87]]. This project embraced the Linked Data principles as a five star rating scheme which indicates compliance with Linked Open Data. The scheme, listed in order of increasing stars, is:

1. Available on the Web
2. Available as machine readable structured data
3. In a non-proprietary format
4. Open standard from the W3C RDF and SPARQL
5. Link to other people's data.

The LOD community continues to publish and interlink data from different sources using RDF. There are now 295 data sets built from a total 31 billion RDF triples, which are interlinked by around 504 million RDF links (September 2011) [87]. This data is freely available to anyone to use and republish. The goal of this community is similar to other "Open" movements such as open source, and open access. Many of the LOD datasets are biomedical information sources valuable for research in healthcare and life sciences.

2.6.3 *Semantic Web in Action*

The Semantic Web is decentralized, there is no objective metric of its impact, but anecdotal reports indicate that the standards and technologies are beginning to be used broadly. For example, companies such as British Telecom, Boeing, Chevron, MITRE, and BestBuy have adopted the technologies for some of their business practices [88]. In the case of Best Buy, the company deployed RDFa in its webpages and used the GoodRelations ontology to publish details about their products on their store blogs and later their product pages [89]. While Semantic Web technologies have been used to improve business to business and consumer applications the most compelling uses have shown promise for biomedical research. For example, Gudivada, et al., [90] used RDF based integration of knowledge to mine and retrieve disease candidate genes for cardiovascular system diseases. Other uses of Semantic Web technology have been applied to integration [91], reasoning [92], discovery [93], search [94] and composition of services [95] and tasks [96].

In a notable example of LOD use for biomedical research, the Linked Open Drug Data (LODD) taskforce [97] linked publically available data about drugs. They integrated clinical trials data (LinkedCT), 5,000 FDA approved drugs (DrugBank), marketed drugs (DailyMed), recorded adverse reactions of drugs (SIDER), disorders and genes from OMIM (Diseaseome), and a database of Traditional Chinese Medicine herbs and genes (TCMGeneDIT) into a combined resource [97]. This resource can be used to find clinical trial information for an herb, the active ingredients, in a pair of drug and herb side effects [98].

Semantic Web technology has significant implications for SBOL, which I define in Chapter 4, as they provide benefits in its development, maintenance, and future uses. Compliance with Semantic Web standards enables SBOL data to be read, manipulated, and interpreted using generic tools such designed to work with RDF and OWL, not just SBOL. These tools are utilized for management of SBOL model structure, creating a scheme for unique identification of elements, and to seamlessly reference third party ontologies, such as the Sequence Ontology [99]. This is similar to the approach taken in SBML to annotating biochemical species and reactions (see Section 3.1.3 in the next chapter), except in SBOL it will be part of the native format. The benefits of using the tools extend to the capability to perform operations such as inference to check consistency, classify the structure, and infer data types. Adoption of RDF in SBOL to encode DNA design data lets computational tool developers to use

SBOL as a format to exchange data and the data can be reused by the Linked Open Data community. My choice of this W3C recommended technology was predicated on the hypothesis that formally modeling knowledge in a computable, standardized, and community supported format will provide long term benefits for the synthetic biology community. In particular, the knowledge modeling capabilities of semantic web technology provide greater abstraction-decoupling-standardization of information components.

2.7 SUMMARY

The computational methods designed as part of this project will remove the technical obstacles to information access and exchange needed for engineering ever more sophisticated synthetic biology solutions. The field of synthetic biology and its practice of principled engineering will benefit from the adoption of Semantic Web information exchange technology, standards, and best practices. The application of this work will advance the potential for innovation in synthetic biology by accelerating the pace of the engineering process.

Chapter 3. TO SHARE DATA IT TAKES A COMMUNITY

A standard is not a standard unless it is adopted by the community. If a community of users is not already sending data back and forth, why should we develop a standard method to make it easier? But, without a standard method to send data back and forth, the effort to do so is a significant impediment. If both the standard's developers and the user community are waiting for each other to move first, at what point will the community become frustrated enough to move towards action? While waiting for such a moment in synthetic biology, how much research effort must be wasted by scientists working in isolation? To avoid this pitfall, I believe that for a standardized information exchange to become reality, a community must form around the vision for data exchange. Successful standards development projects have always actively engaged the community which will ultimately benefit.

In this chapter I discuss the critical role of the Synthetic Biology Data Exchange (SynBioDEX) Group in the development of the Synthetic Biology Open Language (SBOL), a data exchange standard for synthetic biology information. I am a founding member of the group; therefore in this chapter I present our work in building the SBOL community. The technical details of the SBOL standard are presented in Chapter 4. I was inspired by the success of prior efforts, such as MIAME, PDB, and SBML to work with the members of the SynBioDEX group to create a standardized technology framework for synthetic biology. The overall goal of this group is to facilitate communication in the synthetic biology domain by enabling electronic exchange of information.

The SBOL project has emerged to address this goal. Our solution is to create a language to unambiguously describe data using a well-defined, but extensible scheme. We pursued the development of the data model and a standard exchange format as a collaborative group. The data model, SBOL:Core, serves as an organizing structure for information and the format as a standard serialization. The serialization of the data model uses XML, which is compatible with RDF/XML from the Linked Data community described in Chapter 2. It is a standard built on top of standard technology developed by a larger community. The development process and the success of adoption are further described in detail in Chapter 5. As part of my dissertation research, I accepted the responsibilities and privilege of a leader in this group. My primary role in the project was to lead the efforts to define the data model and serialization, and to grow buy-in from the community. To elaborate on the essential nature of community in standards development I discuss this social context of SBOL development.

Throughout the SBOL project, the SynBioDEX group faced social challenges in attempting to meet the needs of both individual scientists and the broader community while creating a useful information technology infrastructure. However, in the field of bioinformatics, significant progress has been made in understanding how to drive adoption of information standards. In this chapter, I draw on the successes of prior grassroots and institutional efforts to adopt standards. Their successes offer hope in changing the research culture of synthetic biology to embrace data standards. Each of these successful standards efforts made a case for its benefits to the community, first. Such a case drives adoption as it makes the usefulness of the solution clear to the potential user [100]. The explanation of capabilities, which I describe in Chapter 5, should help to reinforce the desire for adoption. For synthetic biology in particular, the use cases for software tool interoperability will help to convince the early majority of its usefulness. Alone, the perceived benefits of these capabilities are not enough; they must also be proportional to the required investment to adopt the standard.

Standards developers must create an environment in which new community members feel comfortable beginning to adopt the standard [100]. To further the development of the standard, input must be actively solicited from new members. Keeping use and participation entry costs low is the responsibility of the standards developers throughout the lifetime of the endeavor. The cost to implement compliance with the standard in a software tool or service will be the barrier to entry for most. Strategies which minimize this cost are necessary to succeed. For example, to moderate the implementation costs of SBOL, we developed software libraries which perform serialization and de-serialization (see Chapter 5), documentation which guides implementation (see Chapter 4), and support to answer questions.

Buy in from key stakeholders is critical [101] [102]. In the synthetic biology research community, opinion leaders are found in the well-recognized research labs and companies [103]. In particular, laboratories and companies which already distribute descriptions of DNA components would play an important role in promoting SBOL. If some of these opinion leaders to switch to SBOL, the rest will follow. However, within each local group, a champion of the technology must convince the senior decision makers of the value. As I describe in Chapter 5, three data publishers and three software tool makers have already implemented SBOL in their offerings. Several more are on their way.

Additionally, the expectation that adoption will occur over a significant length of time can help plan the resources needed for the project. Successful adoption is likely to happen incrementally, and requirements need to be set in accordance with this expectation. The complexity of base-line requirements for compliant implementation should be kept minimal. Keeping the cost of basic adoption to a minimum should not only improve ease-of-use by the software developers. Most importantly, benefits should be presented as grounded in the methodology currently in use by the synthetic biology community.

The focus of this chapter is the community, the individuals and organizations which have been the driving force of the SBOL development effort. Adoption of the proposed data exchange solution is paramount to its status as a standard. Only through the *de facto* cooperation of the members can it succeed. In this work, the members of the SBOL community have become my close collaborators. Their initiatives, like the Bacillo Bricks Registry and the BIOFAB Data Access Web Service described in Chapter 5, have helped push SBOL forward. With help from Prof. Herbert Sauro in making the first introduction, I pursued collaborations with software developers working on tools for synthetic biologists. The strategy for my dissertation research was based on the lessons of prior successful biomedical data standards (Section 3.1) and bolstered by the philosophy of openness and desire to share in the synthetic biology community. My new collaborators and I formed the Synthetic Biology Data Exchange (SynbioDEX) and the SBOL Developers groups (Section 3.4). The outcome of the collaborative work is the Synthetic Biology Open Language (SBOL) and a supportive community. We fostered community growth by means of workshop meetings, mailing lists, personal communication, and free online tools (Section 3.3). We enthusiastically presented the ongoing research at conferences and published our position and results in journal publications (Section 3.3.6). As the group matured, we structured the group into a formal organization (Section 3.4) and explicitly defined our values (Section 3.5) I begin the description of the community building process by recounting the successes of prior standards efforts in the next section. The community I describe in this chapter greatly contributed to the specification of SBOL (Chapter 4), and to the demonstration of its value (Chapter 5).

3.1 EARLY EXAMPLES OF SUCCESS

SBOL is not entirely unique: standards have been developed in many areas of biomedical research. In my work, I have drawn significantly on the history of their successful development. Therefore, I start by describing three examples, each demonstrating different approaches used to build communities supporting a standard.

3.1.1 *Protein Data Bank (PDB)*

The Protein Data Bank (PDB) is the oldest electronic repository of biological data. It contains standardized computational representation of structures of macromolecules, such as proteins and nucleic acids. These 3D structures are obtained by methods such as X-ray crystallography or NMR spectroscopy. Not only are these molecular structures an important source of knowledge to use in engineering novel proteins and interaction, but its *pourquoi story* provides synthetic biologists with a history of a successful standardized data model.

The PDB began as a grassroots effort around 1971, and since then it has grown tremendously, as can be illustrated by the number of structures archived. In the beginning, a dozen structures, and now more than 68,000 entries can be found in the PDB [104]. This has made it the authoritative source for structural biology information. This success can be attributed to the responsiveness of the PDB to the evolution of the field, technology and attitudes about data sharing [105]. Throughout the 1970s the PDB founders focused on personal communication with the community. For example, they wrote letters to the authors of articles, inviting them to submit reported structures to the collection. The personal approach engaged early adopters. The strategy worked to get the project started by creating a community around the information resource. Following this model can greatly aid in starting data sharing projects, which depend on community participation.

Driven by the increased appreciation of the value of structural biology, advances in the methods rapidly sped up the pace of structure determination in the 1980's. The growth of the field, as a definitive source of knowledge for the molecular basis of biology and medicine, created the impetus for establishing a policy that would require data deposition into the PDB. By 1989, a formal recommendation specifying requirements for data deposition was published (International Union of Crystallography, 1989). Such policies are premised on the future value of disclosing the detailed structures of macro-molecules. These structures provide tremendous value for downstream researchers. Software can read each structure because their representation is standardized. Recognition of this value was echoed by major journals by requiring PDB submissions concurrently with manuscripts. Furthermore, the National Institute for General Medical Sciences made research funding dependent on such open sharing of data. To support such sharing of structural data and management PDB researchers developed an information infrastructure and new data representation methods. The PDB now coordinates international efforts to integrate, or link, PDB information to related information sources, for example GenBank [106], UniProt (Apweiler et al., 2004), etc. The synthetic biology standards community should critically examine the history of the PDB, gaining the most useful aspects of its history, especially regarding its origins and early decisions made by opinion leaders. The history of prior standards is illustrative of the challenges and solutions which lead to successful adoption and sustained value.

3.1.2 *Data Requirement Success - Microarrays*

Another success story that inspires synthetic biologists is MIAME. This bioinformatics effort organized by the MGED society has become a standard known as the Minimum Information About a Micro Array Experiment (MIAME) [107]. It is a checklist of variables that should be included in every microarray publication [107]. The requirement of this checklist was spearheaded by the international repositories of microarray data Gene Expression Omnibus (GEO) [108] [109], ArrayExpress [110], and CIBEX [111]. However, it was the support of editorial policies at major journals [112] that helped to establish the incentive needed for high adoption across the field. Throughout the last decade MIAME has found broad support. Journal editors now require compliance with its standard for publication. The standard begets uniform information which then allows both meta-analysis and interpretation by any software designed to read such a format. Today, the majority of microarray software is capable of reading and writing such standardized data files. The outcome of the MIAME effort is that results of gene expression studies are now easily accessible for downstream analysis via the web. Synthetic biologists who hope to design ever more sophisticated biological systems can draw upon this example to inform the process of standardization of experimental data exchange.

3.1.3 *SBML - a Standard for Models*

SBML is an important example of a successful standard for the exchange of dynamic models of cellular systems. The Systems Biology Markup Language (SBML) (Hucka 2003) is a community developed and supported standard. It is also directly useful for synthetic biologists, as analysis of models can help select designs with desired behavior or better robustness to perturbations. SBML is used to exchange models between software applications which represent how biological components change over time and relation to one another. The success of SBML can be attributed to its focus on simplicity of representation, software library support, and community buy-in. The experience of developing SBML serves as direct inspiration to SBOL developers. Importantly, Professor Herbert Sauro, founder of SBML, conceived the idea of a need for data exchange standards in synthetic biology. Furthermore, the shared inspiration has brought other experienced members of SBML community, such as Professor Chris Myers, to the SynBioDEX group (Section 3.3.3). Therefore, in the development of SBOL we have aimed to follow a similar strategy.

Researchers who model biological dynamic behaviors choose their strategy for each particular challenge they face. They choose among the different formalisms and computational methodologies to simulate a diverse range of mathematical models of biological systems. For example, some of the mathematical techniques used are ordinary differential equations (ODEs), deterministic hybrid models, differential-algebraic equations (DAEs), partial differential equations (PDEs), and stochastic modeling (Sauro 2006). While these are common types of models to represent synthetic *gene circuits*, alternative formalisms can also be used, such as directed graphs, bayesian networks, boolean networks, and rule-based formalisms (deJong 2002). Among this great diversity of computational methods, quantitative models based on ordinary differential equations (ODEs) are the commonly used form (Sauro 2006). SBML was designed to facilitate the exchange of such models using an XML-based format.

SBML's simplicity relies on the definition of fundamental concepts for dynamic biochemical models. At its core it defines the *Species*, a chemical or other participant of a reaction and *Reaction*, a statement describing change to the quantity of species. Reactions link the product and reactant species with their kinetic laws. Other fundamental concepts such as *Compartment, Parameters*, *Unit definitions*, and *Rules* are also included (Hucka 2003). This model structure allows for a relatively comprehensive representation of biochemical systems and it is consistent with the well-established biochemistry perspective that chemical reactions have reactants and products. One of the strengths of SBML is this simplicity. These basic concepts are easily understood and well known by anyone who has taken a biochemistry course.

An alternative standard for computational models is the CellML language (Lloyd 2004). CellML, on the other hand, represents cellular models using a mathematical description, more closely following the structure of the mathematical equations of the model. This view is capable of representing almost arbitrary mathematical models, providing greater generalizability, at the cost of complexity of the representation.

The richness of SBML is found in the extensible *Annotation* element, which can be used to define the type of a *Species*, *Reactions*, etc. Both CellML and SBML use standard XML based metadata (using RDF) as described by the MIRIAM requirements (Le Novere 2005). The metadata describes the types using terms from standard information resources and ontologies. For example, KEGG compounds and CHEBI terms are used to describe chemical entities, the Gene Ontology or other sources provided by MIRIAM resources are used to identify enzymes which catalyze each reaction.

One of the barriers to adoption of SBML is the ease of encoding models into the standard format created in the many specialized software applications for quantitative modeling. To enable the use of SBML, its authors developed a software library, libSBML (Bornstein 2008). It could be used within existing software to translate the software's internal representation into SBML. The source code of libSBML was made freely and openly available. The result has been overwhelming success, as indicated by its adoption in more than 180 software systems, such as simulators, model editors, and databases [113]. SBML is far from perfect, as not all kinetic model formalisms are supported and individual software projects create models with varied quality. However, the syntactic standardization, enforced by libSBML, produces a base line level of interoperability "good enough" to have gained the considerable buy-in from an active community of researchers. Additionally, the success of SBML can be attributed to the initial effort of a small number of collaborators who adopted the open-innovation model and encouraged community participation. Through support from an international community of interested researchers and participants, it has grown into the *de facto* standard format in its field.

Ongoing development is helping to expand the utility of SBML. For example, a new software library, libAnnotationSBML, links SBML ontology annotations to the web services that describe these ontological terms (Swainston 2009). The growth of capabilities in creating models and the ability to unambiguously annotate the concepts through a curatorial process led to the creation of the BioModels database (Le Novere 2006) (Li 2010). BioModelsDB now holds 420 models, validated by a professional team of curators, and 433 additional models not verified by human inspection. The software applications for simulation of quantitative models provide ready to use tools or at least an advanced starting point for the development of new tools that purposely serve the design-build-test engineering process for synthetic biologists. The library of the models in BioModelsDB includes many genetic regulatory, metabolic, and signaling pathways, thoroughly described, and ready for download into SBML compatible tools, to serve as biological inspiration for new designs.

In the development of SBOL, we have applied the successful strategies of the SBML community. First, we were persistent in the development and maintenance of a community, which can support the standard. We have intentionally kept SBOL simple. SBOL Core, described in Chapter 4, focuses on the most commonly used abstraction in synthetic biology, the 'standardized biological part'. SBOL is immediately useful to synthetic biologists as it can be used to exchange DNA components among software application and used in retrieving components from repositories. To aid software developers we have created software libraries which can be used to adopt SBOL within existing software applications. The lessons learned from key stakeholders in the SBML community have greatly contributed to the initial success of SBOL. Finally, similarly to SBML circa 2001, there is now a confluence of technology and sentiment in the synthetic biology community favorable to the introduction of a standard for data exchange.

3.2 THE SOLUTION INSPIRED BY A DESIRE FOR SHARING

Analogous to the open source software movement a large segment of the synthetic biology community, shares a strong philosophy of openness and desire to share. Such practices are embodied by the principles promoted by synthetic biology organizations, such as the BioBrick Foundation and iGEM in promoting the open exchange of information for engineering biological systems. As Calvert [114] points out, these beliefs strongly contrast with the goals of commercial synthetic biology ventures. Proprietary models must rely on patent based protection, demonstrated by the practices of the J. Craig Venter Institute. However, synthetic biologists pursuing both principles want to realize the vision of programmable biology: where the design is defined first and the constructed biological system behaves as specified in that design. In order to facilitate the design and construction of higher-order gene circuits the increased complexity necessitates reuse of parts and modules [10]. The analogy to computer programming inspires many within the field. Therefore, the benefits of an open approach to carrying out synthetic biology resonate well with many of the researchers. The open source software movement has been widely recognized as a positive influence on the proliferation of information technology. At least some open source components are found in many software products today [115]. Freely and openly distributing source code allows other programmers to re-use it within their own software. The equivalent practice can be seen in the open distribution of DNA components used for synthetic biology. The spread of the ideals of openness in the context of synthetic biology can be largely attributed to the work of the BioBrick Foundation (BBF).

The BioBrick Foundation (BBF) is a public-benefit organization created to ensure that synthetic biology would serve the public interest to benefit all people and the planet. The BBF strongly supports this mission through the support of open sharing of information. Most relevant to the development of the SBOL community, is that the BBF operates Open Wet Ware (OWW), a wiki-based community which provides a virtual location on the web for the sharing and collaborative editing of laboratory and group web pages, courses, protocols, and blogs. Most of the OWW content is centered on synthetic biology and related research areas. The OWW platform helped SBOL developers form the beginning of our community, through the use of its wiki pages and mailing lists. The BBF, originally via OWW, supports SBOL through the distribution of the specification documents through their Request for Comments (RFC) process. Additionally, the BBF operates the BIOFAB, a professional facility to produce well characterized, reliable, standardized biological parts. The BIOFAB has been one of our closest collaborators throughout the work to define SBOL. Furthermore, the BBF shares its origin and founders with the iGEM competition, where the value of sharing is a key principle driving its growth.

The International Genetically Engineered Machine (iGEM) Competition is an undergraduate Synthetic Biology summer program during which students compete by building new biological systems. The iGEM Foundation, which operates the competition, promotes an open community and collaboration by giving each team a kit of biological parts from the Registry of Standard Biological Parts. Student teams incorporate these parts and their own new parts to build and test their designs. At the end of the summer they are required to submit their designs and send the newly created DNA to the Registry so these can be incorporated into next year's distribution.

This community has adopted the open philosophy as part of the effort to cope with the large complexity of biological system design. Additionally, they embraced principle of standardization [9], as it is necessary in order to enable the composition of any of the components with another, from the growing collection. More broadly, in the synthetic biology community there is strong recognition of the need for standardization to enable the engineering of biological systems.

The popularity of the BioBrick assembly format in synthetic biology can be recognized from the success of the Registry of Standard Biological Parts [57]. The Registry is a web-based Wiki information system which contains information about approximately 7,100 BioBricks [77]. By virtue of the open philosophy of Wiki communities, the Registry web site allows synthetic biologists to edit the information as an open science resource aimed at sharing of information between synthetic biology researchers. This resource establishes the links to the corresponding repository of bacterial clones located at the Registry, a record for DNA sequence of each part, references to assembly standards, and information which the users choose to share which varies between entries. However, as it is a portal intended for human users, it lacks agent- or computational-based access to content, with the exception of DNA sequence information, limiting the ability to find parts for a new design, as well as limiting the quality or kind of follow-on analysis services that might be provided for synthetic biology modelers and researchers.

A top-down design process, such as is envisioned for programmable biology, requires a large collection of components which can be used to fulfill the requirements of design specifications. Furthermore, managing DNA sequences on the level of nucleotides becomes unwieldy even for current molecular biology software applications [62]. Therefore, information technology standards are needed to aid the design process and re-use of DNA components found in repositories, such as the Registry. Embracing technical standards serves as an incentive for participating in a community, and this can encourage sharing. Conforming to a community standard pays dividends by allowing the integration of others' work into your own and *vice versa*. Also, conforming to a standard simplifies contribution back, by reducing the burden of useful description down to simply attesting compliance with the standard.

3.2.1 *Building on Semantic Web Standards*

There is a common philosophy between the open synthetic biology community and the Linked Open Data (LOD) community (see Chapter 2 for more on LOD). This commonality inspired Professor John Gennari and me to propose a semantic technology solution for information exchange in synthetic biology. The prior work toward semantic technologies designed to support exactly the kind of open information sharing and exchange which can benefit synthetic biology research.

3.3 FROM COLLABORATION TO COMMUNITY

Throughout the development of SBOL, we followed a grassroots model. SBOL is driven by the community of synthetic biology software developers. These software developers are the stakeholders in the effort to standardize data exchange. The critical role of the community, represented by the Synthetic Biology Data Exchange (SynBioDEX) Group and the SBOL Developers, stems from the commitment of its members to agree to support SBOL. Just as Hammond [116] describes stakeholder to 'buy in' being critical for adoption of health data standards, it is the key to the success for synthetic biology standards. To get buy-in from the community we followed an open development process. In the open process the stakeholders are continuously involved in the formulation of the standard. This engagement in its development has in a large part contributed to its success.

In this section I describe the history of the community. While we pursued our own goals, we followed the practices of standards development groups, such as SBML, described in Section

3.1. We applied their experience in order to increase the chances of success. The history of SBOL is comprised of the elements that were needed to develop a community to support this standard, and illustrates the social process which occurred. Our group is grassroots based. It formed as a small collaboration between interested researchers, but it grew into a community. Now, there are more than 50 individuals from 10 companies and 15 universities and research institutes participating. Within the four years of working together, we shaped the group into a community. We adopted a governance structure, formulated a set of principles to follow, and we strive towards a common set of goals. As membership in the community grew, the identity of the group changed from a collaboration to develop PoBoL, to the SynBioDEX Group, to the current SBOL Developers group. The consensus process we adopted emerged from face to face workshop meetings. The meetings provided a dedicated time for feedback and strengthened the relationships between the individual researchers. Between workshops we used online tools such as mailing lists, wikis, and real-time editing of documents to collaborate. We used a public web site (sbolstandard.org), peer reviewed conferences and journal publications to promote the work more widely. Additionally, we deployed software libraries and received feedback. Deployment is described in Chapter 5. The history of the SBOL community demonstrates the first step of becoming a standard, stakeholder buy-in.

3.3.1 *Grassroots and boots*

The Synthetic Biology Data Exchange group started with a few interested researchers. Scientists, motivated to improve the capabilities of synthetic biologists through data exchange, continued to join the group. The goals for the group were established in a spirited discussion at the first workshop held in Seattle. In Section 3.3.2 I describe the first workshop. The outcome of this meeting was the submission of a Request For Comments documents to the BioBrick Foundation [117], which specified a core data standard for information about BioBrick parts. Emphasizing its preliminary nature, the format was named the Provisional BioBrick Language, (PoBoL). The PoBoL RFC document demonstrated a concrete outcome of the first meeting. Tangible results following the workshops prompted synthetic biology software developers interested in standards to join the Synthetic Biology Data Exchange Group and to organize follow-up workshops. The members of this group represent stakeholders from the synthetic biology community, especially researchers who develop software tools and will most immediately benefit from the standardization of data exchange. Following the next meeting at Stanford University in 2009 (see Section 3.3.2), the name of the main effort was changed to better reflect its broader ambition to Synthetic Biology Open Language (SBOL). This group of researchers has morphed and evolved in terms of membership, but the goal remains the same. Our focus is to forge a consensus on terminology and the technical requirements needed to standardize the computational representation of information used by synthetic biologists. The origin and identity of the organization formed around this shared interest and goal. The foundation of the group around the objective, in contrast to any mandate or top-down policy, has encouraged enthusiasm within the community and with funding organizations. However, going forward, support of the standard through policy and funding incentives will be greatly beneficial (Section 3.5.1). I describe the policy goals for data standards in synthetic biology in Section 3.5.1. The goal and promise of enabling new possibilities in synthetic biology is what has motivated new members to join.

3.3.2 *Workshops*

I was first introduced to other researchers interested in standards for synthetic biology at the Standards and Specifications in Synthetic Biology Workshop held at the Talaris Conference Center in Seattle, WA in April 2008. The workshop was organized by Dr. Sean Sleight, Deepak Chandran and Prof. Herbert Sauro from the Department of Bioengineering at the University of Washington and sponsored by the Microsoft Computational Challenges in Synthetic Biology Initiative. The organizers invited the researchers due to their demonstrated interests in developing a broad range of standards for synthetic biology. The invited talks and discussion covered BioBrick standards, measurement, storage, information retrieval, design, modeling, and software tools. Notably, one of the talks was about community support. In this talk Dr. Michael Hucka recounted his experience of leading the systems Biology Markup Language (SBML) community. He made it abundantly clear that active engagement of the software developers in developing SBML was critical to its success. His talk set the foundation for SBOL as a community developed standard. But, it was the open discussion section of the meeting which was most exciting and productive. Using a whiteboard, we sketched out the first ideas for a common data model. The decision that the project should be free and open source was quickly made and seemed like an implied assumption by many participants. We named it Provisional BioBrick Language (PoBoL) and a small group of us, Raik Gruenberg, Mackenzie Cowell, Jason

Morrison, and I, continued to work on the technical material in the months after the workshop. The prototype model and implementation in OWL gave us concrete results to discuss, but most importantly we had met each other and formed a group around the PoBoL project.

I began participating through the face-to-face meeting at this workshop. It is there I formed the first connections with other researchers working in this area. Afterwards, I continued to collaborate and to nurture the connections I established at this and the workshops which followed: (Figure 3.1) The Synthetic Biology Data Exchange Working Group Meeting at Stanford University in July 2009; the SynBioDEX Group Meeting in June 2010, associated with the International Workshop for BioDesign Automation (IWBDA) in Anaheim, CA; the SBOL Workshop at Virginia Tech, Blacksburg, VA in January 2011; the SBOL Workshop after the IWBDA meeting in San Diego, CA in June 2011; the 6th SBOL Workshop which took place at the University of Washington in Seattle, January 2012; and the SBOL Meetup after the IWBDA meeting in San Francisco, CA in June 2012. The enthusiasm for the meetings has grown, and their frequency has increased to twice a year. The next planned meeting will be in London, increasing the exposure of SBOL to the European community.

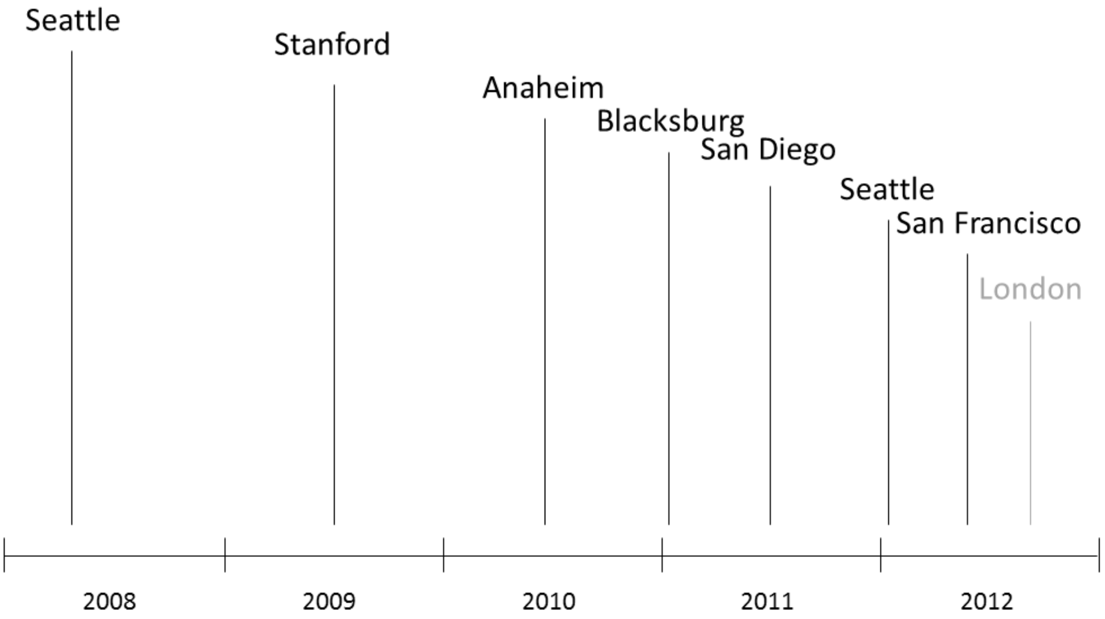


Figure 3.1. SBOL Workshops 2008-2012.

These meetings are critical in establishing the community, growing interest in, and gathering commitments to support the standard. Developers of software tools for synthetic biology researchers attend these meetings to get familiar with the standard, but most importantly to meet the other members of this community. While attendance at workshops is expected, it is not enforced. These person to person meetings help establish the collaboration relationships which have enabled me to carry on the work with a subset of the participants. In order to keep the cost of participation in the development of SBOL at a minimum day-to-day we communicate using an online mailing list

3.3.3 *Mailing list*

Immediately after the first workshop in Seattle, we began communicating via e-mail, and in July 2008 we established a PoBoL mailing list which eventually grew to sixteen members. We pursued the open source community project model and made materials publically available online. After the 2009 meeting at Stanford University, we renamed the project to the Synthetic Biology Open Language (SBOL) and named the group the Synthetic Biology Data Exchange group (SynBioDEX). At this point we transitioned to a mailing list using the SynBioDEX name, in part to attract researchers interested in discussing the data exchange more broadly then our proposal for the SBOL standard. This mailing list remains active and is the publically available mailing list, which anyone, with interest in the domain of data standards for synthetic biology, can join. However, this mailing list receives very few emails and some of the attendees at the Virginia Tech workshop expressed concerns that they do not know who is on the mailing list and that due to its publicly open nature they are hesitant to send informal e-mails conversational in nature. In order to remove the barrier of these fears we decided to form a closed list, with membership limited to those expressly interested in SBOL development. Furthermore, each new member to the list would be introduced when joining the list. This new SBOL Developers (Section 3.4.1) mailing list has become a highly active exchange of ideas about the SBOL standard. Most of the communication about the development progress and discussions about the technical details now occur on the SBOL Developers mailing list. E-mail, mediated by the mailing list, continues to be the most frequent and most important medium for communication in this community.

3.3.4 *Champions of SBOL*

Information technology champions are the individuals who have significant influence on adoption of new technologies in their community [118]. Champions communicate a compelling vision about the benefit of an innovation to the rest of the community. They take a creative idea, chaperone it when resistance is at a peak, and persist until it succeeds or fails [118]. Their work involves communicating among the members of a social system that the idea is sound. In the academic health sciences setting, champions must be nurtured if an information technology innovation is to be successfully diffused [119]. Champions contribute to the innovation process by energetically and eagerly promoting the new information technology. The challenge they face is that before the new innovation is built or adopted the full impact of the benefits cannot yet be demonstrated. The result of their work is community support and erosion of resistance to change. For the SBOL community this role was largely fulfilled by Dr. Cesar Rodriguez, who contributed greatly to its successful adoption.

Dr. Rodriguez took on the role of an information technology champion [118] for SBOL. To find community support, he solicited interest in one-on-one meetings with potential collaborators, during which he described the benefits of the SBOL effort and garnered support. He was able to attract the innovators and early adopters before the benefits of SBOL could be demonstrated. This personal approach is; therefore, it requires dedicated effort. Dr. Rodriguez's support was critical to our success in building such a large and diverse membership.

I met Dr. Cesar Rodriguez at the Stanford workshop in 2009 organized by Prof. Drew Endy. Dr. Rodriguez became my closest collaborator on the SBOL project and co-led the SBOL project with me. The prominent and visible research groups, such as Prof. Endy's group, are critical to driving adoption of innovations. Prof. Endy's is the director of the BioBrick Foundation and was especially important in reaching the iGEM and Open Wet Ware communities. His position as an opinion leader [103] in the broader synthetic biology community in addition to Dr. Rodriguez's advocacy work contributed significantly to gathering support for SBOL. Dr. Rodriguez's efforts led us to secure commitment to implement SBOL in software tools, from both academics and industry alike.

3.3.5 *Free online collaboration infrastructure*

Standards development requires distributed collaboration among the participants between workshops. Different tools are needed to organize technical materials, exchange ideas, plan the next in person meetings, and publicize the work. The coordination of work among the developers of the standard is paramount to keeping the stakeholder engaged throughout the process. For example, the online discussion and exchange of ideas which led to the Stanford meeting occurred under the auspices of the BioBrick Foundation's online infrastructure for collaboration so we could share our results. After the first meeting in Seattle 2008, we used the BioBrick Standards mailing list to communicate the results more broadly. Then, we used the BioBrick Foundation Request for Comments venue to publish the proposals for the PoBoL specification (Galdzicki 2009) and the use of RDF for synthetic biology data exchange (Gruenberg 2009). Following the Stanford 2009 meeting we began using the BioBrick Foundation's Open Wet Ware, a wiki designed as a web based place for labs, individuals, and groups to organize their own information and collaborate with others.

The importance of online tools for organization grew with the number of collaborators. Eventually in May of 2010, we turned to our own domain *sbolstandard.org* hosted by the Google Apps service to increase the flexibility of the website design and visibility of the work. Tools such as OWW and Google Docs, which we currently use, gave us the ability to collaboratively edit documents and to publish them on the web. These capabilities were particularly important for *ad hoc* discussions, to circulate document drafts, and to quickly disseminate small achievements. The Google Docs application has become very useful in coordinating document editing with the increased number of participants. For example, approximately twenty individuals participated in the editing of the SBOL v1.0 specification document [120]. We take advantage of the real-time editing capabilities during regular conference calls and during face to face meetings for taking notes as a group. Additionally, we use the GitHub source code management website to distribute the code of the software libraries we have created. We continue to rely on the BioBrick Foundation Requests for Comments mechanism to publish the specification documents in the broader synthetic biology standards community. The online collaborative infrastructure described here is free for use to the SBOL group, as it would be to anyone else. This lack of an upfront cost to form the community has greatly contributed to enabling SBOL to be developed. This grassroots effort, largely without dedicated institutional funding to support the infrastructure needed, succeeded by using free online collaborative tools.

3.3.6 *Public dissemination of the standard*

In addition to developing the technical materials online, we disseminate information about SBOL through peer reviewed conference and journal publications. This channel of dissemination of the work is traditional in the sciences. In particular we presented abstracts as posters and oral presentations about the ongoing SBOL development work at the International Workshop for BioDesign Automation (IWBDA) in 2009, 2010, and 2011. The IWBDA attracts those researchers working towards solutions which can benefit from a standardized information exchange infrastructure. Many IWBDA participants have joined the SBOL effort. Additionally, following the IWBDA 2010, a small group of SBOL developers, led by Dr. Jean Peccoud submitted a letter to the editor at Nature Biotechnology in which we call for the publication of full DNA sequence with synthetic biology journal submissions (Peccoud 2011). Fully specified DNA sequence is required to allow researchers to re-use synthetic biological systems such as we represent using SBOL. This publication was followed by the publication of the Synthetic Biology Parts Knowledgebase (SBPkb) in PLoS ONE (Galdzicki 2011), in which we demonstrated the benefits of SBOL in information retrieval for design. The SBPkb is described in detail in Chapter 5. Publication of the standard in peer reviewed literature and presentation of the ongoing development at conferences remains a critical method for disseminating information to gather support, but also credibility for the standard.

3.3.7 *Deployment of SBOL*

To solicit feedback about how SBOL functions we arranged for deployment test sites. I received significant help from the SBOL community in the form of appraisals at various stages of the development process. The deployment and feedback received is described in detail in Chapter 5. The feedback on the functional software libraries (Chapter 5) and technical documentation (Chapter 5) did not only contribute to the standard, but also aided in the building of collaborative relationships. These relationships form the basis of the SBOL Developers group as a community.

3.4 SBOL COMMUNITY ORGANIZATIONAL STRUCTURE

The growth of the SBOL community necessitated a formal organization of the group developing SBOL. The progression from a small collaboration to a community necessitated changes to the operation of the group as membership grew and as we aim to create a community which can sustain development of SBOL going forward. In this section I describe the adoption of an organizational structure within the SBOL community. I define the SBOL Developers group and describe the role of the SBOL Editors and Chair based on the group's organizational documents.

Starting in early 2011, the SBOL Developers group perceived the *ad hoc* and undocumented organization as a barrier to understanding how to interact within the group. Based on feedback from senior researchers in the group, it became clear that the group needed a plan to help the members participating better anticipate roles in the group. A governance structure would help specify the responsibility for different aspects of the development process. Additionally, a document explaining this structure would help communicate the identity of the community to outside researchers interested in SBOL. Lastly, creating dedicated roles would offer recognition to the members who do significant work for the community. As the SBOL Developers group had grown to approximately fifty individuals, an organizational structure was needed for the group.

At the San Diego 2011 workshop we adopted a governance plan proposed by Prof. Herbert Sauro. With this organizational structure we adopted the position of Editor, based on the experience of the SBML community and the IETF. That day the SBOL Developers elected three editors, Cesar Rodriguez, Mandy Wilson, and me. Later, at the Seattle 2012 Workshop we added the position of a Chair. The SBOL Chair position designated the senior representative of SBOL, especially in interactions with external organizations which expect a senior researcher to be in charge. Prof. Herbert Sauro was elected unanimously as the SBOL Chair. The SBOL developers at the meetings approved the proposal by a hand vote. We agreed to formally use voting to make decisions during meetings and on the online discussion forum. These basic tenets of how the group operates moved SBOL from an improvised collaboration to an organization better prepared to introduce new members to the community and the development of a data exchange standard.

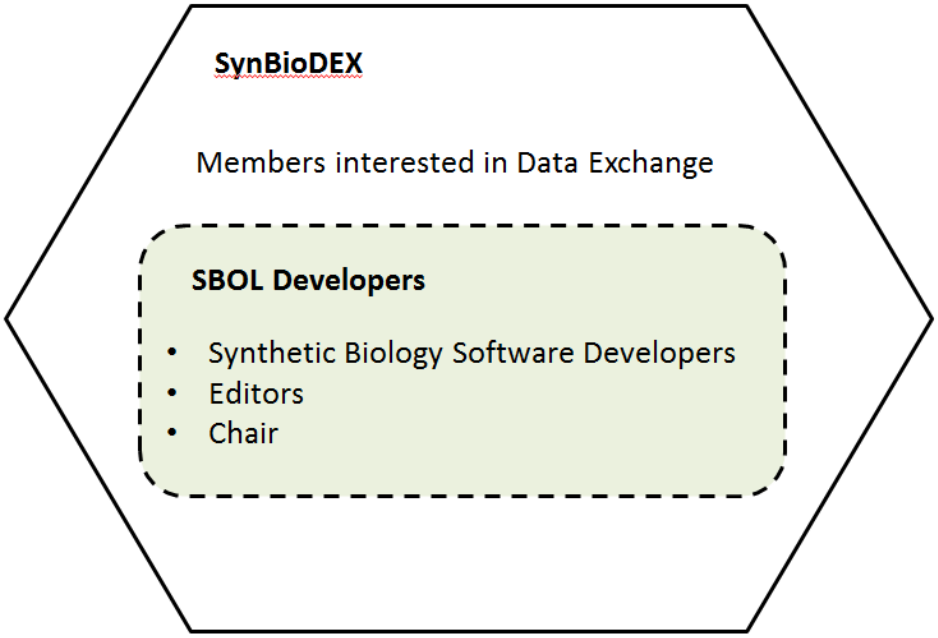

SynBioDEX
Members interested in Data Exchange
SBOL Developers
Synthetic Biology Software Developers
Editors
Chair

Figure 3.2. Synthetic Biology Data Exchange (SynBioDEX) group community includes the SBOL Developers and other researchers interested in developing standards for information exchange in the field.

3.4.1 *SBOL Developers*

The SBOL Developers Group is diverse, ever changing, and requires a dedicated effort to maintain. Its members make up those active in the SBOL community. Most of its members are involved in the development of software for synthetic biology at their respective institutions. They have not only committed to use and comply with the SBOL standard in their own work, but to also develop, improve, and maintain the shared resources. Membership in the Developer's Group is open to all interested parties, although the SBOL Editors check with each other before adding new members to verify that that the new party has a legitimate interest and reason for joining. To date no one has been denied membership. Since September 2011, joining the SBOL Developers Group requires members to accept an invitation which asks them to commit to: 1) Attend the SBOL Workshops; 2) Deliver on time items they have committed to develop; 3) Participate in the ongoing discussions on the SBOL Developers mailing list; 4) Support the SBOL standard in their software projects; and 5) Provide constructive feedback for improving the standard. Membership is actualized by joining the sbol-dev Google Group, an email discussion group and participating in person at workshops. Decisions to make a change to the standard are determined by taking an online vote or by rough consensus at face-to-face SBOL workshops. The members of this group have one vote each on any issue. Therefore, the SBOL Developers are the group who ultimately control the direction of the standards development through a democratic process. Below I describe the roles of individuals within the SBOL Developers group, the SBOL Editors and Chair, who have additional responsibilities.

3.4.2 SBOL Editors

Good documentation is critical to developing any standard. The Editors are responsible for maintaining the documentation, consistency between documents, and acknowledge all contributions. The editors write, make changes, additions, and keep track of the shared documentation, such as the specification documents. The Editors maintain a centralized document repository and the public website. However, all material is created based on requests from the SBOL Developers after discussion in the SBOL Developers mailing list. SBOL Developers can also submit corrections and amendments to the Editors. The text of SBOL specifications is kept publically available. Therefore, Editors are responsible for maintenance of the SBOL web site, and electronic mailing lists; helping to organize the organization of SBOL events, such as the SBOL workshops; and coordinating the publication of SBOL in peer-reviewed journals. Within the source code repository editors ensure code consistency, adequate code commenting, and the availability of tutorial material. To assist the SBOL Developers in reaching rough consensus, the Editors are responsible for establishing voting mechanisms. In addition to the roles described above, the SBOL Editors facilitate the organization subgroups concerned with SBOL extensions. Even though Editors are responsible for the maintenance of the central online infrastructure they do not unilaterally decide on new functionality for the standard, nor decide on the priorities of the group. Decision making in the SBOL Developers Group is established by rough consensus determined by voting.

At the January 2012 workshop in Seattle, the SBOL Developers decided to add two new editorial positions, for a total of five. Drs. Ernst Oberortner and Matthew Pocock were elected for two year terms. At the time of this writing, I, Michal Galdzicki, University of Washington, Ernst Oberortner, Boston University, Matthew Pocock, Newcastle University, Cesar A. Rodriguez, Genome Compiler Corporation, and Mandy Wilson, Virginia Bioinformatics Institute are the SBOL Editors. These Editors are responsible for the maintenance and the quality of the documentation of the SBOL standard.

3.4.3 *SBOL Chair*

The expectation of sustainability of the SBOL effort in the long term necessitated the election of a principal of the SBOL Developers. Prof. Herbert Sauro has served in this role from the beginning; therefore, the group unanimously elected him as SBOL Chair during the Seattle 2012 Workshop. The position of SBOL Chair was created to recognize his contribution, to re- assert the role externally. The Chair is charged with the responsibility to uphold the guiding principles, responsibilities of the Editors, and to provide continuity beyond the terms of the two year Editor terms. Unlike the Editors, the Chair is in a supervisory role, not responsible for day-to-day operations. However, the Chair oversees, and must ensure, the progress towards the long term goals of SBOL. For example, the Chair is responsible for representing the overall SBOL project to funding agencies. Additionally the Chair serves as the communicator, representing the SBOL community to the press, although he can delegate this role. SBOL Developers aspire to create a sustained standards development effort. There is a need for a primary representative to interface with other organizations and ensure longevity of the standards effort.

3.5 SBOL Values

Shared values are the essence of a community. They establish the cohesiveness of a group working towards a goal while accepting new members. Making the common values of a group explicit allows them to be conveyed to new members at the outset, reducing the likelihood of miscommunication of intent. The growth of the SBOL community necessitates a statement of the principles by which it operates. Without an explanation of what the community believes is the best path forward in its development, long-term growth could be jeopardized.

At the January 2012 SBOL workshop in Seattle, Prof. Drew Endy challenged the SBOL Developers to prepare the community for growth beyond the current model of group of collaborators who understand each other implicitly. He asked, “What values does the SBOL community identify with?”. In practical terms, enumerating these beliefs would represent the group to prospective members and the public. A statement of values for SBOL should answer: 1) Why the SBOL Developers are working on standardization, and 2) Why does the group makes a particular choice? For instance, why is the SBOL language “open” and what does that mean? We need an explicit statement of values, to achieve broad adoption the SBOL standard in the synthetic biology community.

The SBOL Developers group needed this explicit statement of values. Prof. Endy started the discussion with the workshop participants, and it continued amongst the Editors. The ideas raised led me and the other Editors to draft a set of themes of beliefs which hold across the community. These themes state what our community believes. To become the *de facto* standard, SBOL must be: **used**, **useful**, and **agile**. Furthermore, these goals are supported by making SBOL **free**, as in beer and speech, and **open**; through transparency and accountability of the governance. Finally, SBOL must be made **for the community** and **by the community**. I believe that by applying these seven themes in the work will help us, SBOL Developers, to make biological engineering technologies available to a wide range of innovators. We have made significant progress towards these goals. We will accomplish the realization of the complete vision by solving the challenging problems faced in transferring synthetic biological designs and experimental data between tools electronically. The SBOL community will develop new technologies and produce the results which, we believe, will ultimately benefit society. This ultimate goal will be accomplished by a community of individuals and organizations in the Synthetic Biology field. Therefore, to achieve these goals, we must work together towards this common goal and allow the values I describe below to guide the SBOL community.

- SBOL must be developed with an eye to promoting its adoption, meaning that it must reflect the emerging needs of the synthetic biology community.

SBOL development must be flexible to the changing needs and practices of the synthetic biological engineers. The development of SBOL should not impinge or hinder the development of new technologies, so it must allow for free extension of the standard. The SBOL community must develop best practices for standard development to support this goal. However, a standard which changes too often is not a standard at all. Backward compatibility is essential to continue the support of applications that are not current to the latest changes. A careful balance between enabling communication and freedom to innovate must play out in practice.

- SBOL, as a standard language, should be *libre* and *gratis*, which means anyone should have the freedom to use it, extend it, and to redistribute copies of any SBOL project artifacts, with or without extensions, without restrictions, and especially without price.

We believe keeping these freedoms will allow SBOL to be used in the broadest range of applications. Development of SBOL as a free standard means, that all documentation, software, and example data should make these freedoms available. Any individual or organizations should be able to use and extend SBOL, and publish and sell derivative work as a part of their individual work; they should not be inhibited by the restrictions placed on SBOL. On the other hand, they should not be permitted to place restrictions on how others use or extend SBOL.

- The SBOL standard should be developed using an 'open source', transparent**,** strategy.

In order to build a strong community and trust, all documentation, software, and example data should be made publicly available on the web, so anyone can read, use, and copy it. This means full text documentation, source code, and data files describing SBOL will be available without cost on the web. To accomplish this, we will strive to maintain web resources so that these documents are publicly available to all.

- The SBOL community is inclusive, and we welcome participation from all interested parties provided they do not impinge on the work of others in the group.

Joining the SBOL Developers Group requires members to: a) Attend the SBOL Workshops, deliver items on time they have committed to develop; b) Participate in the ongoing discussions on the SBOL Developers mailing list; c) Support the SBOL standard in their software projects, and; d) Provide constructive feedback for improving the standard. Through these actions, SBOL developers aim to develop, maintain, and comply with the SBOL standard within their own projects. Participation empowers the SBOL developers with the collective right to steer SBOL's development. The SBOL community supports these values through democratic processes, such as voting and discourse, in order to reach decisions as a group. Each individual's commitment to the group helps to build trust, enhances the group's growth, and creates an atmosphere in which a high standard of quality is mandated. We can depend on SBOL's continued support through the commitment and investment of the members of the SBOL Community.

- The SBOL community will strive to engage external organizations which are interested in understanding, supporting, and utilizing SBOL.

The development of SBOL will benefit from engaging other organizations, standards, and viewpoints. For instance, we will communicate our goals to the broader scientific community, as well as public and governmental agencies. We will be responsive to the needs of the synthetic biologists who ultimately derive value from its use: The publishers who communicate about or use SBOL, as well as the governments and companies who encounter SBOL in their domains. In all of our practices, we will coordinate with these relevant external stakeholders to uphold our values.

- To engage with external stakeholders, the SBOL community will actively pursue partners who can help to represent SBOL in their broader communities.

The SBOL Developers alone will not be able to effectively meet their goals unless SBOL becomes embedded into the larger synthetic biology community and society at large. We rely on the leaders within the SBOL Developer group for support of this value. We depend on this relationship with our external stakeholders to ensure that our research results in ethical, timely, and safe implications for synthetic biology.

- The SBOL community treasures the universal scientific community values of integrity, honesty, and increasing public knowledge.

Ultimately our work on the SBOL project contributes to the ongoing research effort to establish standards in synthetic biology and more broadly to establish an engineering approach for biological systems. Within this work we aim to benefit this approach and are accountable to each other and society.

The values described here will ensure SBOL continues to be a free and open standard for the communication of synthetic biology knowledge, information, and data. The SBOL community will support the synthetic biology community through its unique grassroots based approach in building the standard. SBOL serves as a mean of data transfer from disparate biological engineering systems. To enable data exchange between these systems developed by various organizations, the SBOL Developers rely on supporting partners in order to achieve our goals. The partners' expertise in the diverse areas of the synthetic biology domain is a technical challenge and an asset to the community. Values will drive our work, anchor our community, and are reflected in the technology and results we create.

3.5.1 *Policy*

In the attempt to provide a long term solution to aid the vision of engineering biology, there is also the need to pursue a complementary approach, a top-down model. The top-down model would involve funding organizations to enact policies which require sharing of data. Historically, policies mandating data sharing in a standardized form were found in the environmental and social sciences, where studies can last 30 years and require long term information management plans [121]. Field, et al. make a case for the need to enact such policies in the 'omics or high throughput data fields which are generating massive amounts of data.

Such an approach may be complimentary to the community based model to incentivize participation in submission of standardized data, a process which places a significant cost on the individual researcher in terms of time and consequently funds. In creating and maintaining institutional infrastructure to manage the information, centralizing such an effort does provide economies of scale, although with a substantial direct cost. Additionally, regulatory agencies have a strong interest to encourage participation in order to review outcomes of synthetic biology efforts as necessary. Such policies can be enforced by grant application data sharing plans, specified time periods, and in a accordance with international standards. Journal referees and editors can uphold and extend these policies analogous to the accession number for DNA sequences. Once consensus is reached on the value and need for information sharing, a policy mandating timely and public release of data will be needed [121]. Such policies, which obligate the researchers to submit information in a standardized form would serve a common aspiration in synthetic biology: biological systems need to be easier to engineer. Standardized data aids in the gathering, preservation, and amalgamation of research output by greatly reducing the barrier to accessing the knowledge created.

The SBOL community has had an initial success in promoting such policies with a funding agency. The recently funded $23.6 million program The Living Foundries: Advanced Tools and Capabilities for Generalizable Platforms (ATCG) by the Defense Advanced Research Projects Agency (DARPA) carries the provision requiring SBOL compatibility. A condition of funding is that, "*To encourage interoperability, all applicable design tools and databases developed under the ATCG program should be compatible with Synthetic Biology Open Language (SBOL) core data model.*" The SBOL Developers will help those receiving funds to implement and adopt SBOL.

3.6 SUMMARY

To build a successful standard, a community must form around the vision of its benefits. This vision is reflected in the values and policy outlook for SBOL. Long term benefits for the synthetic biology community will be derived from the well supported technology developed by the SBOL community. Specifically, standardization and dissemination of synthetic biology knowledge resources will greatly increase the potential for its re-use by downstream researchers and engineers.

I based my strategy on the lessons learned from three examples of successful prior standards. In my research I worked with many collaborators whose contributions were critical to the success of building the SBOL community. I described the strategies we took to build a grassroots community. For example, I found eager collaborators through workshops and personal communication, similarly to the early efforts of the PDB founders. A few of these collaborators took on critical roles to the success of the standard. For example, Dr. Cesar Rodriguez championed the standardized information technology and engaged many external stakeholders. These stakeholders became active participants, similar to the MIAME standard. They helped the new standard to grow, and they helped to disseminate its use. Following the example of SBML, we matured as a community through persistence in its development and maintenance. Additionally, we adopted the well tested SBML governance structure with only slight modifications. Through the experience of building SBOL and its community and to guide the development in the future I made our values and future policy ambitions explicit. The goals of SBOL will remain to be used, useful, and agile.

SBOL is still in an incipient stage. It may not revolutionize synthetic biology research overnight; however by providing a template for standardization of data sharing in a research community, it may yet prove to be the catalyst for changes that go well beyond synthetic biology. The confluence of technology between semantic web and synthetic biology has allowed me to leverage existing tools throughout the development process. In the next chapter I describe the technical standard definition and in Chapter 5 how we kept the initial deployment simple, and made the standard immediately useful.

Chapter 4. SYNTHETIC BIOLOGY OPEN LANGUAGE

The result of the collaborative work I described in Chapter 3 is the Synthetic Biology Open Language (SBOL). SBOL is an open language for the exchange of synthetic biological designs. The overall goal of the SBOL is to facilitate unambiguous exchange of data among synthetic biologists and the software tools which aid their research and engineering. Therefore, its aim is to be accepted as the standard for exchange of data in the synthetic biology community. Most importantly, SBOL is a launching point for a community development effort. As software tools adapt to progress in the synthetic biology field, SBOL will need to evolve to meet the changing needs of synthetic biology researchers and engineers. SBOL has had significant success towards this goal as new researchers and software developers continue to join the SBOL community and adopt SBOL. Adoption of SBOL, the result of its deployment into the community, is described in Chapter 5. In this Chapter I present the SBOL specification; this is the technical description of the requirements which make SBOL a standard.

As a language, SBOL is composed of a vocabulary, a data model, and a computer format. The vocabulary defines a specific terminology for concepts. These definitions provide the exact meaning of the terms and help therefore establish their unambiguous use. The data model specifies the relationships of the concepts providing a structure for how the concepts must be organized. Finally, the format is designed for software tools to read and write. The format, is a serialization of the data model. The data is written in a textual format according to the structure of the model. The SBOL format provides a computer readable representation, which is also human readable.

SBOL is also free and open. The SBOL Developers agreed to make the terms of its use and development publically available. It is also free; anyone has the right to use it without any cost. The grating of these rights reflects the values of the SBOL community as described in Chapter 3, Section 3.5. These rights are reflected by the licenses on the software and documentation produced as part of the SBOL project. Furthermore, the strategy for ongoing SBOL development is based on a modular architecture which allows for anyone to extend the SBOL framework.

SBOL is defined by a formal specification document [122]. This document specifies the requirements for SBOL Core, the first and essential component of the overall modular architecture of SBOL. In Chapter 6 I describe future work towards these additional modules, SBOL Extensions. This specification details and explains the set of explicit requirements to be satisfied by the software implementation of the standard. The requirements are written in this document for the purpose of communicating the criteria to the software developers in the synthetic biology community who want to implement SBOL. The document follows the guidelines of the BioBrick Foundation Request For Comments (BBF RFC) Process as specified by BBF RFC 0 [65]. This process, a part of the foundation's Technical Standards Framework, is modeled after the well-established Internet Engineering Task Force (IETF) RFC process. One of my contributions to the collaborative project was to propose, draft, and maintain the specification document for SBOL Core. The specification is the formal technical documentation for the standard. The SBOL project's website (sbolstandrd.org) serves as a more practical form of documentation. The goal of the specification document is to define all the criteria necessary to implement SBOL.

Developers who implementation SBOL enable their software tool to transmit data to or from other SBOL compliant software. To make this process easier I also developed a set of software support libraries. The support libraries implement the standard and enable reading and writing of SBOL formatted files. I served as the primary developer for the first several rounds of the initial implementations, testing, and deployment. These libraries aided the adoption of SBOL because they help reduce the cost of implementing the standard for developers. Availability of the libraries is described in Section 4.9.

In Chapter 3 I described the process and the social context within which SBOL was created and in Chapter 5 I describe the process and results of the deployment of these implementations. In this chapter I describe the final SBOL Core specification, a document which all SBOL Developers approved after rounds of reviews and feedback. In Chapter 6 I describe future work on proposed extensions to the SBOL Core. In Section 4.1 of this chapter I restate the need for the development of SBOL as a data exchange solution for synthetic biologists. In Section 4.2 I introduce the specification document as a medium for the communication of the standard among SBOL Developers and software developers who will implement the standard within their software. In Section 4.3 I describe the scope, define the abstraction used to form the SBOL representation, and provide simple examples of DNA Components. In Section 4.4 I provide a brief description of SBOL Core to introduce the elements of the specification: the vocabulary (Section 4.5); the data model (Section 4.6); and, the serialization (Section 4.8). In

Section 4.7 I provide several examples illustrative of the representation's flexibility and breadth. In Section 4.9 I provide details of the availability of software and documentation. Finally, I conclude with a discussion of the technical capabilities of SBOL in terms of the benefits gained from compatibility with Semantic Web standards.

4.1 MOTIVATION FOR THE DEVELOPMENT OF SBOL

Synthetic biologists assemble segments of DNA to form devices and systems with more complex functions. A number of software tools have been developed to help synthetic biologists to design, optimize, validate and share these DNA systems, but the lack of a defined information standard for synthetic biology makes it extremely difficult to combine the appropriate applications into a refined systems process. To move the synthetic biology field towards best practices in engineering, synthetic biologists need software that can unambiguously interpret and exchange information about DNA components.

The lack of a standard exchange format means that synthetic biology information access and transfer is limited to manual efforts such as copy-and-paste and ad hoc scripts. Not only are these prohibitively lengthy approaches to data transfer, they can also be error-prone, either due to changes in the underlying architectures of the data sources or simple human error. Establishing a standard exchange format would not only save time, but would also help reduce the errors of manual transfer.

A standard exchange format would also provide a greater range of tools available to synthetic biologists. Although a wide variety of software tools exist, in some cases, software developers write their own applications due to the difficulty inherent in designing interfaces between software tools. A standard exchange format would alleviate their need to develop interfaces or duplicate software, which in turn would free them to develop new tools. Furthermore, if a standard format enabled programmatic access to public information resources, such as the Registry of Standard Biological Parts [57] and the BIOFAB Electronic Datasheets [123], software developers would be able to take advantage of these repositories directly within their applications. For example, CAD and modeling tools, such as TinkerCell [40] and iBioSim [124, 125], would be able to retrieve components for new designs. These scenarios are described in more detail in Chapter 5. A gene network design created by tools such as the Proto

Biocompiler [126] could be further refined by Eugene into collections of physical implementations [71].

In addition to improving the ability to share data across applications, a standard format would make it easier for synthetic biologists to exchange data with their collaborators at other sites. For example, synthetic biologist researchers could use software such as GD-ICE [127] and Clotho [128] to integrate their own data from local laboratory repositories with their collaborators' designs and publicly available data. A synthetic biologist who designed new DNA constructs with a software tool such as GenoCAD [129] could send them to a collaborator who would then review and edit them using Gene Designer [130, 131].

The definition of a standard for electronic information exchange would also help refine the standards for the DNA components themselves, through an iterative process of feedback to synthetic biology research groups concerned with standardization.

In summary, a standard exchange format would encourage reuse of existing DNA components, and it would reduce error caused by manual or *ad hoc* data exchange. Researchers could collaborate more effectively and it would save time which could then be used for advancing research and the development of new software tools.

Electronic exchange of synthetic biology information in a common format and using a common vocabulary will encourage the creation of interoperable software to support the information needs of synthetic biologists. Software developers will be able to write fewer data converters and offer access to a larger number of data sources. Finally, compatibility with the Semantic Web information technology I introduced in Chapter 2, will serve as leverage for the software developed by this broad community, as it will facilitate reuse of previously generated knowledge across independent research efforts.

4.2 Introduction to the Specification

The SBOL specification document communicates the requirements which the software implementations must satisfy when exchanging descriptions of DNA components. First, formalizing the specification as a BioBrick Foundation Request for Comments (BBF RFC) [122] helped the SBOL Developers form a consensus of the requirements of version 1.1.0 of SBOL Core. Second, it serves to inform future developers as to requirements for implementations of compliant systems.

During the drafting process, I received formal comments from nine members and SBOL Editor, Mandy Wilson helped a great deal in editing the specification document. After two rounds of revisions based on comments from the SBOL Developers mailing list, I submitted a final version to the BioBrick Foundation RFC repository for public comment. This document defines the vocabulary, a set of preferred terms and a core data model. This is a common computational representation which can be implemented to allow software to unambiguously interpret information about DNA components. The goal of the specification is to define the terminology and relationships as explicitly as possible.

4.3 INTRODUCTION TO SBOL

In order to provide a shared understanding between engineers seeking to exchange DNA designs, SBOL provides a common definition of the concepts needed. The work to define SBOL Core provides the fundamental elements which serve as a start of the standardization process. The SBOL Core is the first module of SBOL. It is the start of a systematic solution for the representation and exchange of the vast and inherently complex biological systems information needed for synthetic biology. The overall objective of the SBOL project is to represent and manipulate data that spans scales from plasmids, to cells, to tissues. Beginning the standardization process at the DNA level is a prerequisite necessary to realize the full potential of synthetic biology.

To encourage expansion of this core's capabilities, the guiding principle is openness. Therefore SBOL allows and expects extensions to the Core, proposed by the community. This collaboration in defining the common information exchange framework is driven by the community of researchers participating in the Synthetic Biology Data Exchange Group [132]. To easily allow for further expansion of the standard the group follows an open process for the evolution of SBOL [25]. This process starts with the definition of SBOL Core. Below I define its scope, abstraction level, and provide simple examples of to illustrate its intent.

4.3.1 *Scope*

Version 1.1.0 of the SBOL Core data model is limited to the description of discrete segments of DNA: DNA components. To remove ambiguity when specifying the design of synthetic DNA, the information about DNA components is structured. DNA components described using the SBOL core data model may have an associated DNA sequence, or they may be left as abstract descriptions. This flexibility allows for the description of DNA component designs which have not yet been realized, as well as those which are specific implementations of that design.

This version does not, however, provide a mechanism to represent the biological complexity of complete cellular systems beyond the DNA level. Additional biological knowledge needed to engineer aspects of complete biological systems will be modeled by future SBOL extensions. Furthermore, extensions of SBOL may add the ability to describe DNA components before and after a process, such as assembly, evolution, or implementation of a design *in silico*. Existing tools, such as GenoCAD, Eugene, and TASBE already offer solutions for bridging the “before and after”, so they can provide a basis for future specifications.

4.3.2 *Abstraction Level*

Within SBOL, we consider DNA regions as elements of design for DNA circuits [54], analogous to electrical circuits [53]. This conceptualization of DNA segments as an element of design is a level of abstraction used to form the basis of engineering synthetic biological systems [9]. This level of abstraction (Figure 1) has been shown to be useful in the practice of forward engineering of biological systems [133]. Therefore, SBOL defines these elements as DNA Components (Figure 1) in the SBOL vocabulary, and represent them as computational objects in the SBOL core data model.

DNA Components form the basic objects used in design, assembly, testing, and analysis. For example, DNA components can be hypothesized to have a biological function, deemed necessary for DNA assembly processes, or serve the synthetic biologist as landmarks in analysis.

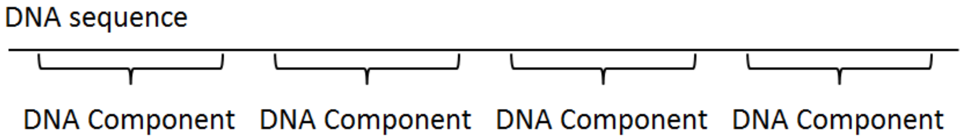


Figure 4.1. Basic abstraction level of identified DNA sequence segments as DNA Components.

4.3.3 *Examples of Simple DNA Components*

An example of the design of an expression cassette is shown in Figure 2; it illustrates DNA components along a DNA sequence. The symbols used represent their role in gene expression.

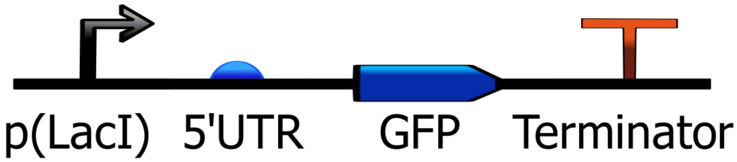


Figure 4.2. Example visualization of a series of DNA components, including a promoter, a 5'UTR, a CDS, and a Terminator. Together, these DNA components constitute the design of the expression.

An example DNA construct which fulfills the design specified in Figure 2 is the BioBrick™ Part BBa_J04430 [57] (Figure 3). This example illustrates the representation of a DNA construct as a DNA component with annotations.

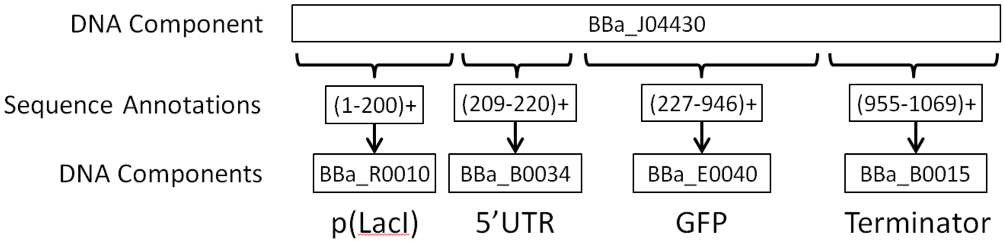
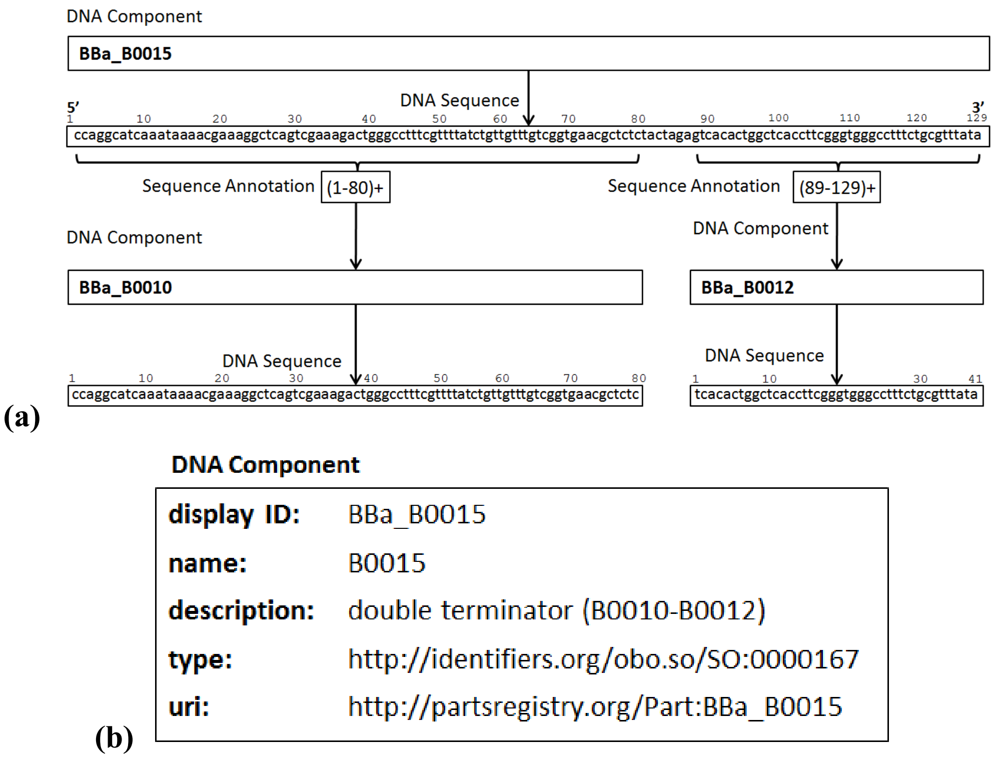


Figure 4.3. Diagram of BioBrick™ BBa_J04430 represented by SBOL objects. Sequence annotations of BBa_J04430 are used to describe the location of DNA components that are found within its sequence. These annotations are DNA components which correspond to the design specified in Figure 4.2.

Each DNA component can be further described with additional information (Figure 4).


DNA Component
BBa_B0015
DNA Sequence
5'
3'
1
10
20
30
40
50
60
70
80
90
100
110
120
129
ccaggcatcaaataaaacgaaaggctcagtcgaaagactgggcctttcgttttatctgttgtttgtcggtgaacgctctctactagagtcacactggctcaccttcgggtgggcctttctgcgtttata
Sequence Annotation
(1-80)+
Sequence Annotation
(89-129)+
DNA Component
BBa_B0010
DNA Component
BBa_B0012
DNA Sequence
1
10
20
30
40
50
60
70
80
ccaggcatcaaataaaacgaaaggctcagtcgaaagactgggcctttcgttttatctgttgtttgtcggtgaacgctctc
DNA Sequence
1
10
30
41
tcacactggctcaccttcgggtgggcctttctgcgtttata
(a)
(b)

DNA Component

display ID:	BBa_B0015
name:	B0015
description:	double terminator (B0010-B0012)
type:	http://identifiers.org/obo.so/SO:0000167
uri:	http://partsregistry.org/Part:BBa_B0015

Figure 4.4. BBa_B0015, a sub-component of BBa_J04430, ***(a)*** *with DNA sequence and sequence annotations.* ***(b)*** *To describe BBa_B0015 in more detail a set of fields for a human readable ID, name, and text description is defined. Structured information fields will enable basic retrieval capabilities ie.* type *using the Sequence Ontology [99], [134]; and* uri *as a unique identifier.*

4.4 Description of SBOL Core

The SBOL Core addresses the needs associated with sharing the design of DNA Component information across computer networks. It standardizes and facilitates information exchange for synthetic biologists using recommended practices and information technologies for data on the web. SBOL version 1.1.0 defines the representation of DNA designs. Core concepts are defined by the SBOL vocabulary, the result of a consensus process reached by the SBOL Developers group. The Core data model structures these concepts to describe DNA designs, and the SBOL format defines the serialization of the model.

SBOL Core vocabulary defines the core concepts using a simple definition to clarify their intended use in the structured description of synthetic DNA designs. SBOL Core terms are DnaComponent, DnaSequence, and SequenceAnnotation. To provide user-defined groupings of

DnaComponents, SBOL Core also defines a Collection. To classify DnaComponents by type additional terms and definitions must use those defined by the Sequence Ontology [99], [134]. For example, a promoter region, coding sequence, and transcriptional terminator are all defined by the Sequence Ontology. Terminology outside of the scope of the Sequence Ontology should be submitted as a new term requests to its curators. For example, many terms needed for DNA construction, such as the BioBrick assembly standards are not available from the Sequence Ontology at this time. As the SBOL ontology is expanded it will provide a richer vocabulary for the description of synthetic biology constructs.

The SBOL Core data model specifies the object and data properties associated with the concepts defined by SBOL Core vocabulary. It represents a consensus of the minimal information needed to describe DNA sequences used in synthetic biological designs. For each SBOL vocabulary term the data model defines a Class. The model then specifies how individual instances of each class should relate to each other and their data elements. For example, DnaComponent is the class or type of object that represents a 'DNA Component'.

The SBOL Core format is a specifically defined serialization of the SBOL Core data model. The syntax provides is the common form of the data for both the senders and receivers, enabling the exchange at the technical level. The SBOL format is a strictly defined XML [135] serialization which is also valid RDF/XML syntax [136]. The XML defined for SBOL format includes characteristics of RDF. I developed this format in collaboration with Dr. Evren Sirin from Clark & Parsia, LLC and proposed it as a solution to the SBOL Developers. Its dual compatibility reconciles concerns SBOL Developers expressed about the complexity of RDF and the provision for the future growth of SBOL.

4.5 SBOL CORE VOCABULARY

The SBOL Vocabulary defines terms used in SBOL. Below we define terms for the Core. These term definitions are written for the synthetic biology software developer so that they may be applied in practical software development. The definitions are not written from a formal ontological perspective, as would be needed to define the terms in unambiguous language outside of the context of the synthetic biology field. Additional terms, such as those related to gene expression and DNA construction, are being considered as extensions in collaboration with the Sequence Ontology project. The SBOL:Core terms are defined to be used as concepts common to descriptions of DNA sequences in synthetic biology. The vocabulary also specifies an official URI for each term. The URI consists of a namespace, http://sbols.org/v1# for version 1 terms, followed by a name fragment. For example, *DnaComponent* is the name fragment of the URI for the term “DNA Component” http://sbols.org/v1#DnaComponent. The shorthand for SBOL vocabulary terms URIs is used in the rest of this document, e.g. *sbol:DnaComponent*.

sbol:DnaComponent A DNA component represents a segment of DNA that serves to abstract the DNA sequence as an individual object, which can then be manipulated, combined, and reused in engineering new biological systems.

sbol:DnaSequence The DNA sequence is a contiguous sequence of nucleotides. The sequence is a fundamental information object for synthetic biology and is needed to reuse components, replicate synthetic biology work, and to assemble new synthetic biological systems. Therefore, both experimental work and theoretical sequence composition research depend heavily on the exact base pair sequence specification associated with *DnaComponents*.

sbol:SequenceAnnotation The sequence annotation is the position and direction of a notable sub-sequence found within the *DnaComponent* being described. Annotations provide the link which describes the DNA sequence of a component in terms of other components, *subComponents*. When a DNA component is abstract, SequenceAnnotations specify relative positions between *subComponents*.

sbol:Collection A collection is an organizational container, a group of *DnaComponents*. For example, a set of restriction enzyme recognition sites, such as the components commonly used for BBF RFC 10 BioBricks™ could be grouped. A *Collection* might contain DNA components used in a specific project, lab, or custom grouping specified by the user.

4.6 SBOL CORE DATA MODEL

This section defines the structure of the SBOL Core model. In Figure 5, the UML class diagram notation is used to describe the Core model classes, their properties, and the main associations between classes. Section 9 provides complete examples encoded in SBOL. There are four classes in the data model, *DnaComponent*, *DnaSequence*, *SequenceAnnotation*, and *Collection,* which correspond to the four concepts needed to unambiguously describe the DNA design of synthetic biological systems. Each instance of a *DnaComponent* class refers to an actual or planned DNA component. When using SBOL to describe information about DNA components, an instance of the *DnaComponent* class MUST be created. The *DnaComponent* instance MAY specify an associated *DnaSequence* instance that it pertains to, and MAY be described using *SequenceAnnotation* instances to specify the position of *subcomponents* (*DnaComponent*). *Collection* instances MAY have associated *DnaComponent* instances. These concepts are illustrated in Figure 4.5.

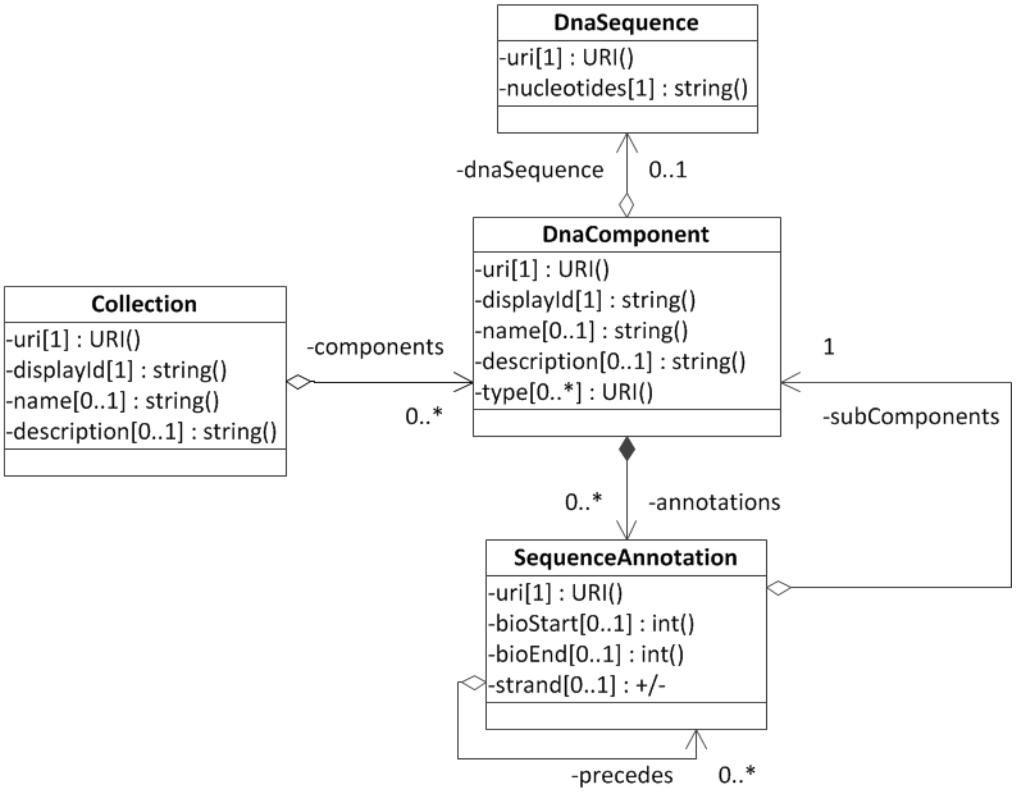


Figure 4.5. SBOL core data model is specified using a UML 2.0 diagram [137]. Classes (rectangles) are named at the top and connected by associations (arrows). Each association is labeled with its role name, and has a range type and a plurality, such as "exactly zero or one dnaSequence*" [0..1] or "one and only one* subComponent*" [1]. These can be interpreted as Sets of objects which are instances of the Class specified. An arrowhead indicates that the association can be traversed in that direction. Diamonds classify the association. Open-faced diamonds are shared aggregation, meaning the object at the end of the arrow can exist independently of the source object, and filled diamonds indicate composite aggregation, or a part-whole relationship, which means that a part instance must be included in at most one whole and cannot exist independently. Data properties for objects of each class are listed in a separate compartment below, with the cardinality and corresponding data types specified.*

Next, I discuss the unique attributes and requirements of each class in the SBOL Core data model.

4.6.1 *DnaComponent Class:*

Instances of the *DnaComponent* class represent segments of DNA as defined by sbol:*DnaComponent*. The *DnaComponent*'s DNA sequence can be annotated using *SequenceAnnotation* instances, positionally defined descriptors of the sequence which specify additional *DnaComponent* instances as *subComponents*. A *DnaComponent* may specify one *DnaSequence* instance it abstracts. *DnaComponent* instances may also be grouped into *Collections*.

DnaComponent instances are required to have *uri* and *displayId* data proprieties. The *uri* property uniquely identifies the instance per the definition of URI [138]. The URI of the instance is intended to be used whenever a reference to the instance is needed, such as when referring to a *DnaComponent* in a *Collection* or *SequenceAnnotation*. The *displayId* is a human readable identifier for display to users. For example, users could use this identifier in combination with the namespace of the source as an unambiguous reference to the DNA construct.

One of the recommended data properties is *name*. The *name* of the DNA component is a human-readable string providing the most recognizable identifier used to refer to this *DnaComponent*. It often confers meaning of what the component is in biological contexts to a human user. A *name* may be ambiguous, in that multiple, distinct *DnaComponent*s may share the same *name*. For example, acronyms are sometimes used (eg. pLac-O1) which may have more than one instantiation in terms of exact DNA sequence composition. As these names are intended for human consumption, they should be kept short and meaningful, by using an acronym, or re-using names that have commonly been used in the literature.

The *description* is another recommended data property. The *description* is a free-text field that contains text such as a title or a longer free-text-based description for users. This text is used to clarify what the *DnaComponent* is to potential users (e.g. engineered Lac promoter, repressible by LacI). The description could be lengthy; therefore, so it is the responsibility of the user application to format and allow for arbitrary length.

The *type* property is a reference to a URI from the Sequence Ontology. These provide a defined terminology of types of *DnaComponents*. For example, a promoter, coding sequence (CDS), and transcriptional terminator are all defined by the Sequence Ontology [99], [134]. Conforming to Sequence Ontology allows SBOL to leverage a significant external community resource and to ensure that SBOL annotated data are also compatible with standards and ongoing developments in the genomics and genetics fields.

In addition to the data properties, the specification recommends that a *DnaComponent* instance have the *dnaSequence* and *annotations*. The *dnaSequence* property specifies the DNA sequence using an instance of *DnaSequence*. The *annotations* properties link a *DnaComponent* to *SequenceAnnotation* instances. *SequenceAnnotations* specify the position and direction of a *DnaComponent* that describes a *subComponent* of this DNA component.

4.6.2 *DnaSequence Class:*

Instances of the *DnaSequence* class contain the actual DNA sequence string. This specifies the sequence of nucleotides that comprise the *DnaComponent* being described. A *uri* property is required in the same form as for *DnaComponent*. The *nucleotides* property is required and strictly defines the criteria for a valid DNA sequence. For example, the base pairs must be represented by a sequence of lowercase characters corresponding to the 5' to 3' order of nucleotides in the DNA segment described, e.g. "actg". The full validation criteria for the string value are explicitly stated in the specification [122].

4.6.3 *SequenceAnnotation Class:*

Individual instances of the *SequenceAnnotation* class provide the position and direction of *subComponents* (i.e. *DnaComponent*s) that are found within the annotated *DnaComponent*. This property specifies the DNA sequence feature being annotated on the *DnaComponent's* sequence. The *DnaComponent* value serves to indicate information about the subsequence at the position specified by the *SequenceAnnotation*'s location data properties or the relative position object property.

Location can be specified by the *bioStart, bioEnd* positions, and *strand* of the *subComponent*, along with the DNA sequence. As a convention, numerical coordinates in this class use position 1 (not 0) to indicate the initial base pair of a DNA sequence. This convention is followed by the broader Molecular Biology community, especially in the relevant literature.

Furthermore, the *DnaSequence* value of the *subComponent* ie the exact sequence found in the interval specified by the Location Data. The strand orientation, or direction, of the *subComponent*'s sequence relative to the parent *DnaComponent* is specified by the *strand* [+/-]. For *strand*: '+' the sequence of the *subComponent* is the exact sub-sequence, and for '-' it is the reverse-complement of the parent *DnaComponent*'s sequence in that region.

Alternatively, relative positions of *subComponent*s can indicate the order of *subComponents* when there is not enough information to specify exact positions. Relative positions specified by indicating the *precedes* relationship to other *SequenceAnnotations*. *Precedes* indicates the intended location by specifying that a *SequenceAnnotation* is to come before another when *DnaSequence* information becomes available. For example, you may want to say the promoter *SequenceAnnotation precedes* the CDS *SequenceAnnotation,* which *precedes* the terminator *SequenceAnnotation.* This ordering gives us the position, relative to other *SequenceAnnotations* (which can have a location or a relative position (using *precedes*)). In the case of a *DnaComponent* with a mix of locations and relative positions in its *SequenceAnnotations* the specification defines strict criteria for logical consistency, preventing nonsensical combinations being created. For example, during a validation process, the set of *precedes* relations on the *SequenceAnnotation* are required to be linearized to a sequence.

4.6.4 *Collection Class:*

Individual instances of the *Collection* class represent an organizational container which helps users and developers conceptualize a set of *DnaComponents* as a group. For example, a set of restriction enzyme recognition sites, such as the components commonly used for BBF RFC 10 BioBricks™, could be placed into a single *Collection*. A *Collection* might contain DNA components used in a specific project, lab, or custom grouping specified by the user. Any combination of *DnaComponents* can be added to a *Collection* instance, annotated with a *displayID*, *name*, and *description* and be published on the web or transferred directly.

The *components* property specifies the *DnaComponents* which are members of this *Collection* and represent DNA segments for engineering biological systems. For example, standard biological parts, BioBricks, pBAD, B0015, BioBrick Scar, Insertion Element, or any other DNA segment of interest as a building block of biological systems. *Collection* can have a *name*, which is a human readable and recognizable identifier. The *name* should confer what is contained in the *Collection*. It may often be ambiguous (e.g. Mike's Arabidopsis Project A; Parts from Sleight, et al. (2010) J.Bioeng; BBF RFC 10 DNA Components; or, My Bookmarked Parts). The *description* property is a free-text field, which should contain human-readable text describing the *Collection* for users to interpret what this *Collection* 'is'. This text should focus on an informative statement about the reason for grouping the *Collection* members. The *description* should allow users to interpret the reason for inclusion of members in this *Collection* (eg

"Collecting parts which could be used to build honey production directly into mouse-ear cress"; "T9002 and I7101 variants from Sleight 2010, designs aim to improve stability over evolutionary time"; "Components useful when working with BBF RFC 10").

However, arbitrary groupings and new *Collection* instances should not be created and named when the groupings are not defined or whenever an arbitrary set is possible. *Collections* should only be used to represent a grouping that is useful to a user.

4.7 EXAMPLES

Sharing information about a variety of DnaComponents using the SBOL allows unambiguous specification of their DNA sequence-based descriptions. This section presents examples illustrative of different SBOL use cases.

4.7.1 *Annotated Composite DnaComponent*

The first example is the SBOL Core model for the BioBrick™ BBa_I0462. The BBa_I0462 *DnaComponent* codes for the LuxR protein when inserted downstream of a promoter (Figure 6). Information comes from the Registry of Standard Biological Parts [57] to describe this canonical composite BioBrick™ part.

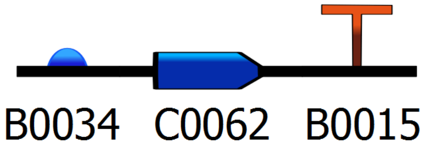


Figure 4.6. Simple DNA design, BBa_I0462 [57] drawn using SBOL Visual symbols in TinkerCell [64] and composed of BBa_B0034, BBa_C0062, BBa_B0015 DnaComponents. The icons are labeled with a shorthand notation of the displayId *from the Parts Registry [57].*

In Figure 7a the BioBrick™ part BBa_I0462, a *DnaComponent*, is depicted with annotations of three *DnaComponent*s: a ribosome binding site (BBa_B0034), the coding sequence for LuxR (BBa_C0062), and a double terminator BBa_B0015. In Figure 7b, the same *DnaComponent* is described using pseudocode as an example.

Instances of SBOL Core model classes are written as abbreviations.

Abbreviation key:

$DC_{Ø}$ – a *DnaComponent* w/o type, w/o sequence

DCt – a *DnaComponent* w/ type

DCs – a *DnaComponent* w/ sequence

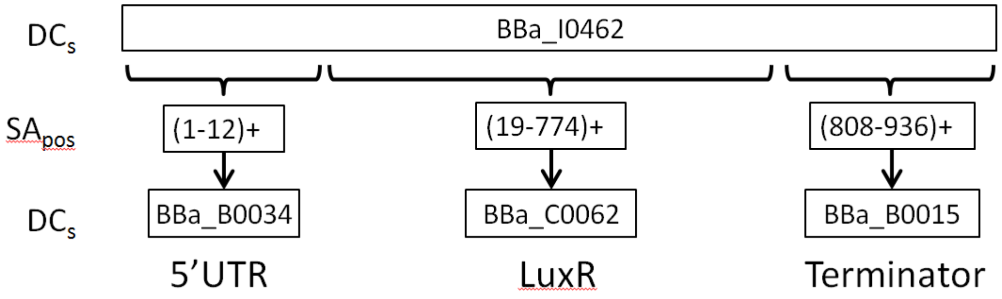
DCst – a *DnaComponent* w/ sequence and type

$SApos_N$ – a *SequenceAnnotation* w/ position coordinates [N-ordinal notation only]

$SArp_N$ – a *SequenceAnnotation* w/ relative position (precedes)

$SArp_{Ø}$ – a *SequenceAnnotation* w/ relative position (terminal SA)

Col – a *Collection*



(a)

```
DnaComponent [
  uri: http://partsregistry.org/Part:BBa_I0462
  displayId: BBa_I0462
  name: I0462
  description: LuxR protein generator
  annotations:[

    SequenceAnnotation [
       uri: http://sbols.org/anot#1234567
       bioStart: 1
       bioEnd: 12
       subComponent:[

          DnaComponent [
             uri: http://partsregistry.org/Part:BBa_B0034
             displayId: BBa_B0034
             name: B0034
             type: ribosome_entry_site
          ]
       ]
    ]

    SequenceAnnotation [
       uri: http://sbols.org/anot#2345678
       bioStart: 19
       bioEnd: 774
       subComponent:[

         DnaComponent [
            uri: http://partsregistry.org/Part:BBa_C0062
            displayId: BBa_C0062
            name: luxR
            type: CDS
         ]
       ]
    ]

    SequenceAnnotation [
       uri: http://sbols.org/data#3456789
       bioStart: 808
       bioEnd: 936
       subComponent:[

         DnaComponent [
           uri: http://partsregistry.org/Part:BBa_B0015
           displayId: BBa_B0015
           name: B0015
           type: terminator
         ]
       ]
    ]
  ]

  DnaSequence [
    uri: http://sbols.org/seq#d23749adb3a7e0e2f09168cb7267a6113b238973
    nucleotides:
aaagaggagaaatactagatgaaaaacataaatgccgacgacacatacagaataattaataaaattaaagcttgtagaagcaataa
tgatattaatcaatgcttatctgatatgactaaaatggtacattgtgaatattatttactcgcgatcatttatcctcattctatgg
ttaaatctgatatttcaatcctagataattaccctaaaaaatggaggcaatattatgatgacgctaatttaataaaatatgatcct
atagtagattattctaactccaatcattcaccaattaattggaatatatttgaaaacaatgctgtaaataaaaaatctccaaatgt
aattaaagaagcgaaaacatcaggtcttatcactgggtttagtttccctattcatacggctaacaatggcttcggaatgcttagtt
ttgcacattcagaaaaagacaactatatagatagtttatttttacatgcgtgtatgaacataccattaattgttccttctctagtt
gataattatcgaaaaataaatatagcaataataaatcaaacaacgatttaaccaaaagagaaaaagaatgtttagcgtgggcatg
cgaaggaaaaagctcttgggatatttcaaaaatattaggttgcagtgagcgtactgtcactttccatttaaccaatgcgcaaatga
aactcaatacaacaaaccgctgccaaagtatttctaaagcaattttaacaggagcaattgattgcccatactttaaaaattaataa
cactgatagtgctagtgtagatcactactagagccaggcatcaaataaaacgaaaggctcagtcgaaagactgggcctttcgtttt
atctgttgtttgtcggtgaacgctctctactagagtcacactggctcaccttcgggtgggcctttctgcgtttata
  ]
]
```

(b)

Figure 4.7. Annotated Composite DnaComponent ***(a)*** *A diagram of the SBOL instances used to describe* `BBa_I0462`*. The gaps shown between the sequence annotations are unannotated segments of DNA.* ***(b)*** *Pseudocode is used to demonstrate the use of core data model structure and data fields in a complete example of a DnaComponent.*

4.7.2 *Multi-Tiered Annotated DnaComponent*

The next example depicts the subcomposition of BBa_I0462 in terms of each of its *subComponents* (Figure 8).

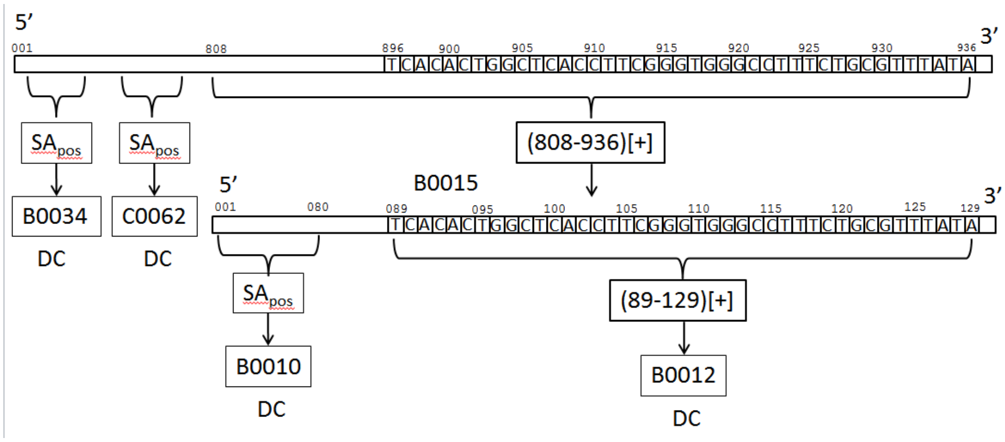


Figure 4.8. Expanded instance of BBa_I0462, which demonstrates key features of SBOL:Core:model. In this instance, the BBa_B0015 component of BBa_I0462 from the examples above is composed of two elements itself, BBa_B0010 and BBa_B0012. The letters of the DNA sequence in the two top DnaComponents *is omitted, so only the sequence corresponding to BBa_B0012 is shown.*

4.7.3 *Partially Realized Design Template*

This example illustrates the partial specification of designs in terms of *DnaComponent* layout constraints. Figure 9 demonstrates the use of *SequenceAnnotations* with a Relative Position to specify the order of *DnaComponents* within a planned *DnaComponent*.

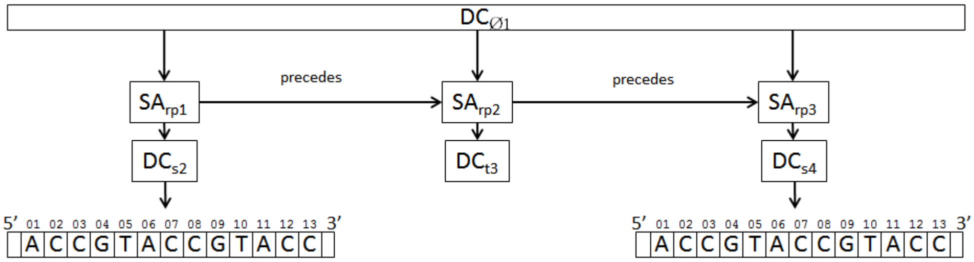


Figure 4.9. Design template for DnaComponent *$DC_{Ø1}$ specifies that at least three DnaComponents must be present in this design. Their ordering is constrained, DC_{s2} precedes DC_{t3} and DC_{t3} precedes DC_{s4}. In this template the DC_{s2} and DC_{s4} already have a DnaSequence specified, however DC_{t3} does not, instead it specifies a type which it must be constrained to. Therefore, the DC_{t3} component can be filled in to match the type constraint later.*

4.7.4 *Collection*

The Collection class provides an organizational container for multiple *DnaComponent* instances. The example in Figure 10 shows a Collection with multiple DnaComponents grouped together and ready to be shared between software applications.

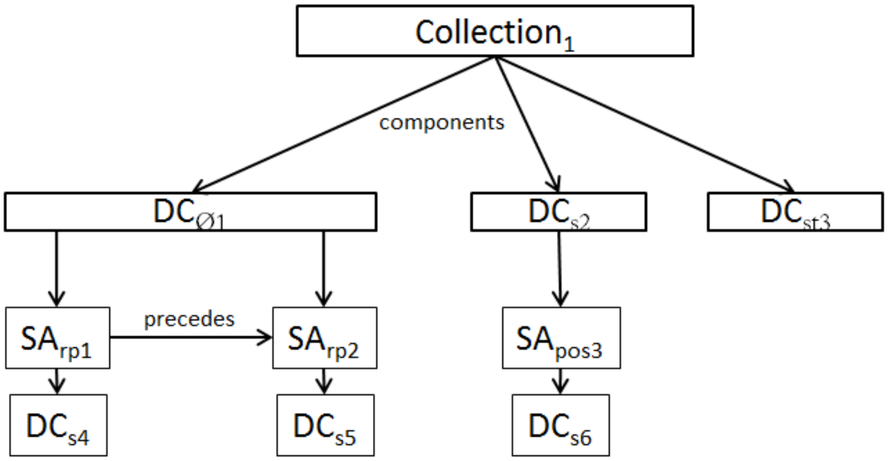


Figure 4.10. Collection$_1$ is a convenience object to group DnaComponents *$DC_{Ø1}$, DC_{s2}, and DC_{st3}. Described collections are a natural conceptualization of a group of objects to be shared at one time or that serve a specific purpose.*

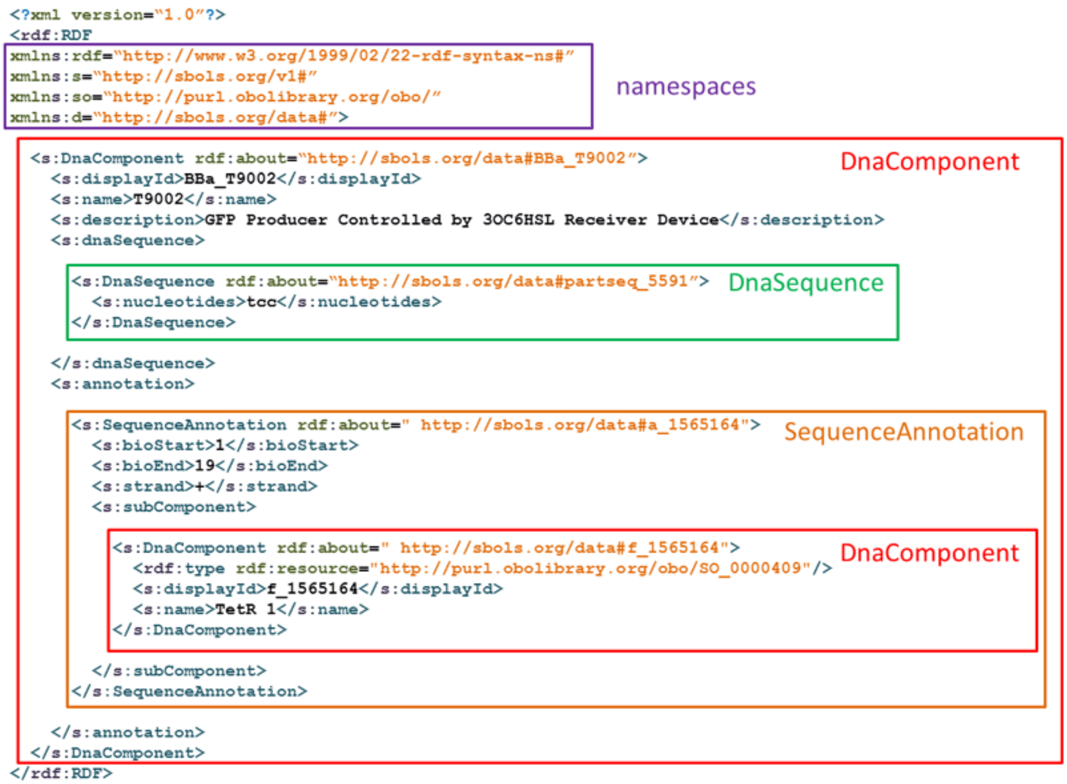
4.8 Serialization

The SBOL file format is used to express instances of the Core data model for storage and transmission. It ensures that SBOL data is read consistently by SBOL software, for example using libSBOLj. To confirm that data in SBOL format can be read by another SBOL tool it can be validated to certify consistency. The format is a strictly defined subset of the RDF/XML syntax. An XML schema (XSD), developed by Dr. Evren Sirin including contributions from me, specifies these constraints to define a valid SBOL document. The schema and SBOL document validation is described below in Section 4.8.1. However, as SBOL files are also valid RDF documents, SBOL can be read by any RDF tool and interpreted as a graph. The implications of interpreting SBOL as RDF are described in more detail in Chapter 6.

We defined a subset of RDF/XML as to not sacrifice the ease of use of plain XML. This is especially important for developers planning to parse SBOL format using a typical XML parser [139]. We made this choice to leverage the main advantage of XML; developer's familiarity with the technology. Use of XML ensures that the SBOL format is familiar to most software developers because XML is a very popular syntax for data transmission. In Figure 11 I show an example of a simple SBOL document to illustrate the structure and the recognizable XML look of the SBOL format. The document is declared as an XML document and is a valid RDF document, enclosed by RDF tags. It begins with a namespaces section which defines namespaces used.

Namespaces are containers that provide context for SBOL identifiers (i.e. URIs). Namespaces are used to create a unique context for the identifiers. There are two types of namespaces in an SBOL document. The language namespaces (e.g. SBOL namespace "http://sbols.org/v1#"), which are always the same and the data namespaces, which vary for each data source (e.g. parts registry namespace "http://partsregistry.org/part/").

The rest of the document is composed of the SBOL elements found in the file. In Figure 11 a typical *DnaComponent* is defined with *displayId*, *name*, *description*, and a *DnaSequence*. Optional *SequenceAnnotations* are next, including an example of a nested *DnaComponent* definition for the *subComponent*. This example was generated by the Parts Registry to SBOL converter. For a description please see Chapter 5.



```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:s="http://sbols.org/v1#"
xmlns:so="http://purl.obolibrary.org/obo/"
xmlns:d="http://sbols.org/data#">
  <s:DnaComponent rdf:about="http://sbols.org/data#BBa_T9002">
    <s:displayId>BBa_T9002</s:displayId>
    <s:name>T9002</s:name>
    <s:description>GFP Producer Controlled by 3OC6HSL Receiver Device</s:description>
    <s:dnaSequence>

      <s:DnaSequence rdf:about="http://sbols.org/data#partseq_5591">
        <s:nucleotides>tcc</s:nucleotides>
      </s:DnaSequence>

    </s:dnaSequence>
    <s:annotation>

      <s:SequenceAnnotation rdf:about=" http://sbols.org/data#a_1565164">
        <s:bioStart>1</s:bioStart>
        <s:bioEnd>19</s:bioEnd>
        <s:strand>+</s:strand>
        <s:subComponent>

          <s:DnaComponent rdf:about=" http://sbols.org/data#f_1565164">
            <rdf:type rdf:resource="http://purl.obolibrary.org/obo/SO_0000409"/>
            <s:displayId>f_1565164</s:displayId>
            <s:name>TetR 1</s:name>
          </s:DnaComponent>

        </s:subComponent>
      </s:SequenceAnnotation>

    </s:annotation>
  </s:DnaComponent>
</rdf:RDF>
```

Figure 4.11. SBOL document serialized as a subset of RDF/XML syntax, with markup highlighting sections which correspond to the SBOL Core model.

4.8.1 *SBOL XML Schema and Validation*

The XML serialization for SBOL format it is defined by an XML schema. It provides the constraints on the structure and the types of XML elements which are valid RDF and define valid SBOL. This strategy allows the SBOL Developers to just use XML, but keep the flexibility of RDF when advantageous. Tools which directly read SBOL at the XML level depend on the predicable order and structure defined by the SBOL schema. It is composed of two files, one which defines the SBOL subset of well-formedness constraints of RDF/XML (rdf.xsd), and the other defines the structure based on the SBOL Core data model (sbol.xsd). See Section 4.9 for availability. The structure is defined in terms of the order and nesting of allowed elements. The types are defined to those allowed by the Core specification. For example, we decided to use typed elements (i.e. <sbol:DnaComponent>) and to use referencing and nesting in a consistent way.

The schema can be used to validate SBOL documents using an XML Schema processor. Validation confirms that an SBOL document is *well-formed* and also *valid* in that it follows the defined structure. This functionality is provided by libSBOLj or any XML Schema processor. XML Schema processors are a part of most XML aware tools, such as NetBeans or Eclipse. For example, xmllint in the libxml2 library is a commonly used tool on Linux systems.

4.9 Availability

The libSBOL software libraries provide an API to read and write SBOL documents. Software developers can use these libraries to add import and export of SBOL files to their software. The libraries support the serialization and deserialization of core data model objects into SBOL format.

- libSBOLj – the Java version and reference implementation
- libSBOLc – a C language implementation
- libSBOLpy – provides Python API to libSBOLc

The current version of libSBOLj was developed by Dr. Evren Sirin, with contributions from Nicholas Roehner and Dr. Matthew Pocock. The preliminary implementation of libSBOLc and libSBOLpy were Developed by Jeffrey Johnson. My contribution to these projects, the development process, and how it influenced SBOL is described in Chapter 5. The libraries are maintained by the SBOL Developers and are under continuous development. The sources are licensed using Apache License, Version 2.0, a free open source software license, and they are made available at the SynBioDex site on Github (http://github.com/synbiodex).

The SBOL project website (http://sbolstandard.org) serves as the main portal to the project. It contains documentation, contact information, and links to current locations of any software libraries and the latest updates on the project. The sbolstandard.org materials are licensed under a Creative Commons Attribution 3.0 Unported License, granting the information free to anyone who wishes to use it on condition that they provide attribution.

4.10 Summary

In this chapter I described the technical specification for a community standard as solution to enable data exchange among synthetic biology software. The specification is composed of a vocabulary, data model, and serialization format. The vocabulary defines the concepts to provide a shared understanding among developers. The data model defines the structure of the concepts by specifying the relationships among them. The serialization format is the textual format of the digital objects for storage and transmission of the data. Finally, the software libraries provide the utilities to software developers so they can add the ability to import and export the data. I discuss the capabilities and demonstration of use of the SBOL Core representation in Chapter 5. The adoption of RDF technology is not just a solution to some immediate practical challenges, but most importantly it provides support for sustained development in the context of a growing and evolving community. The implications of this choice as discussed in Chapter 6.

Chapter 5. DATA AND KNOWLEDGE EXCHANGE

Use of a standard demonstrates acceptance and recognition of added value of the technology by the community. The first steps towards use are critical to its proliferation and widespread adoption. The technology adoption model suggests that the innovators need to make the case to early adopters. The case for adoption is built on assessment and demonstration of the capabilities of the standard. The assessment of capabilities helps potential adopters judge whether the standard offers additional value. Demonstrations of use provide convincing evidence of functioning infrastructure for potential adopters. Importantly, independent implementations of an exchange standard demonstrate operational sufficiency, portability, and most importantly buy-in from the participants. Enough technology and documentation must exist for an independent party to implement the standard to show operational sufficiency. Therefore, the theoretical capabilities are technically possible to achieve with an acceptable cost to the implementer. Independent implementations also generate community experience in porting the standard from one environment to another. The subsequent party that implements the standards gains the benefit of the experience of the previous implementer in terms of portability of the technology. Finally, evidence of buy-in gives reassurance to participants that there is community support. The return on the investment for the potential adopter is derived from the coordination of work the standard provides with that community. The adoption of SBOL by the community of synthetic biology software developers described in Chapter 3 is contributing to its success.

The evidence for success is the growing adoption of SBOL in the synthetic biology software community. I substantiate this claim by the demonstrations of its use by independent researchers in their software tools. With the support of the SBOL Developers group, I put SBOL into use as a language for data exchange. As a group we were able to show SBOL used as an interchange format. Additionally, I was able to demonstrate information retrieval using SBOL and Semantic Web technology. SBOL can be used to share and retrieve data and then re-use it across application, enterprise, and community boundaries.

In this chapter I present the use of SBOL as a common representation applied to multiple scenarios. I begin the chapter with a description of its capability to represent and to transmit template designs, annotated DNA sequence, and collections of DNA components (Section 5.1). Then, I explain the iterative development process I used with my collaborators to implement SBOL software libraries for Java, C, Python, and using XSLT (Section 5.2). Finally, I discuss the use of SBOL for peer to peer exchange, publisher to consumer distribution (Section 5.3), as well as to aid information retrieval (Section 5.4). In this chapter I describe in more detail how three synthetic biology data providers, the BIOFAB, Parts Registry, and the BacilloBricks registry at Newcastle University deliver designs in SBOL format. As an example of use I also describe three software tools, iBioSim, GeneDesigner, and Clotho which work with SBOL. I chose these tools to demonstrate SBOL capabilities. In Chapter 6 I provide a complete list of software tools which support SBOL. I begin with the capabilities which motivated me to develop SBOL and for my collaborators to adopt it.

5.1 Capabilities

For synthetic biologists to create biological systems a method for the communication of designs, components, and their descriptions is needed. The ability to electronically communicate designs throughout the engineering process is paramount. For synthetic biologists this entails the management of DNA sequences and their abstractions using software tools which unambiguously interpret the designs. Manually transferring a design from one software application to another is time consuming and error prone. This cost is particularly high when a researcher is assessing many software tools to aid their workflow. Both the synthetic biologist and the developers of software to aid them will benefit from electronic exchange and unambiguous interpretation of design files. The focus of the SBOL Core data model, described in Chapter 4, is the representation of designs composed of DNA components. The representation provided by the core model is designed to be used in several different cases.

In the exchange of designs among researchers there are different roles which each participant can play. Two common models for exchange are the transmission from peer to peer and the transmission from publisher to consumer. In the peer to peer exchange, a synthetic biologist may be interested in sending a design to a collaborator or within their own workflow from one software application to another application. The common solution for both scenarios is to require a shared representation between the peers. The sender transforms their data to the shared representation and the receiver interprets it transforming the data into their own representation. The SBOL core data model serves as this common representation. Furthermore, the flexibility of the SBOL core model enables several different arrangements of information to be represented.

Various levels of detail can be sent using SBOL, depending on the step within the synthetic biology workflow. For example, at the beginning of the design phase a rough sketch may be available, but the DNA sequence is not yet selected. These designs serve as *DNA design templates* (Section 5.1.1), their exact sequence will be added at a later step. The consequence of absent DNA sequence is that other information, such as the position of the subcomponent annotations is still missing. The next step in the design of a synthetic DNA construct is the addition of a DNA sequence. Once the sequence is added the sub-components from the template serve to *annotate the DNA sequence* (Section 5.1.2). The third case is the publication of a *collection of DNA components* (Section 5.1.3) for use by other researchers as subcomponents in their own designs. These capabilities of the SBOL standard fulfill the baseline requirements for synthetic biology software to interoperate. Below I describe each capability, motivated by a scenario based on the design and engineering of evolutionarily robust genetic circuits by Sleight et al. [140].

5.1.1 *Send DNA Design Template*

Synthetic biologists should be able to communicate criteria for designs to colleagues before selecting a specific DNA sequence. SBOL provides the capability to send a *design template*, a representation of a design independent of its sequence. The need for design templates in biological engineering can be illustrated by a scenario in which, a synthetic biologist aims to improve evolutionary stability of BBa_T9002 (example drawn from [140]). T9002 is a genetic device which produces GFP on induction with an AHL input [79], but this functional trait is lost within 20 generations [140]. In order to improve upon the original design, Sleight, et al. needed to alter the composition of the DNA sequence while maintaining the overall function of the device. Therefore, they must specify an abstraction of the design in terms of the DNA components to use, but at this stage the exact sequence is not significant.

The most common method to represent a genetic design is as a diagram. The diagram depicts the genetic level layout, as an ordered set of template DNA components along one strand of DNA (Figure 5.1). The diagram is the most intuitive methods to describe the design template. Such a design template may include the identifiers for the constituent DNA components and have their types specified, but not their DNA sequence. Diagrams, similar to this one can be found in synthetic biology primary literature, or can be created in synthetic biology design software application, such as TinkerCell [64]. These diagrams represent an abstraction of the structure of the DNA sequence as segments of DNA. The representation specifies requirements for the design, one part must be a promoter, another a GFP coding sequence, but it does not include the DNA sequence. It is possible that many different DNA base pair sequences will fulfill the requirements. SBOL Visual was created to represent specifically this kind of diagram. SBOL Visual symbols represent a type of DNA component. For example, the design shown in Figure 5.1 specifies a linear layout of the circuit along a single strand of DNA. The design includes the specification of types of DNA components with names and simplified partsregistry.org IDs, as well as component names where appropriate [141]. The example shown in Figure 5.1contains no additional information in "hidden" form (eg DNA sequence, model parameters).

However, some interpretation of the diagram is needed, such as the order and type of components. A generalization of this use case is to not specify IDs and instead to specify "other" required parameters for the design specification. The new design (Figure 5.1) calls for the use of DnaComponents in a specified order: R0040 (pTetR), B0034, C0062(luxR), B0010, B0012, R0062 (pLuxR), B0032, E0040 (gfp), and J61048 (replacing the B0015 of the original design). After creating this alternate design the researcher can delegate the work to physically build it.

A diagram like the one in Figure 5.1 does not contain the sequence. The next task to fulfill the design is to add the sequence information. Only then one can plan the DNA assembly or synthesis process. While it is possible to use one integrated software tool to do both tasks, a more powerful approach is to design once, and enable the researcher to choose which tool to use in subsequent tasks. Currently, Clotho Apps, Eugene, MatchMaker, DeviceEditor, and GenoCAD offer computational utilities to realize a design based on a *design template*. Alternatively, a DNA sequence editor such as ApE, VectorNTI, or Geneious, can be used to manually compile the sequence.

Saving the design in an SBOL file will allow the design to be completed in any software which provides the automated or semi-automated solutions to choose a specific sequence. Most importantly the ability to test multiple tools in an *ad hoc* fashion in order to choose the most suitable downstream tool is a capability only SBOL, a common standard, can provide.

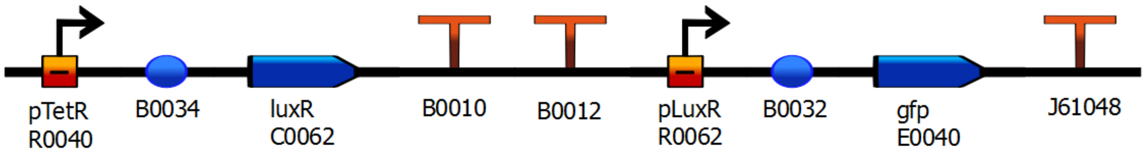


Figure 5.1 The T9002_J61048 design from TinkerCell.

Encoding the information in a design template into an SBOL data file allows the information to be manipulated by the software tool which read it. For example, the DNA sequence for each segment or the entire design can be filled in. Unlike a diagram image, the main advantage of encoding the design template represented in the diagram as an SBOL formatted file, is the ability for it to serve as input for other software which can help complete the DNA sequence.

5.1.2 *Send Annotated DNA Sequence*

Synthetic biologists should have the ability to share their designs with colleagues, publish them with journal articles, or simply be able to refer to them in future projects. A complete design has a fully specified sequence and annotations which specify each of its sub-components. The complete design may be sent out for assembly or synthesis as DNA in physical form. The same design object can then be used to describe the realized construct, a DNA segment which can be reused as a component of higher order devices. Synthetic biologists should be able to send such rich descriptions of DNA sequences, which define a DNA component, and expect that the recipient will be able to read it. The *annotated DNA sequence* in SBOL is a representation of a DNA segment described in terms of position specific sub-components and standardized types defined by terms from the Sequence Ontology [99]. This SBOL case is analogous to the familiar GenBank flatfile format [142] with some modifications.

In the practice of synthetic biology, specific composition of DNA components, in terms of the sequence of DNA bases, is of utmost importance. The representation of this level of information is the focus of the SBOL core model. The annotation of sequence, designation of noteworthy regions of DNA segments, is crucial in planning laboratory tasks, interpreting verification results, and understanding how it was built. Sequence annotations are also necessary to denote functional regions of a design.

Continuing with the example scenario described in Section 5.2.1, Sleight, at al. delegate the task of completing, or realizing, and building the T9002_J61048 design in Figure 5.1 to another team member. Realization of the design requires the sequence to be filled in using Eugene [71]. The result is a compilation of the DNA components fulfilling the design that was previously specified. To do this, the Eugene tool needed a collection of DNA components from the partsregistry.org translated into Eugene Part files (see section 5.3). Annotated DNA sequences, such as the output of Eugene, can also be created within DNA editing software or specialized design realization tools such as, MatchMaker, DeviceEditor, or GenoCAD. The annotated DNA sequence is a completed design which must be either assembled or synthesized. Before the physical DNA can be obtained, the steps needed to, assemble or synthesize and then verify the sequence DNA post-assembly must be planned. Therefore, output from the design phase should serve as direct input to software tools which offer assembly optimization strategies, such as j5 [143] and GenoCAD [129]. Alternatively, the complete design could be sent to DNA synthesis service providers such as DNA 2.0, GENEART, or IDT which synthesize DNA *de novo*.

While the scenario described here involved tools not designed to specifically work in concert as an integrated workflow, the analogous process could be performed in a specialized tool package. The TASBE tool chain also supports the fulfillment of a BioCompiler created design in MatchMaker. Furthermore, iBioSim, biological CAD tool, could generate the complete design and feed into any of these downstream tools. GenoCAD could be used to make the Design Template or to read one generated elsewhere and when given the collection of components. GenoCAD automates the process of assigning sequence to the design. Further options include using Spectacles or Eugene Scripter to read the output directly into Clotho, and use its assembly algorithms to plan the assembly. These alternate examples allow for the transmission of designs, but only between tools developed by the same group to specifically work in jointly. Without the community agreed to common SBOL format for the annotated DNA sequence, the option to form *ad hoc* connections would not be available to the synthetic biologist.

5.1.3 *Publish Collection of DNA Components*

An important expansion of the data exchange scenarios described in Sections 5.2 and 5.3 is the publication of large number of DNA components by an organization operating a repository which serves many potential synthetic biology users. For example, a dedicated bio-fabrication facility, such as the BIOFAB, may want to distribute a collection of standardized DNA components which can be re-used as sub-components for the design of novel DNA circuits. Recipient software tools, such as Clotho, should be able to read these collections and use them in designs. For example, they could be used as part files for assembly in Eugene.

To help the publishers distribute data, SBOL provides the option to represent a set of DNA components as one or more collections. SBOL collections are a representation of a set of DNA components. For example, the BIOFAB has released three collections: the Modular Promoter Library, the Random Promoter Library, and the Terminator Library. These customized groupings allow the description to specify both the type of the component, but also how the promoter sequence was generated, as modules or randomly. SBOL is extensible and therefore an alternative approach for the BIOFAB would be to create an extended classification scheme which includes a vocabulary for how the sequence was generated. Such an approach would be equivalent to extending the SBOL model into a more complete and verbose ontology. However, the immediate practicality of this approach places an increased burden on the developer implementing SBOL compliance. Such vocabulary extensions require additions to the SBOL structures and require more sophisticated underlying software. Such increase in the complexity of the SBOL model would have been inhibitive to its early adoption and deployment. Therefore, SBOL Developers agreed to create the Collection class as a placeholder solution for an *ad hoc* grouping capability.

This scenario is an extension the common publisher-consumer model for distribution of data from an authoritative source, a publisher, to many targets, the consumers. In this model the publisher’s goal is to make data available to as large a number of consumers as possible. In this model the publisher decides how and in what form to distribute their data. For example, the BIOFAB, PartsRegistry, and JBEI-ICE each have their own distribution formats which include unique information specific to the organization. However, SBOL provides a common framework therefore the consumer can create a single interface and receive data from all three sources in the example.

The importance of the standardized description becomes more essential when the recipient would like to filter for components which match design criteria. For example, in SBOL we use the Sequence Ontology types used to classify DNA Components which can be used to select components by type. For example, the Sequence Ontology terms described above could be used to retrieve DNA components by type across all three resources. I hope publishers will take advantage of SBOL's extensibility, as adoption of SBOL grows, to add more complete descriptions to better help recipients interpret published collections.

5.2 Deployment

I approached the SBOL development process based on an iterative strategy. I deployed early versions, followed by gathering requirements in the form of feedback, and then revised based on the feedback to the design in repeated stages. Using this cycle of deployment and feedback (Figure 5.2) I refined both the software and the SBOL model. Using the iterative development strategy I was able to improve the capabilities SBOL offered to better meet the needs of the SBOL Developers group. Opening the development process to feedback and contribution from collaborators improved buy-in from the community. Buy-in from a diverse group promoted its acceptance as a valuable solution to data exchange.

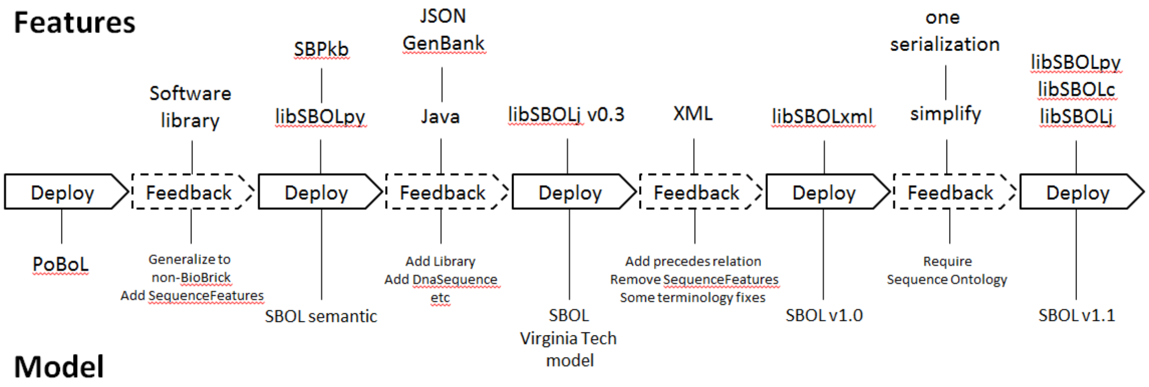


Figure 5.2. SBOL development consisted of alternating deployments and gathering feedback. The iterations are depicted as a timeline of both the delivered software features (top) and the SBOL core model evolution (bottom).

The iterative process began when I released the initial core data model representation, named PoBoL, which later would be renamed to SBOL by the collaborative group. This early version of the standard was implemented using OWL/RDF [117]; however it did not include a software library to support implementation. From the feedback I received it was clear, a basic library which would implement SBOL import and export was needed. Such a library would help developers adopt the standard within their software by providing the serialization and de-serialization methods. Additionally, the core model needed to be more general. PoBoL was conceived specifically to represent BioBrick Standardized Biological Parts, but SBOL should aim to support any DNA sequence used in the synthetic biology context. Furthermore, a demonstration of the data exchange proof of principle using SBOL in a simple implementation would help demonstrate its purpose.

For the second iteration, I addressed these concerns by implementing SBOL-semantic, a more general information model for synthetic biology, using the Web Ontology Language (OWL). This new work built on the Provisional BioBrick Language (PoBoL) [144]. I built SBOL-semantic using OWL/RDF so as to be compliant with Semantic Web information technology standards that allow SBOL data records to be read, manipulated, and interpreted using generic tools such as Protégé [145], RDFlib [84] and Sesame [85]. To provide a utility library specific to SBOL I implemented the libSBOL library using the Python programing language. I then used this library to enable data exchange using SBOL within other applications. These tools were used for management of SBOL model structure, to create a scheme for unique identification of elements, and to reference the Sequence Ontology [146], a third party ontology. The choice of W3C recommended technology was made on the premise that modeling knowledge in a computable, standardized, and community supported format will provide long term benefit for the synthetic biology community (See also Chapter 6). I then demonstrated the use of SBOL by creating the Standardized Biological Part knowledgebase [147], which is described in detail in Section 5.4.

I presented this proof of concept implementation at the 2010 International Workshop for BioDesign Automation in Anaheim, CA. The initial reaction of the participants was clear interest in the project and an invitation to collaborate. Following the SBOL workshop, I approached potential collaborators who would help testing by deploying the technology at their sites. In the context of the collaborative group we formed, I continued to receive feedback throughout the rest of project.

My point of contact for the collaboration with the BIOFAB was Dr. Cesar Rodriguez. In order to address the need for a library which would implement SBOL data exchange in the BIOFAB's Java software, I needed to apply the changes to the model and then re-implement using Java. In October 2010 I sent a proposal to the BIOFAB to develop a new library in Java and made the initial version of the software available to Dr. Rodriguez in January 2011.

The libSBOLj library was able to serialize and de-serialize SBOL as RDF/XML and serialize to Javascript Object Notation (JSON) files and the GenBank flat file format.

libSBOLj Features

- Create and Access SBOL objects
- Read and Write SBOL RDF
- Write SBOL JSON
- Read GenBank flat file format

In the deployment I pre-packaged the JAR file including any third party libraries it depends upon. Also, I included detailed documentation of the classes and methods including a tutorial, examples of use, and the source code with the release. To facilitate the feedback process I made the library available on the GitHub source code repository where *issues* could be tracked as revisions are made. Dr. Rodriguez provided an initial round of feedback. I then released libSBOLj v0.3 to the entire SBOL developers group in an effort to identify additional deployment sites.

Professor Chris Myers, University of Utah, and Dr. Timothy Ham, JBEI, performed a detailed code review and Nicolas Roehner, University of Utah, and Dr. Rodriguez offered follow-up comments after having used the library within their respective applications.

Feedback

- Remove the GenBank flat file parser/ BioJava
- Use native Java functions to replace GNU-crypto package
- Throw exceptions to replace logging
- Package in one-jar
- Follow Java coding conventions strictly
- Proposed changes to the specification

The common thread throughout this round of feedback was to simplify the deployment package.

At this stage the SBOL Developers, not only offered feedback but began to volunteer their time to implement the support software library. For example, Allan Kuchinsky offered help from Agilent Technologies to implement the next version of the libSBOL library. Trevor Smith from Agilent Technologies implemented libSBOLxml, a simplified version of the library in Java, which reflected the feedback on the prior versions. His implementation followed the SBOL v1.0

model. He was able to release the library to the SBOL Developers soon after I submitted the specification document [120] to the BioBrick Foundation Request For Comments repository. The Boston University, Newcastle University, and University of Utah teams used this library within their own software. Their demonstration of its use at the SBOL Workshop in Seattle is described in Section 5.3.2. Following this successful demonstration and motivated further to simplify the software support for SBOL the Developers agreed to a single serialization format.

Over the next few months additional members of the SBOL Developers groups helped again. Dr. Evren Sirin from Clark and Parsia, LLC., re-implemented libSBOLj to support the new SBOL v1.1 format and created an XML Schema for the SBOL serialization. He included a basic validator tool with the libSBOLj library to help developers check their SBOL format. Furthermore, I worked with Jeffrey Johnson from Professor Herbert Sauro's group to implement libSBOL using the C language, libSBOLc, and to create libSBOLpy, a Python language wrapper of the C version.

Following this successful test of the library the second company Clark & Parsia, LLC., a small software company focused on semantic technologies, offered to re-write the library to support not only the updated data model in XML but to also make the XML compatible with RDF. Dr. Evren Sirin, from Clark & Parsia, re-wrote the library to create a single serialization which both meets the requirements for use as standard XML and allows this same serialization to be interpreted by generic RDF tools. It is through this agile and iterative development process of the technology, both software and the data model, that we were able to demonstrate the use and adoption of SBOL as a standard.

These supporting software libraries in three languages, combined with their documentation and the standard specification document, constitute the deployment of the standard to the broader community. The cycles of deployment, feedback, and revisions helped improve the technical aspects of the project. Most importantly the participants had implemented the standard and were ready to demonstrate its use.

5.3 Demonstration of Use

To illustrate capabilities that are described in Section 5.1, I demonstrated the ability to exchange synthetic biological designs within software tools built for synthetic biologist. My collaborators and I performed these demonstrations. These demonstrations illustrate SBOL capabilities through use in several scenarios. I first describe the publication of Collections of DNA Components using the Parts Registry, BIOFAB electronic datasheets, and the Newcastle Bacillo Bricks repository. Then, I describe how the individual annotated DNA sequences from those collections were utilized within the biological design tools Clotho, iBiosim, GeneDesigner, TinkerCell. These demonstrations show the successful application of the SBOL data exchange standard.

First, I describe the publication of annotated DNA components from the Parts Registry. Then, I describe the work of two sites which adopted SBOL to publish two independent repositories of components: the BIOFAB's SBOL collections of constructs and the Newcastle BacilloBricks repository. Furthermore, I describe the adoption of SBOL within software tools which utilize the public repositories to provide their users with a rich set of components to use in new designs. The first of these tools is GeneDesigner from DNA 2.0 which provides a web based import of the BIOFAB collections. Finally, I recount a demonstration of exchange performed by three independent teams. In this demonstration, the Newcastle BacilloBricks registry provided the source components, the Utah team designed a new construct in their iBioSim tool, and the Boston team imported the final design into Clotho, their integrated management application framework.

5.3.1 *Publication of Collections of DNA Components*

The first demonstrations involve the publication of entire collections of DNA components on the web. I chose to start with this demonstration as it immediately makes DNA components freely and openly available to other software tools, which then make use of DNA components downstream. The Parts Registry was available publicly and freely on the web at the beginning of this project. Therefore, to demonstrate re-use of parts described in SBOL, this was the most opportune example. Later, I was also able to demonstrate these capabilities using in collaboration with the BIOFAB (Section 5.3.1.2) and Newcastle University (Section 5.3.1.3).

5.3.1.1 Publication of Parts Registry Data

To create the first collection of DNA components in SBOL I used the information available from the Registry of Standard Biological Parts (partsregistry.org). This collection is a semantic knowledgebase for synthetic biology which used SBOL Semantic, an early version of SBOL (see Figure 5.2). I also extended it with new terminology acquired from the Registry to describe biological parts. The extended SBOL *class* structure helps to add more informative information retrieval capabilities (see Section 5.4).

To transform the Parts Registry data, I first extracted the Registry data and mapped its structure of tables, its relational schema, to SBOL semantic. This mapping served as my translation table to transforming the Registry data of 13,444 part entries and the associated Sequence Features to OWL/RDF. Using a script, I converted 13,444 Registry part records with their associated Sequence Features from the Registry format to the SBOL semantic (OWL/RDF) form. Each Registry part record was also associated with the Registry's Sequence Feature table, a position based description of the nucleotide sequence (see Figure 2 for example sequence features such as a 'terminator'). I then mapped the Registry Sequence Feature table to the SBOL Sequence Annotation and Feature Class structures and performed the analogous translation into OWL/RDF.

As part of the transformation of Registry data I used the categories attribute of the Registry *parts* table to provide a richer description of parts. The Registry includes a total of 346 categories organized as a hierarchy of 28 top level categories (e.g. chassis, classic, dna, function, plasmid, plasmidbackbone, primer, promoter, proteindomain, proteintag, rbs, regulation, ribosome, rnap, terminator, etc. For full listing see Supporting Information Table S1 in [147], which contains the list of terms extracted from the Registry data, and File S1. In [147], which contains the generated OWL encoded semi-structured controlled vocabulary used throughout this work). These categories are a rich vocabulary used to describe parts and constitute a controlled vocabulary, created and maintained by the Registry staff, while its use is enforced by the Registry website software application. The categories form the basis of organization for the Registry Catalog website. Thus, to provide an effective structure for querying the Registry information, I needed to augment our core SBOL semantic ontology with this terminology. To do so, I auto-generated a class structure within SBOL semantic that mimics the registry category structure. For an example, see Figure 3. The addition of these descriptions as extended classes, which are not found in the standard, demonstrates the open nature of the framework [25] by extending its class structure to support the needed concepts from the Registry. In this demonstration I showed the Parts Registry translated into SBOL. This translation could be queried as part of the SBPkb (see Section 5.4.1). The automatic translation also demonstrates the feasibility of the approach for other large scale collections.

Recently, I updated the process described above to be compliant with the new SBOL version 1.1. I re-implemented the Parts Registry data to SBOL conversion process using Extensible Stylesheet Language Transformations (XSLT) an XML transformation language. The tool works as a web based request to the Parts Registry, and transforms its native XML output to SBOL. As this *converter* tool runs at the time of the request, the SBOL data from the Registry are always up to date.

5.3.1.2 Publication of BIOFAB Data

The BIOFAB: International Open Facility Advancing Biotechnology (BIOFAB) developed and then published three collections of DNA components using SBOL on the web. As the first independent publisher of SBOL data, the BIOFAB's ability to publish Collections of DNA Components in an early version of SBOL format, demonstrated initial operational sufficiency. They released the Modular Promoter Library composed of 125 sequence promoters, the Random Promoter Library of 156 promoters, and the Terminator Library of 40 terminators. The Modular Promoter sequences are composed of modules whose boundaries were selected rationally. However, this work is not published in literature, therefore no description of how the random promoter sequences were generated is provided. The Terminator Library is comprised of a mixture of natural, synthetic, random spacers, and other negative controls. Since, SBOL is extensible; an alternative approach for the BIOFAB would be to create an extended classification scheme (analogous to the work described in Section 5.3.1.1). However, to include a vocabulary for how the sequence was generated would require the BIOFAB scientists to specify the Sequence Ontology type of each component, as well as create a new class structure to describe how the promoter sequence was generated, as modules, randomly, and only then could it be published. Instead, a simpler approach was applied and the newly introduced Collection object was used to create these *ad hoc* groupings and to publish the data on the web via their Data Access Web Service [148] using a modified version of libSBOLj, which provides JSON serialization. Additionally, the BIOFAB extended the model to include quantitative functional characterization information to describe the performance of the constructs. In 5.3.2 I describe how these collections can be used in the GeneDesigner software tool made by the DNA synthesis company, DNA 2.0.

Dr. Cesar Rodriguez implemented the SBOL file downloads as part of the BIOFAB Electronic Datasheets. The server side BIOFAB Data Access Web Service provides descriptions of BIOFAB generated designs using the JSON based serialization (Figure 5.2). Dr. Rodriguez was able to use the libSBOLj library to publish the SBOL JSON data. Additionally, Data Access Web Service includes a functional characterization information extension of the SBOL framework. This extension is a proposed representation for DNA component performance information. The early adoption of SBOL by the BIOFAB, an independent synthetic biology center, demonstrates that SBOL fulfills that center's need to publish its designs on the web.

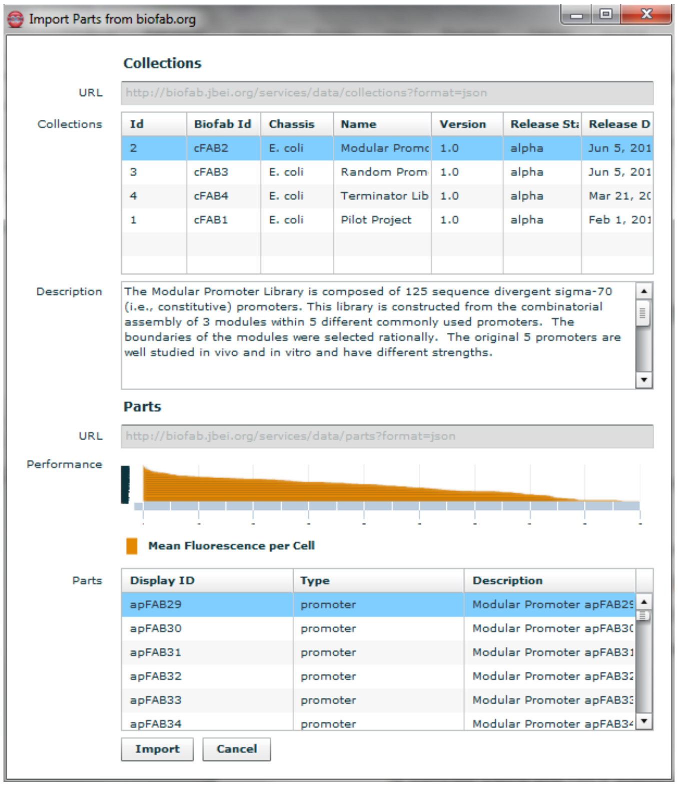
5.3.1.3 Publication of Newcastle University Data

Further confirmation of operational sufficiency was demonstrated by a team from Newcastle University. Dr. Matthew Poccock and Goksel Misirli implemented a conversion of the Bacillo Bricks repository at Newcastle to SBOL. The BacilloBricks registry contains a large collection of standard virtual parts (SVP) [23] comprised of an SBOL DNA component and a SBML model annotated with meta-data. SVPs are used as modules to build dynamic models of synthetic genetic systems. Their implementation used the libSBOLxml library to serialize 2995 DNA components from *Bacillus subtilis* in the SBOL xml serialization. In the next section I describe how the University of Utah team used a few selected SBOL DNA components from BacilloBricks to demonstrate their use of SBOL in a new design.

5.3.2 *Use of SBOL for Design*

The second set of demonstrations show the use of DNA components within software tools which use SBOL to help design. The tools use disparate native formats, thus until now they had not been interoperable. Currently, the availability of public collections of synthetic DNA components in the standardized SBOL representation opened the possibility of their re-use in new designs. I describe the independent implementations which made re-use of DNA components encoded in SBOL possible. The first example is use of the BIOFAB collections in the Gene Designer software tool to design new genetic constructs. In the second example I describe how iBioSim was used to make a new design, using DNA components from the BacilloBricks registry, and then loaded into Clotho, an information management system. The work described here was performed by collaborators from the SBOL Developers group which demonstrates the active adoption of SBOL and illustrates its use as a standard by independent researchers.

<u>*Use of BIOFAB components in Gene Designer*</u>: This DNA 2.0 software provides a graphical interface for the design of synthetic DNA segments (Villalobos 2006). Gene Designer's set of drag-and-drop operations enable researchers to manipulate genetic constructs and to optimize the codon usage within the designed genes. Version 2.0.170 includes an import option from biofab.org (Figure 5.3).



Import Parts from biofab.org

Collections

URL: http://biofab.jbei.org/services/data/collections?format=json

Collections

Id	Biofab Id	Chassis	Name	Version	Release St:	Release D
2	cFAB2	E. coli	Modular Promc	1.0	alpha	Jun 5, 201
3	cFAB3	E. coli	Random Prom	1.0	alpha	Jun 5, 201
4	cFAB4	E. coli	Terminator Lib	1.0	alpha	Mar 21, 20
1	cFAB1	E. coli	Pilot Project	1.0	alpha	Feb 1, 201

Description: The Modular Promoter Library is composed of 125 sequence divergent sigma-70 (i.e., constitutive) promoters. This library is constructed from the combinatorial assembly of 3 modules within 5 different commonly used promoters. The boundaries of the modules were selected rationally. The original 5 promoters are well studied in vivo and in vitro and have different strengths.

Parts

URL: http://biofab.jbei.org/services/data/parts?format=json

Performance

Parts

Display ID	Type	Description
apFAB29	promoter	Modular Promoter apFAB2
apFAB30	promoter	Modular Promoter apFAB3
apFAB31	promoter	Modular Promoter apFAB3
apFAB32	promoter	Modular Promoter apFAB3
apFAB33	promoter	Modular Promoter apFAB3
apFAB34	promoter	Modular Promoter apFAB3

Figure 5.3. Gene Designer import panel. Offers choice of BIOFAB collections, displays performance data for each collection, and allows a user to pick the Part to import.

This import functionality uses the JSON serialization of SBOL provided by the BIOFAB Data Access Web Service. The import panel displays four collections available from the BIOFAB (fourth collection "Pilot Project" is not found on BIOFAB's public web pages). A *Part*, which corresponds to a SBOL DNA Component, can be chosen and imported from one of these collections. The import function then generates a new Gene Designer DNA Element using the identifier and DNA sequence from the SBOL DNA component, in the main interface of Gene Designer.

Design using BacilloBricks in iBiosim: In the second example three groups of collaborators used SBOL to exchange DNA components between a registry, a CAD tool, and an information management system. CAD software (i.e. iBioSim, Tinker Cell), typically stores knowledge locally, therefore they had limited access to knowledge about components known to work from previous projects. However, availability of a broad range of components for creating new designs is tremendously valuable to a synthetic biologist. Re-use is absolutely necessary for realistic chances of the design's success. These tools use disparate formats, thus are not interoperable with each other First, the Newcastle team selected three SVPs from BacilloBricks: 1.) a constitutive ftsH SigA promoter; 2.) the response regulator Spo0A coding sequence, and; 3.) the ydfHI terminator. These three component types were selected for the data exchange demonstration because they can be combined to form an expression cassette. Newcastle made these parts available to Utah as a SBOL document for import into the iBioSim CAD tool. The Utah team created a simple expression cassette design using the iBioSim application. They combined the three components and added an appropriate RBS DNA component. They sent the device design described in SBOL back to Newcastle and also to the Boston University team. The Boston team was able to load the design into the Clotho application an information management system which enables an integrated workflow environment for synthetic biology.

5.4 INFORMATION RETRIEVAL FOR STANDARD BIOLOGICAL PARTS

To support DNA component re-use I built a computationally accessible knowledgebase of information about standard biological parts. I designed this library leveraging the engineering principles of standardization, decoupling, and abstraction. If synthetic biologists had effortless access to information about previously used parts, they could use this information to more efficiently design and plan for the assembly of new genetic devices. When already available components exist, and have been shown to work, their reuse would allow a biological engineer to focus on meeting design requirements, rather than re-creating prior work of others. I performed this work in collaboration with Dr. Cesar Rodriguez, Deepak Chandran, and Professors Herbert Sauro and John Gennari.

5.4.1 *Standard Biological Parts Knowledgebase*

We created the Knowledgebase of Standard Biological Parts (SBPkb) as a publically accessible Semantic Web resource for synthetic biology [147]. The SBPkb allows researchers to query and retrieve standard biological parts for research and use in synthetic biology. Its initial version includes all of the information about parts stored in the Registry of Standard Biological Parts (partsregistry.org). SBPkb transforms this information so that it is computable, using our semantic framework for synthetic biology parts. This framework, known as SBOL semantic, was built as part of the Synthetic Biology Open Language (SBOL), a project of the Synthetic Biology Data Exchange Group. SBOL semantic, an early version of SBOL Core, represents commonly used synthetic biology entities, and its purpose is to improve the distribution and exchange of descriptions of biological parts. I describe the data, methods for transformation to SBPkb, and finally, I demonstrate the value of our knowledgebase with a set of sample queries. I use RDF technology and SPARQL queries to retrieve candidate "promoter" parts which are known to be both negatively and positively regulated. This method provides new web based data access to perform searches for parts that are not currently possible.

5.4.2 *Introduction*

The Standard Biological Parts knowledge base (SBPkb), the initial version of a biological parts library that supports remote queries. This library builds on knowledge from the Registry of Standard Biological Parts (partsregistry.org), which we described in Chapter 2. I adapted and transformed data from the registry that describes standard biological parts using RDF, into SBOL, as described in Section 5.3.1.1. Next, I demonstrate how the SBPkb can be queried using standard RDF technology (SPARQL queries) to retrieve parts that may be relevant to a synthetic biologist. I take as a use case queries about promoter parts. In the results section, I show (a) that such queries cannot be pragmatically answered with current technologies, and (b) that the approach allows researchers to carry out query refinement. For the latter, I show that our promoter query can iteratively be made more specific, so that the query results in smaller lists of parts, and where these parts are better matched to specific design criteria.

5.4.3 *Making SBOL semantic data available for retrieval*

I loaded the SBOL semantic data created by the transformation process described in Section 5.3.1.1 into a framework for querying RDF data, creating the Standard Biological Parts Knowledgebase resource (SBPkb) (see Section 5.4.7 for availability). As I show in the results section, these categories can be used to directly query the SBPkb for specific features of parts.

The semi-structured controlled vocabulary resulting from this process does not fulfill many of the criteria of formal ontology design [149]. The structure created reflects the organization found in the Registry, and is not a proper class hierarchy. The SBOL semantic structure was directed towards SPARQL query information retrieval, translating the existing Registry information to a Semantic Web technology standard to enhance its potential for re-use. This utilitarian approach provides immediate benefit of data access and lays out the scope of the knowledge engineering challenges which face the synthetic biology community. Challenges of formally structuring information for future use in multiple applications are especially evident in large collections such as the user-driven and community-supported data source for our work, the Registry of Standard Biological Parts. However, the main contribution of this work is to provide a pragmatic solution for synthetic biology users, and establish the need for improvement of information resources in the field.

5.4.4 *Results: The case of the promoter*

To illustrate the functionality of SBPkb I describe a hypothetical case for its use to research the availability of promoters for a new design. We asked the knowledge base to answer the following question, “Which promoters can I use for a design?” Because “promoter” is a class in our controlled vocabulary, this is a straightforward SPARQL query to ask of our SBPkb (see query #1 in File S2), and it returns 538 parts that are annotated as promoters.

Although this query seems simple, we must compare the capabilities of SBPkb to current technology: How would one answer this question, with current technology, i.e., directly of the Parts Registry? Unfortunately, the only way to retrieve this set of parts is by manual browsing of web pages, and then manual compilation and analysis of the results listed on these web pages (also see the comparison section below). Additionally, SBPkb and SPARQL allow researchers to easily refine queries and provide cleaner, more useful results. Users can also narrow the search to a more specific type of promoter. In this section, we describe how our initial query can be step-wise narrowed to a much more specific query that returns only six parts from our knowledgebase.

As a first step, we ask what information is associated with these parts—we carry out a SPARQL *describe* query (query #2 in File S2) that lists the complete set of properties associated with all promoters. This query would have a lengthy, large result, but we can sample only a few entries to explore the information space; Table 1 shows one sample entry from this query result. By looking at all available properties of a part, researchers may discover ways to narrow or improve their query. For example, an initial exploration may lead us to decide that the *status* property is important (we do not want any "deleted" parts), and that we want only parts that have DNA sequences listed. This refined query (query #3 in File S2 produces 529 parts (it eliminated seven "deleted" entries, and two without DNA sequences).

Trivially, we can also ask these sorts of "data cleaning" questions of the entire SPBkb. For example, we found that 12,152 of the 13,444 total part records have an associated DNA sequence and have not been marked for deletion (query #4 in File S2). Currently, many parts are larger in DNA sequence length than is financially prudent to directly synthesize, however not impossible using the latest methods [150]. Therefore, it is noteworthy that only 5,166 are marked as *Available* or as *Sent* to the Registry as clones (query #5 in File S2).

5.4.5 *Comparison with current capabilities*

To validate our (cleaner) result of 529 promoter parts found via our SPARQL query and the SBPkb, we also attempted to answer this question by exhaustively browsing the Parts Registry. First, we dismissed an information retrieval approach that might use heuristic algorithms based on text searches of the word "promoter" within the Registry's web pages (e.g. a Google search). Although careful construction of good heuristics might lead to accurate results, a simple text search will result in many entries that mention "promoter" but are not themselves promoter parts.

We used an exhaustive manual method, systematically exploring all web pages in the 'Promoter' category of the Parts Registry Catalog. When information appears about parts, the Registry Catalog typically displays the information in a table. Therefore, whenever we encountered a page with parts labeled as a category of promoters, we copied the corresponding table into a spreadsheet application (MS Excel™). This exploration results in 42 separate web pages (many with several tables) and a total of 833 promoter parts. (Data collected by MG on Aug 3, 2010 from partsregistry.org/Promoters/Catalog). Because the same part can be found on multiple web pages, the same part identifier can be copied onto the spreadsheet multiple times. We removed these duplicate entries using the Remove Duplicates Data Tool in Excel™ and obtained a unique list of 474 promoter entries. Finally, we noted that two of these lacked DNA sequence information, a requirement of our "cleaner" query.

The set of 472 entries that we found manually are all included in the set of 529 promoters returned by SBPkb. That is, no information was "missed" by our knowledgebase. SBPkb also retrieved 57 additional entries that appear to be bona fide promoters, from a variety of subcategories. We attempted, but were unable to discover why these particular promoters were missing from our manual browsing of the web pages (see Table S2. for this list of 57 promoters).

It should be clear that exhaustive web page browsing is not a scalable approach to searching for a particular class of biological part. Indeed, the registry instead is a community-based, wiki-style collection of parts dedicated to capturing information about parts. Supporting such queries is a novel design consideration for a semantic web of data in synthetic biology. Query answering is a central design feature of the SPBkb, and as we demonstrate next, our initial query can be narrowed to return a much smaller set of parts, yet still maintain the ability to exhaustively search the knowledge base.

5.4.6 *Design Query Refinement*

The process of query refinement, or improvement of the query, as a specification of information needs, involves exploration in order to discover information about a topic [151]. We again look through the results of query #2 in File S2 to find additional criteria by which to search SBPkb. The query driven exploration process helped us discover the rich source of structured information derived from the Registry categories. Among the results of this query (Table 1), we found that the example promoter part belongs to the type or category, 'sigma70_ecoli_prokaryote_rnap'. The categories, represented as OWL *classes* in SBOL semantic, provide the capability to refine queries for promoters. For example, to narrow the selection to only those promoters, which are expected to work with the *Escherichia coli* RNAP σ^{70} holoenzyme (Eσ^{70}) and therefore to have an expected peak efficiency at the exponential growth phase [152]. This query (query #6 in File S2) results in 367 "Eσ^{70}" promoters, a subset of the 529 promoters found in our initial query. This list of 367 is the most likely candidates to use for common synthetic biology experiments in *E. coli* for which measurements are taken at mid-exponential phase. The capability of retrieving specific parts from the thousands of entries within SBPkb by selection criteria such as the *class* structure of biological system contexts will allow synthetic biologists to find parts relevant to their design.

Not only were we able to retrieve promoter parts based on specific factors (σ), but available to us as selection criteria were also Registry categories which specify the expected mode of regulation. For example, during the design of a new genetic Barkai-Leibler oscillator [153, 154] the synthetic biologist may want to find all pre-existing promoters that can be both 'positively regulated' AND 'negatively regulated', i.e., dual-regulated promoters (query #7 in File S2). Our query returned just 36 unique promoter parts meeting these criteria (note that this query result is not necessarily a subset of the 367 "Eσ^{70}" promoters). The Barkai-Leibler oscillator relies heavily on the availability of such dual-regulated promoters; therefore, having knowledge of all dual-regulated promoters available in the Registry is highly advantageous to the synthetic biologist. Since a sufficient number of dual-regulated promoters are available, the search can be further limited to promoters for known specific inducers and repressors that are appropriate for the new design. The SBPkb includes information from the Registry Features table; therefore, for our final refinement we further restricted our query to return promoters that have sequence annotations of known transcription factor binding sites, i.e., operator sites. This example query (Figure 4) returns just six parts and their known binding sites (Table 2). A selection of these six candidates provides a list small enough that each one can be examined in greater detail for relevance to a specific design.

During planning stages of a new synthetic biology research project investigation of prior work is an important phase of forming a new design. This process involves the exploration of available information resources for the purpose of discovery of candidate components to leverage in such a design. The SPARQL *describe* query in SBPkb can help identify information types or classes, such as Registry categories and data fields that hold information management, engineering, or biologically relevant information. These facts, or descriptions of parts, can then be used to search across the entire information collection to identify parts relevant to a particular design specification or criteria. This ability to quickly identify specific parts that match design criteria provides a method that enables fast and thorough exploration of prior work.

5.4.7 *Implementation and Availability*

To construct SBOL semantic we used Protégé 4.0.133 (protege.stanford.edu) and used a RDFlib (rdflib.net), a python library, to perform programmatic additions of class terms and individuals during the data import process. We obtained the Standard Biological Parts Registry data from (partsregistry.org/Registry_API) on April 6, 2010. The downloaded information was provided in the form of two MySQL tables formatted as XML, a table of parts and a table of Sequence Features. These were converted into a text based delimited format to serve as input for SBPkb. We created python import scripts to parse the input tables from the Registry and libSBOL, a python library, to aid population of SBOL structures to generate the RDF/XML form of the data for SBPkb (synbiolib.sourceforge.net).

We have made the SBPkb data accessible via SPARQL a W3C recommended query language for RDF queries, with remote access (through a RESTful HTTP interface) provided using the Sesame 2.3.1 (openrdf.org) software. The SBPkb (sbpkb.sbolstandard.org) as a SPARQL accessible knowledge base is a publically available Semantic Web computational resource for the synthetic biology community.

5.4.8 *WikiDust: An example interface*

SBOL DNA components from the Parts Registry can serve as candidate components for new designs using a graphical interface. Here I describe the WikiDust plugin a search interface for TinkerCell [64]. I helped Jeffrey Johnson from Prof. Herbert Sauro’s group to develop this plugin.

WikiDust [155] uses TinkerCell’s graphical interface to build a query for a desired DNA component. The query is used to retrieve information from the repository of DNA components. TinkerCell offers a palette of icons representing DNA component types, such a promoters, coding sequences, terminators, etc. which can be dragged onto the canvas, arranged graphically, and labeled.

Using TinkerCell, a synthetic biologist can design a model of a synthetic gene circuit. Then, using the WikiDust plugin the researcher can query the SBPkb for a DNA component selected on the canvas. The query is built up from the context of the gene circuit design on the canvas and helps retrieve a candidate component found in the SBPkb repository. The queries are formulated by interpreting the label and DNA component type from the TinkerCell API and mapping it to the corresponding Sequence Ontology term. For example, different queries are issued if the researcher types in a label using a BioBrick identifier (e.g. BBa_B0015) or a common name (e.g. GFP). The plugin then issues the query using the SPARQL query language [156] over an HTTP connection. It returns a list of candidate components. Users can then refine their search by entering additional keywords or phrases, and can use the source link to open and browse the corresponding Parts Registry web pages for more detailed information. It uses the information it received to populate any missing information fields for that TinkerCell object. For example, it retrieves the DNA sequence, name, and identifier.

The main advantage of WikiDust is that it builds the query for the user and therefore hides the queries written in the SPARQL syntax from users. This prototype for a query builder interface demonstrates the feasibility of integrating the SBPkb information retrieval capabilities into a synthetic biology CAD software tool.

The WikiDust plugin currently uses version 2 of the SBPkb. This SBPkb is an updated version of SBPkb, which runs in the Stardog RDF database by Clark & Parsia, LLC. It is populated with all Parts from the Parts Registry converted using "Converter" into SBOL v1.1 (see Section 5.3.1.1). In the future I hope other CAD tools will take advantage of the standardized information retrieval capabilities demonstrated by the SBPkb.

5.4.9 *Discussion*

To effectively build new systems from prior work and best practices, synthetic biologists developed an initial framework and standards for the description of engineered biological devices [79], [157]. The common approach of storing data about biological parts in a spreadsheet is convenient for a small laboratory. Our experience in synthetic biology research suggests that sharing such information between collaborating laboratories requires a significant coordination effort. Furthermore, *ad hoc* organization of part description information is too ambiguous to establish an efficient engineering pipeline for synthetic biology. The process of engineering synthetic biological systems relies on specialized software tools to: model systems, aid design, and plan assembly. For software to help researchers make appropriate design decisions, biological parts must be described using an unambiguous language, such as SBOL. To reconcile the need for engineering with base pair precision with the inherent complexity of biological system dynamics at multiple scales, there is a need for software tools to have the ability to exchange information about the entire spectrum of the domain of synthetic biology. Working towards the goal of defining an unambiguous computational language for synthetic biology, we have created Standard Biological Parts Knowledgebase (SBPkb). This public resource uses the Synthetic Biology Open Language semantics (SBOL semantic) as its organizing structure and demonstrates its use for information retrieval.

Current methods for finding previously described biological parts are insufficient to realize new synthetic biology designs with increased sophistication. To create such integrated systems from parts and modules synthetic biologists must overcome significant challenges posed by the uncertainty and complexity of biology [44]. Synthetic biologists need to be able to draw on large numbers of examples of prior work to learn from the successes and failures of previous efforts. We have populated the SBPkb with the thirteen thousand parts from the Registry of Standard Biological Parts, and we have made it available for public use. Purnick & Weiss [44] reported that the most complex system built up to that time, as measured by the number of regulatory regions within a design, was six. Automatically searching the SBPkb, for existing candidate parts, will increase the number of part options to consider in designs. This ability, to quickly query part information from the large repository of knowledge provided by the Registry, removes one significant barrier in the exploration of prior work.

The ability to query SBPkb using a remote query protocol can serve to extend the capabilities of computational tools which support design work. Software designed to help synthetic biologists to plan designs can greatly benefit from a computationally accessible search interface. Information retrieved from SBPkb by SPARQL is returned as SBOL semantic RDF/XML therefore can easily interpreted by the receiving application. For example, TinkerCell [64],[158], a computer aided design application, could use SBPkb queries to fulfill designs based on combinations of specific requirements. We demonstrated one such hypothetical query for promoter parts controlled by dual modes of regulation. TinkerCell, and other design tools, could take advantage of query results to suggest these candidate parts to a user who is building a new Barkai-Leibler oscillator. The use of query refinement as a method for specifying design requirements would be an important methodological development towards automating the design to production pathway in synthetic biology.

SBOL semantic is based on the robust principles and technology developed by the Semantic Web research program. The utility of the approach we described provides information retrieval services via a standard query language, SPARQL.

Due to the amount of detail inherent in any biological system and the distributed nature of scientific research, a semantic-web based solution for organizing synthetic biology data is the suitable choice. The SBOL framework described in this work can be used to unambiguously describe, remotely query, and therefore electronically retrieve information about biological parts. In the ideal scenario, researchers would also use a front-end software application to submit parts, as TinkerCell's WikiDust plugin is used to retrieve parts from the SBPkb. Embedding SBPkb query utilities in the user friendly graphical interfaces of software will help us bring these capabilities into the workflow of active synthetic biologists.

In the validation portion of this work we demonstrated that searching for part information using a manual process is not a scalable or pragmatic method. Searching the web pages requires manual compilation and curation for each information query; such methods are not scalable in the face of the continually growing number of available biological parts. Using SBOL semantic to describe synthetic biology concepts not only allows electronic retrieval, but offers the ability to select specifically defined subsets of parts. This further integration of SBOL semantic with software will help encourage re-use of previously described components, a best practice of synthetic biology.

Reuse of components in synthetic biology research is one key way in which biologists can more easily engineer and construct new systems with increased complexity. The SBOL framework allows us to capture the semantics of richly-structured descriptions and to incorporate new information needed for design in synthetic biology. Automation of design promises to make building biological machines more efficient. Finding parts that meet the specifications of designs is a critical aspect of automation of the engineering process. Leveraging Semantic Web tools (such as SPARQL) to perform information retrieval can fulfill this need and offer additional benefits such as consistency checking and classification through automated inference. Adopting these capabilities to biological system design should allow engineers to use previously created solutions and apply them to solve novel problems.

5.5 CONCLUSIONS

The overall goal of this work is to reduce the technical barrier to sharing structured information describing synthetic biology constructs and their functional characterizations. Information about engineering components is a key property of the results of synthetic biology research. I believe sharing results of engineering efforts within the laboratory, with collaborators, and publicly, is needed to increase the potential for re-use of engineering components. Therefore to improve the likelihood of re-use through sharing, the information must be described in terms of characteristics useful in understanding the potential function of produced components.

Chapter 6. CONCLUSIONS AND FUTURE DIRECTIONS

Every time you create standardized interfaces, you open the door to 3rd party services and products.
--Vinton G. Cerf, Vice President and Chief Internet Evangelist of Google

The development of SBOL is the first step forward in standardization of information exchange in synthetic biology. SBOL enables researchers to exchange DNA designs using disparate synthetic biology software tools to support their biological engineering workflow. The notion of SBOL is derived from biomedical informatics best practices, Semantic Web technology, and firmly rooted in requirements from the synthetic biology community. Synthetic biologists are researchers and engineers studying and building organisms to develop and improve new medical treatments, biofuels, bioremediation, and nutrition. Their research results contribute not only solutions but also reflect on our understanding of natural biological system design. These researchers rely on the engineering principles of standardization, decoupling, and abstraction to make biological design research a tractable endeavor. My overarching goal in this dissertation was to apply these principles to improve the computational tools synthetic biologists use. The practical result is enabling these engineers to transfer designs from one computational tool to another. This technical ability also enables researchers and companies to share designs between groups. Most importantly, SBOL opens the door to third party services and products.

The SBOL standard is free, open, and sustainable. Anyone has the right to use SBOL without cost. The specification document, software, and documentation are all licensed using free and open source licenses. Furthermore, the standard is supported by a community. This support is manifested by the contribution of the members to its development and in its adoption. I give the credit for its success to the SBOL Developers community, as this group has adopted it as their standard method for exchanging DNA designs. Furthermore, SBOL Developers are now working to extend the standard to incorporate additional information critical to engineering biological functions and systems.

The success of SBOL is evidenced by its growing community and adoption in software. According to a self-report survey I performed among the SBOL Developers in July 2012 and earlier reports; there are fourteen computational tools which currently support SBOL:

1. Eugene – Script
2. Hermes – Clotho Messenger App *

3. J5 – Assembly planning *
4. SBOL GenBank conversion utility *
5. VectorEditor – DNA design editor [159] *
6. JBEI-ICE – Public Repository *
7. iBioSim – CAD system modeling *
8. Gene Designer – Gene sequence optimization and design
9. Registry of Standard Biological Parts – using SBOL converter
10. MoSeC – Model-to-sequence conversion tool
11. Bacillo Bricks – Catalogue of parts and models
12. Tinker Cell – CAD tool for synthetic biology via WikiDust
13. GSL – Amyris, Inc. internal language [160]
14. BIOFAB Electronic Datasheets – via the Data Access Web Service [148]

Some tools, marked with an asterisk in the list above, are capable of both reading and writing SBOL and the others operate in only one direction. These software tools and the SBOL support were all developed by independent groups which have representatives participating actively in the SBOL Developer group. These stakeholders in the synthetic biology software community have demonstrated their support of the SBOL standard. There are over fifty members in the SBOL Developers group [132]. They are representatives of twenty-two institutions and are equally representatives of both academia and industry. Additionally, nine peer reviewed publications already cite SBOL [161]. These first adopters of SBOL have independently demonstrated use of SBOL as I described in Chapter 5. Its strong adoption by these independent and diverse stakeholders suggests that SBOL fulfills a needed role as a data exchange language. Furthermore, it indicates that developers of other tools should implement SBOL import and export support to electronically communicate with a large number of synthetic biology software tools.

SBOL provides the capabilities which put software tools on the path to improving the efficiency of the engineering cycle. For example, biological engineers can now begin to expect DNA components to be available for download in SBOL format. In this dissertation I showed how the one SBOL representation is used for both peer-to-peer exchange and re-use from a data publisher project for a new design in another project. I demonstrated not only the community adoption of a static format, but its evolution over time. Furthermore, as synthetic biology evolves the SBOL Developers will update the standard to continue to meet the specific information needs of synthetic biologists.

In this chapter I convey the conclusions of my dissertation and discuss future directions. First, I summarize the main contributions of my dissertation research and its implications for both synthetic biology and biomedical informatics. Then, I discuss the challenges and limitations of SBOL. These challenges present opportunities for further research and expansion of the SBOL standard. Finally, I conclude the chapter with an overall vision for the engineering of whole organisms.

6.1 Dissertation Summary

In this dissertation I presented the development of SBOL. The research I presented is the beginning of the development of a standardized computational language to organize the vast and inherently complex biological systems engineering information. This information is needed to realize the full potential of rational design for synthetic biology.

In Chapter 2 I introduced the field of synthetic biology and the Semantic Web technology needed to develop SBOL. The motivation for this work is based on the premise that synthetic biologists pursue biological design research to both inform the practice of forward engineering and as a practical test of understanding how biological systems function. The pragmatic test of the completeness of knowledge about the system is the ability to reconstitute an organism from its basic parts. Synthetic biologists research how living systems function by designing and constructing their DNA anew. In this engineering and research field the DNA designs are the core information elements which synthetic biologists manipulate and manage using software tools. I then proposed a solution for the sustained development of a standard based on Semantic Web standards and technology. These standards and technology are designed to support a network of semantically described data on the web and therefore enable a robust computational framework for the exchange of such information.

In Chapter 3 I discussed the role of the synthetic biology community in forming the Synthetic Biology Open Language (SBOL). I provided a look at the history of successful standards in biomedical informatics and systems biology which underlines the importance of the social context in which standards are developed. The key features of standards communities are a grassroots beginning and the responsiveness to the evolution of the field. The standard they create must be simple, immediately useful and its implementation must be supported by free and open source software libraries. Furthermore, I reported on how I engaged the community stakeholders to form the collaboration and eventually the SBOL Developers group. I concluded this chapter with a discussion of the impact on the broader synthetic biology community in terms of policy requirements introduced by funding agencies to mandate the use of the SBOL standard.

In Chapter 4 I described the design of the SBOL architecture along with the technology developed. I focused on the concept of DNA components as a design object for synthetic biologists. I described the vocabulary, data model, and serialization format which specify the requirements of the SBOL Core. To provide a controlled vocabulary and potential for rigorous classification of DNA components I discussed the use of the Sequence Ontology within SBOL. Additionally, I provided the information about the availability of the software libraries and further documentation for the SBOL standard project. The formal specification and supporting software libraries described in this chapter helped other SBOL Developers implement the standard.

In Chapter 5 I described the capabilities SBOL provides, the deployment strategy, and the demonstrations of its use for data exchange. I discussed the use of SBOL to send DNA design templates, to send annotated DNA sequence, and to publish collections of DNA components. Then I reported the results of the deployment of SBOL to the community. I describe how scientists from the BIOFAB used SBOL to publish DNA designs and the use of that information in Gene Designer. Then, I described the round trip exchange from the Bacillo Bricks Repository to iBioSim for design, then to Clotho, and back to the Bacillo Bricks Repository. Finally, I described the use of SBOL in the SBPkb for information retrieval of DNA components from the Standard Biological Parts Registry, to select candidate components for a new design.

My success in aiding the formation of the SBOL community is substantiated by the fifty-two members of the SBOL Developers group. The enthusiasm for the adoption of SBOL is evidenced by the twelve software applications which have already adopted the standard. Furthermore, the outlook for future of SBOL is bolstered by the contribution of work by the members to community projects, such as to develop extensions, maintain the software, and provide funding for workshops. The consequence of these achievements is the prospect of increasing efficiency of the biological engineering process.

6.2 Implications for Synthetic Biology

The introduction of SBOL to the broad synthetic biology community bears the promise of augmenting the biological engineering workflow. This standard method for exchanging synthetic biology data reduces barriers to transitions between the stages of the engineering cycle. For example, it is now easier to retrieve candidate components within a computational design tool. This tool then transfers the new design composed of sub-components to another tool for assembly planning. Finally, these designs can be submitted to institutional or public repositories. A self-reinforcing cycle of design, construction, testing, and then analysis and re-use of successful designs is the vision for an effective modern biological engineering process. The potential gain in efficiency could be tremendous when the possibility of re-use of components across different engineering and research projects is considered.

Software tools, such as Tinker Cell and iBioSim, are used for design in synthetic biology and already capture rich descriptions of constructs. These designs map naturally to the SBOL structure. Tinker Cell, a synthetic biology CAD tool, provides a powerful platform to explore putative genetic circuit designs. The capabilities of this tool to produce realistic designs can greatly benefit from direct computational access to a repository of existing parts and previously formulated designs. I helped develop this capability through the implementation of the SBPkb and the collaboration to develop the WikiDust query interface.

Also, the ability to export the final or proposed design to the assembly planning stage provides a benefit to the engineer in terms of accurately transferring the design layout. This also saves time by reducing the need for manual entry into assembly planning software. Preparing a plan for the manipulation of a DNA sequence is an essential step of the synthetic biology engineering process. To aid in assembly planning, synthetic biologists use DNA sequence editing software or specialized tools such as j5. iBioSim now provides the ability to export a design created and pass it to a tool such as Clotho. However, even though transfer of a design from iBioSim to the j5 software has not yet been demonstrated, it is conceivable that we can expect it to simply work. Both tools support the necessary export and import functions. With the growing adoption of SBOL, such untested exchange scenarios should be possible since the each implementation conforms to a common standard.

An additional implication of the capabilities of the supporting software is that SBOL files with a complete and annotated DNA sequence can be converted to or created from the popular

GenBank flat file format. This enables synthetic biologists to still use the widely known molecular biology software tools, which allow viewing and editing of DNA sequence at the base pair level. These tools need to be interoperable with SBOL annotated data through conversion to GenBank flatfile format. Most importantly there is a free conversion service to and from GenBank flatfile format. The format is commonly used by DNA editor software, such as the application A Plasmid Editor (ApE) which uses GenBank as the native ApE file format. This web service is currently available from JBEI on the web [162]. Thus, users will be able to open, edit, and save SBOL annotated constructs using ApE, or any sequence editor which reads and writes GenBank format files.

Finally, the adoption of SBOL reduces the need for the ubiquitous use of spreadsheets to manage DNA sequence associated information. The use of generic spreadsheet software to manage DNA sequences is sufficient only at small scales and only within a single group or, more likely, a single investigator. Giving synthetic biologists the ability to easily transfer structured information to management software such as Clotho, or JBEI-ICE, provides a method to share the information about the created synthetic constructs. Furthermore, by providing this utility, the process of transfer to and from design tools becomes more automated. Therefore it avoids the need for re-entry of the information through a structured interface. By reducing the barrier to adoption of management software, the use of the SBOL standard also promotes the adoption of improved engineering practices. For example, a tool like Clotho provides utilities for validation and checking of conformance to assembly standards. These benefits are especially valuable in large groups working towards a common engineering goal or between groups. In both situations the downstream recipient relies and must depend on the quality of work of others.

To realize the vision of engineering based on predictable designs subcomponent elements need to have predictable functional properties. To determine the necessary and sufficient conditions for predictability synthetic biology researchers must be able to reproduce the results of research published in scientific literature. The exciting results of biological engineers are published every week, but it is very difficult or impossible to replicate the work. Building new systems directly from the information provided in an article is practically impossible without receiving additional information or even materials from the authors [19]. The SBOL standard provides the first step towards a solution, a set of strict criteria to specify the DNA sequence design. These designs constitute the minimum necessary information to replicate a published design. SBOL does not require the use of sequence annotations or even the inclusion of a DNA sequence, but if they are provided it specifies exactly how they must be structured. These requirements ensure that the receiving party can understand what they receive. For example, recently the BIOFAB began to systematically characterize a large number of professionally designed DNA components. These DNA components are available in SBOL format and therefore compliant software, such as Gene Designer from DNA 2.0, can read the data exactly as intended by the publishers. Going forward, the adoption of SBOL will contribute to making the re-use of DNA components from publications more predictable. To encourage such practice the SBOL community will have to establish relationships with the journals in the field to support and eventually require DNA designs to be submitted to SBOL design repositories with submission for publication.

These benefits can lead to an improved coordination of work among an interconnected network of synthetic biology researchers, companies, and public repositories. The SBOL standard provides the foundation for a computational framework of exchange for synthetic biology. SBOL already supports the representation of a range of design cases, from the theoretical design template to a complete collection of DNA components. Furthermore, the standard can be extended; collaborators are already working on new extensions. The strength of this approach comes from the SBOL community. The members of the group are empowered to develop new solutions when the need arises. Further fostering of this social context can lead to a sustainable growth for the SBOL development strategy in the long term.

6.3 IMPLICATIONS FOR BIOMEDICAL INFORMATICS

The use of SBOL in the synthetic biology community fulfills the promise of knowledge representation in a new domain and presents new opportunities for the development of additional applications and theory. In order to facilitate information exchange, information technology tools are needed to put the SBOL into practice. I took advantage of some of these tools in the development of the libSBOL libraries and the SBPkb. Information about DNA components annotated using the SBOL model can be used by automated tools to enable services such as more accurate search and knowledge management. To achieve such functionality information about components is described or annotated with the SBOL vocabulary and the Sequence Ontology (SO) [99] to give the information explicit meaning, making it easier for machines to automatically process and integrate the information available. This process requires the acquisition of that information, its encoding in SBOL, making it available through the libSBOL API and the SBPkb query interface to developers, and a synthetic biology desktop and web software tools.

The SBOL representation is serialized in RDF/XML syntax; it can be interpreted as an RDF graph. This interpretation of the SBOL file format is possible because the SBOL data model is implemented using W3C Semantic Web standards. Such interpretation allows the use of standard parsers and query languages to access the encoded information. RDF/XML syntax may be parsed directly from the file form, or can be accessed using a query protocol (i.e. SPARQL). Leveraging the RDF/ XML syntax and its interpretation as an RDF graph is a benefit to developers when managing SBOL data. The first benefit is the unique identifier system based on URIs adopted from RDF. It allows SBOL data from two sources to be merged into one dataset without mitigating synonym conflicts. No additional routines need to be written to enable simple integration.

Furthermore, SBOL employs the Sequence Ontology (SO) [99] for DNA component types. The SO extends SBOL to include terms which provide a richer description for the annotation of sequences. The richer vocabulary and semantics enables additional descriptive capabilities within the SBOL semantic framework. The detailed descriptions of sequence will allow for a more informed relationship to other extensions of SBOL. By providing a controlled vocabulary of terms and the relationships between them, the use of SO will provide the foundations for integration, and smart retrieval of DNA components. Additionally, the SO biological region terminology has implications for the potential functional roles of sequence regions. The interpretation depends on SBOL DNA components described using the SO controlled vocabulary and relationships between the ontology’s terms. For example, the relationships between RNA transcript elements and sub-regions, such as the 5’ UTR and coding sequence are specified. Such relationships could be leveraged to perform simple semantic validation of designs. A case where a 5’UTR is not part of a transcript region should raise a flag to the designer.

Additionally, compatibility with the RDF/ XML is advantageous because RDF is a W3C recommended technology for the Semantic Web. SBOL data can be read, manipulated, and interpreted using generic Semantic Web tools. The community of developers who may be able to take advantage of SBOL data is thus greater than the SBOL Developers group. For example, as part of the Semantic Web Health Care and Life Sciences Interest Group [163] there is a burgeoning community of biomedical informaticists using Semantic Web data for medicine and basic research. These researchers will now be able to access the rich source of biological engineering knowledge contained in SBOL encoded designs.

The choice of the Semantic Web technology and standards provides validation and opportunity for research in the biomedical informatics field. The use of a formal representation of human-made living machine designs extends the applications of knowledge representation (KR) to the biomedical domain. This application of KR is immediately useful to scientists in the domain field. The representation of DNA designs such as SBOL enables a novel solution for designing new biological designs through the re-use of components and the information retrieval of candidate components for new designs. These capabilities are further augmented by the use of the Sequence Ontology, an existing informatics resource. The development of SBOL also constitutes a new application of Semantic Web research, enabling the use of one representation for multiple purposes. Finally, the iterative design strategy in concert with my leadership role as a biomedical informatics researcher contributed to the collaboration which resulted in the successful deployment of SBOL. The development of the SBOL community, the specification of the Core, and the deployment I presented in this dissertation are an initial milestone for SBOL. In the next sections I discuss the limitations and the opportunities for future research.

6.1 Limitations and Opportunities for Extensions

In this dissertation I described the current capabilities of SBOL for the transmission of designs in three use cases 1) DNA design template; 2); Annotated DNA sequence and 3) Collection of DNA components. These use cases, derived from the SBOL Developers discussions, are a small set chosen as the most critical to address first. Additionally, the use cases were also only applied in three scenarios, the peer to peer exchange between software tools, the publication of collections of DNA components and use of SBOL for design through informational retrieval. However, alternate scenarios and additional capabilities will be needed to realize a complete information life cycle for synthetic biology.

SBOL serves as an exchange language for the design stage and assembly planning. I did not complete the iterative development and deployment strategy for additional use cases and scenarios. Therefore, SBOL does not yet enable transitions for the complete engineering cycle. One of the critical next steps is further requirements gathering and development for the remaining stages of engineering. For example, the immediate utility of representation of quantitative characterization results of DNA component performance is needed. Work towards this goal is underway within the SBOL Developer group under the leadership of the BIOFAB.

6.1.1 *The need for the representation of function*

The current representation does not include functional information. I did not expand on the formal definition of the biological function of the DNA components. The Sequence Ontology provides definitions for the types of DNA components, but these definitions do not extend to the function of the RNA and proteins which is coded by the DNA. A formal description of function, which can be computationally interpreted, will be needed to improve the specificity of information retrieval, modeling, and testing. For retrieval applications the descriptions will need to express the function in such terms that can be retrieved when design criteria are specified. Mathematical models of the function components will also be necessary to compose new devices and systems from heterologous components. Finally, functional descriptions in terms of such models will be needed to perform the empirical testing in order to validate or invalidate the overall design models.

One of the first needs is to extend the SBOL representation through the addition of the potential functional roles of DNA, RNA, protein, and other molecular components. The scope of this work is particularly important for the description of genetic regulatory networks. For example, the terms and relations needed to connect a promoter and its cognate factors. Related work, such as BioPAX [164, 165] provides an approach to the qualitative representation of gene regulation using OWL and could provide a starting point for the analogous representation for forward engineering. First, we will need to evaluate the BioPAX representation to assess the suitability of re-using it within the SBOL framework. Using this representation we will strive to describe knowledge such as a transcription factor up-regulates or down-regulates the expression of downstream genes. Then, we will choose a suitable existing framework or create a combination based on prior work and the needs presented by our collaborators. The outcome of this potential SBOL extension project will be the functional role representation for SBOL. Those results will be implemented as part of the SBOL representation framework. The terminology and relationships needed to model and construct such networks is well studied and therefore should provide a goal with a scope achievable in a short timeframe.

To facilitate meaningful use of synthetic biology information transferred using the SBOL framework there are multiple possibilities for expanding SBOL. The most significant is the inclusion of functional descriptions. However, there is also a need for the representation of empirically derived quantitative performance data. Such information can be used for parameterization of models. Also a representation of rules and restrictions which aid in design and assembly will be needed for automation of the physical processes involved in the construction of new organisms. Additionally, laboratory information is necessary to realize a fully sufficient representation for the replication and experimental validation of prior work. Finally, new methods will have to be developed to extend retrieval capabilities for design using quantitative specification criteria.

6.1.2 *The need for formal validation*

The main limitation of this work was the lack of validation of the results by a formal study. For example, I did not measure the effect on efficiency for the engineering cycle. Validation of efficiency improvements would benefit synthetic biologists, as it would improve the case to be made to stakeholders in the broader synthetic biology field. In my work I claim that SBOL will improve efficiency through the coordination of work in synthetic biology. I believe the stakeholders involved in the SBOL Developers group are experts in this field and attest to its benefit through their active contributions and adoption of SBOL. Formal studies of improved efficiency can relevant for specific enterprise settings, in contrast to academic research settings.

Another limitation is the lack of formal validation of the necessity of SBOL requirements for reproducibility in the laboratory. Progress towards predictable engineering of biological systems is dependent on reproducibility of synthetic biology research results. A broader goal is to expand SBOL to include information required for the representation to be necessary and sufficient to replicate the design. A part of such work would be to validate whether SBOL includes the minimal information needed to replicate and re-use DNA components. Such a study is a long term research endeavor and would require the participation of a group of stakeholders representative of the diversity of synthetic biology research. SBOL is a bridge that will enable such studies to be performed.

A final limitation is that the ultimate goal of synthetic biologists may be to create algorithms which predict the DNA sequence directly from design requirements. Similarly to the related synthetic biology field of DNA computing, the sequence could be produced entirely by a computational algorithm. When this goal is achieved the current form of SBOL will diminish in utility. The DNA sequence will no longer be the core design element. The specification of DNA components will only be useful to the specialized facility or equipment which converts the functional design to the physical DNA molecules. These limitations offer fertile ground for future research which I describe in the next section.

6.2 FUTURE DIRECTIONS

In the continuation of this work there is a need to extend the knowledge representation framework for the full specification of synthetic biological designs and the engineering process. Current information resources do not offer sufficient computational utilities for the knowledge essential to build and test newly proposed designs. There remains the need to develop a robust semantic information system infrastructure to manage the vast complexity of biological system design and the large number of components throughout the synthetic biology engineering process. To accomplish this, the research and development which resulted in the current SBOL must be extended. Additional modules, SBOL Extensions, must be developed to extend the SBOL Core representation to include information about DNA component performance and host context. Furthermore, SBOL Extensions to define dynamic models, such as are represented in SBML, a human readable script, and a visual representation is needed. Research is needed to design an information system, which would facilitate a complete and rapid engineering of synthetic biological systems. The tools that need to be developed would provide computational access to logically consistent information throughout the entire engineering process. Furthermore, the rapid pace of development in synthetic biology research and enabling biotechnology presents significant challenges for the stability of information system design and the computational tools that support them. To confront these difficulties I espouse the iterative software development strategy driven by collaborators and community feedback for future work. Applying this methodology takes advantage of the fertile environment for collaborative research within the synthetic biology community.

6.2.1 *Extension of the application of Semantic Web technology and standards*

To accomplish the data management and retrieval tasks needed to support the complete information cycle during the engineering process, the SBOL representation needs further development. For example, to improve the quality of information resources available as components for design, a methodology to ensure consistency of the resources is needed. It is not currently possible to retrieve DNA components across three disparate resources (see Chapter 5 data publishers) by specifying terms such as promoter or terminator. Since in SBOL DnaComponent types are required to use terms from the Sequence Ontology, these standardized terms can be used to select components by type in any SBOL resource. For now, the goal is to encourage the use of the SO ontology. Then, as SBOL adoptions grows, to encourage publishers to take advantage of SBOL's extensibility, to add more complete descriptions to better help recipients interpret published collections. However, if two applications provide contradictory information to a local SBOL knowledgebase we will be presented with a challenge to integrate the facts specified by the different. Our efforts to provide access to a repository shared between applications which contain data structures with potentially inconsistent facts would present perplexing or erroneous results.

To address such conflicts and therefore to maintain a well-managed information resource I would leverage the expressiveness of OWL-DL and automated reasoning. OWL-DL offers the ability to apply practical reasoning algorithms needed to determine consistency. The language is an RDF/XML-based serialization and is therefore compatible with the current SBOL format. In addition to the basic subsumption relationship, encoded as the subclass hierarchies, other relationships such as grouping, materialization, and part-whole aggregation can be considered to model knowledge about relationships between DNA components. The specification of formal logic definitions using OWL-DL most importantly supports consistency checking and more sophisticated information retrieval. The automated reasoning capabilities are a characteristic feature of description logics which allows the exploitation of description of the model to draw conclusions about information within a knowledge base.

A portion of validation tasks for integrated data can be accomplished through logical consistency checking by automated reasoning services provided by the Pellet inference engine [166]. The type of validation that we will build into the libSBOL utility is to perform data cleaning [167]. We will be able to remove some random and systematic errors from the SBOL data through filtering, merging, and translation. By applying ontology based methods [168] to identify such conflicts we hope to alleviate a significant amount of time needed for the knowledge discovery process. To test these methods we will assess the data imported from the SBPkb, BacilloBricks repository, and the BIOFAB. We will go through a subset manually and identify the occurrence of validation issues such as unique value violations, contradictory relationships, domain constraint violations, among others [169]. This approach to identification and subsequent removal of these errors will inform design of libSBOL validation utilities and DL axioms. The new data verification functionality will then be applied to improve data comprehensibility of the SBPkb dataset and integration of data from software tools. We would then collaborate with other groups to demonstrate generalizability of the approach to other situations.

I look forward to building on the foundation established by the SBOL framework to support additional capabilities, specifically to take advantage of reasoning services for ontologies formalized in OWL. Semantic Web inference engines, such as Pellet [170], Hermit [171], and Fact++ [172] perform consistency checking and classification/realization. These tools validate and generate new inferences about a set of axioms based on logical constraints and restrictions defined in OWL. Therefore, to develop significant improvements to SBOL, the vocabulary will have to conform with ontology design best practices [149] and be defined using OWL-DL class restrictions. Therefore, to impart these capabilities I plan to formalize SBOL class definitions to make SBOL into an authoritative ontology for synthetic biology.

Synthetic biology research is highly distributed. In the future I envision, not just individual repositories, but a network of repositories. Such repositories may range from those that contain predominately DNA Component designs described in peer reviewed publications, or be a collection of DNA Components professionally fabricated by organizations such as the International Open Facility Advancing Biotechnology (BIOFAB). As long as all these repositories are compatible with SBOL, then researchers can retrieve designs from any selection of these repositories. The SBPkb is the first node in a framework for a semantic web of distributed knowledge in synthetic biology. This vision is a small scale synthetic biology application of the Semantic Web.

6.2.2 *SBOL Extensions*

To aid in the design of transcriptional devices, I will work with the SBOL Developers community to extend SBOL in order to describe information needed for the complete information lifecycle in synthetic biology. For example, to include performance and host context information and to provide utilities for visualization, scripting, and modeling. Some of this work is already underway at the collaborating sites [173].

The SBOL Performance extension could describe how the protein products of DNA components behave. For example, BBa_F2620 a gene circuit that responds to 3-oxo-hexanoyl-HSL ($3OC_6HSL$) could be described in terms of its performance [79]. As Canton et al., propose the downstream gene synthesis rate (i.e. GFP) at different concentrations of inducer could be referred to as its static performance. Its dynamic performance could be measured over a time course to define how the downstream synthesis rate changes over time. Additionally, response to other analogous small molecules inputs can provide compatibility information. Finally, a description of the reliability of the DNA component over generations of the culture can be important measures of performance which can inform future redesign of the DNA component [140].

The SBOL Host context extension could describe the many other biological entity components of a cell and its relevant environment. For example, the host extension could include the information models for cells, proteins, small molecules, and cellular compartment structures. This physical description of the cell's environment would be important for maintaining reliable records of the conditions under which DNA components are put into practice.

The SBOL Modeling extension would serve to represent models such as those currently encoded as SBML. We are working on re-using the SBML representation within SBOL and developing a method for embedding SBOL descriptions into SBML itself. Such an extension could specify how components can be combined together [174] and regulated. For example, to specify the interaction between transcriptional regulatory proteins and their cognate sequences, we could use simplified representation of functional relationships. Towards this goal we plan to leverage related work such as the BioPAX effort (biopax.org) [165], [175] to specify the potential role of a promoter and factor pair, not the mechanism by which it occurs. A qualitative relationship between promoter parts and regulatory proteins will allow us to query and infer intended and unintended interactions. (The ability to carry out such inferences will require the use of a Semantic Web inference system such as Pellet.) For example, an instance of the promoter pLuxR (BBa_R0062) can be annotated as having an activating role on downstream expression in presence of LuxR protein and $3OC_6HSL$. Such a representation of gene regulation information is limited, but forms a framework for regulatory element information retrieval. In general, we aim to expand SBOL so that it can support consistency checking of designs as a way to do initial validation of a design and to help identify possible design problems early in the engineering process.

Work on the SBOL Script and Visual extensions is already in progress. The purpose of these extensions is to re-represent the computational information stored in SBOL format into a human understandable form. For example, each of the Sequence Ontology types used to classify DNA components would have a corresponding visual symbol. Such symbols can then be used in software such as TinkerCell and GenCAD to communicate to the user the type of the component. The SBOL Script extension would be the analogous text based representation which could be used to type the information in shorthand and then have a CAD tool or translation tools re-represent it in SBOL.

We plan to improve and extend SBOL in the near future. Our goal is to re-engineer SBOL into an ontology which supports the forward engineering practice of synthetic biologists. In particular, we aim to include enough information to support consistency checking and design coherence, as described in the discussion section. By automating reasoning, using the semantic definitions of biological components, we aim to provide improved design automation functionality for CAD software, such as TinkerCell. More broadly, we expect to leverage the ability of the OWL language to capture rich semantics, and to support 'intelligent' information retrieval and reasoning capabilities as envisioned by the Semantic Web. This further integration of SBOL with software will help encourage re-use of previously described components, a best practice of synthetic biology.

6.3 FINAL CONCLUSION OF VISION

SBOL, the new data exchange standard is a foundational technology for synthetic biology. This standardization of information exchange project was augmented by the strength of the collaborative community. Its development process relied on the open collaborative model and succeeded in creating a standard which is free for use to both academics and industry. I believe

SBOL will become an integral part of the synthetic biology engineering tool kit. It is a tool which makes engineering more efficient. It will directly contribute to the vision of rationally designing whole new organisms and significantly advance understanding of natural ones.

It is estimated that the value of the global synthetic biology market will grow from $1.6 billion in 2011 to $10.8 billion in 2016 [176]. This explosive growth can only be sustained if the products can be brought to market much faster than the continuing reliance on genetic engineering processes developed in the 1970s. The path forward to an improved drug and therapy development pipeline requires drastic improvements in the tools available to synthetic biologists. SBOL is one such tool which has the potential to help engineers building better, faster, and stronger solutions to the world’s most pressing challenges. For SBOL to become truly useful there is a need for continued development, evolution with the field, and maintenance of the information exchange infrastructure. Biomedical informatics research is necessary to overcome the social, technical, and information challenges ahead.

REFERENCES

1. Thwaites, T., *The Toaster Project: Or a Heroic Attempt to Build A Simple Electric Appliance from Scratch*. 2011, New York City: Palgrave Macmillan.
2. Johnson, I.S., *Human Insulin from Recombinant DNA Technology.* Science, 1983. **219**: p. 632-637.
3. Berg, P., *Co-chairman's remarks: reverse genetics: directed modification of DNA for functional analysis.* Gene, 1993. **135**: p. 261-264.
4. Keasling, J., *The Promise of Synthetic Biology.* The Bridge, 2005. **35**: p. 18-21.
5. Gibson, D.G., et al., *and Cloning of a Mycoplasma genitalium Genome.* Science, 2008. **319**: p. 1215-1220.
6. Peccoud, J., *Gene Synthesis: Methods and Protocols*. Methods in Molecular Biology. Vol. 852. 2012: Humana Press.
7. Khalil, A.S. and J.J. Collins, *Synthetic biology: applications come of age.* Nature reviews. Genetics, 2010. **11**: p. 367-79.
8. Ruder, W.C., *Synthetic Biology Moving into the Clinic.* 2012. **1248**.
9. Endy, D., *Foundations for engineering biology.* Nature, 2005. **438**: p. 449-453.
10. Slusarczyk, A.L., A. Lin, and R. Weiss, *Foundations for the design and implementation of synthetic genetic circuits.* Nature Reviews Genetics, 2012. **13**: p. 406-420.
11. Arkin, A.P., *Setting the standard in synthetic biology.* Assessment, 2008. **26**: p. 771-774.
12. Kelly, J.R., et al., *Measuring the activity of BioBrick promoters using an in vivo reference standard.* Journal of biological engineering, 2009. **3**: p. 1-13.
13. Müller, K.M. and K.M. Arndt, *Chapter 2 Standardization in Synthetic Biology.* 2012. **813**.
14. Shetty, R.P., et al., *Engineering BioBrick vectors from BioBrick parts.* Journal of Biological Engineering, 2008. **2**: p. 5.
15. BBF. *The BioBricks Foundation Technical Program* 2012; Available from: biobricks.org/programs/technical-program/.
16. Fielding, R., et al. *Hypertext Transfer Protocol -- HTTP/1.1*. IETF Request for Comments: 2616 1999; Available from: http://tools.ietf.org/html/rfc2616.
17. Brazma, A., *On the importance of standardisation in life sciences.* Bioinformatics (Oxford, England), 2001. **17**: p. 113-4.
18. Searls, D.B., *The Roots of Bioinformatics.* PLoS Computational Biology, 2010. **6**: p. e1000809.
19. Peccoud, J., et al., *Essential information for synthetic DNA sequences.* Nature Biotechnology, 2011. **29**: p. 22.
20. Hinds, P.J. and S.P. Weisband, *Knowledge Sharing and Shared Understanding in Virtual Teams*, in *Virtual Teams That Work: creating conditions for virtual team effectiveness*, C.B. Gibson and S.G. Cohen, Editors. 2003, Jossey-Bass: San Francisco. p. 21-36.
21. Blum, S.D., *A Call to Revise the TDWD Standards Development Process*, 2000. p. 1-16.
22. Wu, K. and C.V. Rao, *Computational methods in synthetic biology : towards computer-aided part design Kang Wu and Christopher V Rao.* Current Opinion in Chemical Biology, 2012: p. 1-5.
23. Cooling, M.T., et al., *Standard virtual biological parts: a repository of modular modeling components for synthetic biology.* Bioinformatics, 2010. **26**: p. 925-931.

24. Peccoud, J., et al., *Targeted development of registries of biological parts.* PLoS One, 2008. **3**: p. e2671.
25. Grunberg, R. *BBF RFC 30: Draft of an RDF-based framework for the exchange and integration of Synthetic Biology data*. 2009; Available from: http://hdl.handle.net/1721.1/45143.
26. Szybaslki, W. and A. Skalka, *Nobel prizes and restriction enzymes.* Gene, 1978. **4**: p. 181-182.
27. Arkin, A.P., *Synthetic cell biology.* Current opinion in biotechnology, 2001. **12**: p. 638-44.
28. Benner, S.A. and A.M. Sismour, *Synthetic biology.* Nature reviews. Genetics, 2005. **6**: p. 533-543.
29. Elowitz, M.B. and S. Leibler, *A synthetic oscillatory network of transcriptional regulators.* Nature, 2000. **403**: p. 335-338.
30. Gardner, T.S., C.R. Cantor, and J.J. Collins, *Construction of a genetic toggle switch in Escherichia coli.* Nature, 2000. **405**: p. 339-342.
31. Belkin, S., *Microbial whole-cell sensing systems of environmental pollutants.* Current Opinion in Microbiology, 2003. **6**: p. 206-212.
32. Joshi, N., et al., *Novel approaches to biosensors for detection of arsenic in drinking water.* Desalination, 2009. **248**: p. 517-523.
33. Alper, H., K. Miyaoku, and G. Stephanopoulos, *Construction of lycopene-overproducing E. coli strains by combining systematic and combinatorial gene knockout targets.* Nature biotechnology, 2005. **23**: p. 612-616.
34. Lu, T.K. and J.J. Collins, *Dispersing biofilms with engineered enzymatic bacteriophage.* Proceedings of the National Academy of Sciences, 2007. **104**(27): p. 11197.
35. Anderson, J.C., et al., *Environmentally Controlled Invasion of Cancer Cells by Engineered Bacteria.* Journal of Molecular Biology, 2006. **355**(4): p. 619-627.
36. Izallalen, M., et al., *Geobacter sulfurreducens strain engineered for increased rates of respiration.* Metabolic Engineering, 2008.
37. Justin, G.A., *Biofuel cells as a possible power source for implantable electronic devices*, in *Department of Bioengineering*2004, University of Pittsburgh: Pittsburgh.
38. Willner, I., *BIOELECTRONICS: Biomaterials for Sensors, Fuel Cells, and Circuitry.* Science, 2002. **298**(5602): p. 2407-2408.
39. Afonso, B., P.A. Silver, and C.M. Ajo-Franklin, *A synthetic circuit for selectively arresting daughter cells to create aging populations.* Nucl. Acids Res., 2010: p. published online February 11, 2010.
40. Ro, D.-K., et al., *Production of the antimalarial drug precursor artemisinic acid in engineered yeast.* Nature, 2006. **440**: p. 940-943.
41. Aubel, D. and M. Fussenegger, *Mammalian synthetic biology--from tools to therapies.* BioEssays : news and reviews in molecular, cellular and developmental biology, 2010. **32**: p. 332-45.
42. Tigges, M. and M. Fussenegger, *Recent advances in mammalian synthetic biology-design of synthetic transgene control networks.* Current opinion in biotechnology, 2009. **20**: p. 449-60.
43. Basu, S., et al., *A synthetic multicellular system for programmed pattern formation.* Nature, 2005. **434**: p. 1130-4.

44. Purnick, P.E.M. and R. Weiss, *The second wave of synthetic biology: from modules to systems.* Nat Rev Mol Cell Biol, 2009. **10**: p. 410-422.
45. Stricker, J., et al., *A fast, robust and tunable synthetic gene oscillator.* Nature, 2008. **456**: p. 516-9.
46. Hasty, J., et al., *Synthetic Gene Network for Entraining and Amplifying Cellular Oscillations.* Physical Review Letters, 2002. **88**: p. 3-6.
47. Seelig, G., *Enzyme-Free Nucleic Acid.* 2007. **1585**.
48. Qian, L. and E. Winfree, *Scaling up digital circuit computation with DNA strand displacement cascades.* Science, 2011. **332**(6034): p. 1196.
49. Macdonald, J.T., et al., *Integrative Biology Computational design approaches and tools for synthetic biologyw.* 2011: p. 97-108.
50. Endy, D., *Foundations for engineering biology.* Nature, 2005. **438**(7067): p. 449-453.
51. Carlson, R., *The changing economics of DNA synthesis.* Nature Biotechnology, 2009. **27**: p. 1091-1094.
52. Hasty, J., D. McMillen, and J.J. Collins, *Engineered gene circuits.* Nature, 2002. **420**: p. 224-230.
53. Kærn, M., et al., *The engineering of gene regulatory networks.* Annual review of biomedical engineering, 2003. **5**: p. 179-206.
54. Savageau, M.A., *Design principles for elementary gene circuits: Elements, methods, and examples.* Chaos, 2001. **11**: p. 142.
55. Shetty, R.P., D. Endy, and T.F. Knight Jr, *Engineering BioBrick vectors from BioBrick parts.* Journal of Biological Engineering, 2008. **2**: p. 5.
56. Drubin, D.A., J.C. Way, and P.A. Silver, *Designing biological systems.* Genes & Development, 2007. **21**: p. 242-254.
57. Registry. *Registry of Standard Biological Parts*. 2012; Available from: http://partsregistry.org.
58. Knight, T.T., et al. *Idempotent Vector Design for the Standard Assembly of Biobricks*. in *Artificial Intelligence*. 2003.
59. Knight, T.F. *Draft Standard for Biobrick Biological Parts*. BioBricks Foundation Request For Comment (BBF RFC 10) 2007; Available from: http://hdl.handle.net/1721.1/45138.
60. GinkgoBioworks. *BioBrick Assembly Manual*. 2009; Available from: http://ginkgobioworks.com/support.
61. RAE, *Synthetic Biology: scope, applications and implications*, 2009, The Royal Academy of Engineering: London.
62. Clancy, K. and C.A. Voigt, *Programming cells: towards an automated 'Genetic Compiler'.* Current Opinion in Biotechnology, 2010. **21**: p. 1-10.
63. Kahn, H.J., *[Invited Tutorial] EDIF Version 350 / 400 and Information Modelling.* 1995: p. 2010-2010.
64. Chandran, D., F.T. Bergmann, and H.M. Sauro, *TinkerCell: modular CAD tool for synthetic biology.* Journal of Biological Engineering, 2009. **3**.
65. Anderson, C., et al. *BBF RFC 0: Instructions to BBF RFC Authors*. in *October*. 2008.
66. Davis, M.W. *A Plasmid Editor*. 2009 February 20, 2010]; Available from: http://www.biology.utah.edu/jorgensen/wayned/ape/.
67. Invitrogen, *Vector NTI Advance™*, 2010, Life Technologies: Carlsbad, CA.

68. INSDC. *The Feature Table*. 2009; Available from: http://www.ncbi.nlm.nih.gov/collab/FT/.
69. Benson, D.A., et al., *GenBank.* Nucl. Acids Res., 2008. **36**(suppl_1): p. D25-30.
70. Densmore, D., et al., *A platform-based design environment for synthetic biological systems*, in *The Fifth Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations*2009. p. 24-29.
71. Bilitchenko, L., et al., *Eugene - A domain specific language for specifying and constraining synthetic biological parts, devices, and systems.* PLoS one, 2011. **6**: p. e18882.
72. Hillson, N.J., R.D. Rosengarten, and J.D. Keasling, *j5 DNA assembly design automation software.* ACS Synthetic Biology, 2012.
73. Bartley, B., 2010: Seattle, WA.
74. Campbell, E.G., *Data Withholding in Academic Genetics: Evidence From a National Survey.* JAMA: The Journal of the American Medical Association, 2008. **287**: p. 473-480.
75. Lizarazo, M. *Judging/Judging Criteria*. iGEM 2008 2008 November 17, 2008]; Available from: http://2008.igem.org/Judging/Judging_Criteria.
76. Brown, J., *The iGEM competition : building with biology.* Synthetic Biology, IET, 2005. **1**: p. 3-6.
77. Registry. *Registry of Standard Biological Parts*. 2010 Feb 29, 2010]; Available from: http://parts.mit.edu.
78. Ham, T.S., et al., *Design , implementation and practice of JBEI-ICE : an open source biological part registry platform and tools.* 2012: p. 1-8.
79. Canton, B., A. Labno, and D. Endy, *Refinement and standardization of synthetic biological parts and devices.* Nature Biotechnology, 2008. **26**: p. 787-793.
80. Berners-Lee, T., J. Hendler, and O. Lassila, *The semantic web.* Scientific American, 2001. **284**(5): p. 34-43.
81. W3C. *Semantic Web Standards Wiki*. 2012; Available from: http://www.w3.org/2001/sw/wiki/Main_Page.
82. W3C. *Semantic Web*. 2012; Available from: http://www.w3.org/standards/semanticweb/.
83. Gennari, J.H., et al., *The evolution of Protégé: an environment for knowledge-based systems development.* International Journal of Human-Computer Studies, 2003. **58**(1): p. 89-123.
84. Krech, D. *RDFLib*. 2010 2009 Apr 20]; Available from: http://www.rdflib.net.
85. Broekstra, J., A. Kampman, and F. Van Harmelen. *Sesame: A generic architecture for storing and querying rdf and rdf schema*. in *Proc of the 1st Int'l Semantic Web Conference (ISWC 2002)*. 2002. Sardinia, Italia: Springer.
86. Berners-Lee, T. *Linked Data*. Design Issues 2006; Available from: http://www.w3.org/DesignIssues/LinkedData.
87. W3C. *LinkingOpenData W3C SWEO Community Project*. 2012; Available from: http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.
88. Feigenbaum, L., et al., *The semantic web in action.* Scientific American Magazine, 2007. **297**(6): p. 90-97.
89. Macmanus, R. *How Best Buy is Using The Semantic Web*. ReadWriteWeb 2010; Available from: http://www.readwriteweb.com/archives/how_best_buy_is_using_the_semantic_web.php.

90. Gudivada, R.C., et al., *Identifying disease-causal genes using Semantic Web-based representation of integrated genomic and phenomic knowledge.* Journal of biomedical informatics, 2008. **41**(5): p. 717-729.

91. Anwar, N. and E. Hunt, *Francisella tularensis novicida proteomic and transcriptomic data integration and annotation based on semantic web technologies.* BMC Bioinformatics, 2009. **10**(Suppl 10): p. S3.

92. Roldán-García, M., et al., *KA-SB: from data integration to large scale reasoning.* BMC Bioinformatics, 2009. **10**(Suppl 10): p. S5.

93. Zheng, G. and A. Bouguettaya, *Service-based analysis of biological pathways.* BMC Bioinformatics, 2009. **10**(Suppl 10): p. S6.

94. Dietze, H. and M. Schroeder, *GoWeb: a semantic search engine for the life science web.* BMC Bioinformatics, 2009. **10**(Suppl 10): p. S7.

95. Lamprecht, A.L., T. Margaria, and B. Steffen, *Bio-jETI: a framework for semantics-based service composition.* BMC Bioinformatics, 2009. **10**(Suppl 10): p. S8.

96. Sutherland, K., et al., *Knowledge-driven enhancements for task composition in bioinformatics.* BMC Bioinformatics, 2009. **10**(Suppl 10): p. S12.

97. Samwald, M., et al., *Linked open drug data for pharmaceutical research and development.* Journal of cheminformatics, 2011. **3**(1): p. 19.

98. Jentzsch, A., et al., *Linking open drug data.* Proceedings of the Second Triplification Challenge 2009, Graz, Austria, 2009.

99. Eilbeck, K., et al., *The Sequence Ontology: a tool for the unification of genome annotations.* Genome biology, 2005. **6**(5): p. R44.

100. Davis, F.D. and B.F.D. Davis, *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology.* MIS Quarterly, 1989. **13**: p. 319-340.

101. Yasnoff, W., P. O'Carroll, and A. Friede, *Public health informatics and the health information infrastructure*, in *Biomedical informatics: computer applications in health care and biomedicine*, E.H. Shortliffe and J.L. Cimino, Editors. 2006, Springer: New York. p. 537-63.

102. Leonard, K.J., *Critical Success Factors Relating to Healthcare ' s Adoption of New Technology : A Guide to Increasing the Likelihood of Successful Implementation.* Electronic Healthcare, 2004. **2**: p. 72-81.

103. Katz, E. and P.F. Lazarsfeld, *Personal influence: The part played by people in the flow of mass communications*. 2006: Transaction Pub.

104. RCSB. *Protein Data Bank (PDB) - An Information Portal to Biological Macromolecular Structures*

2012; Available from: http://www.pdb.org.

105. Berman, H.M., *The Protein Data Bank: a historical perspective.* Acta crystallographica. Section A, Foundations of crystallography, 2008. **64**: p. 88-95.

106. Benson, D.a., et al., *GenBank.* Nucleic acids research, 2006. **34**: p. D16-20.

107. Brazma, A., et al., *Minimum information about a microarray experiment (MIAME)-toward standards for microarray data.* Nature genetics, 2001. **29**: p. 365-371.

108. Edgar, R., M. Domrachev, and A.E. Lash, *Gene Expression Omnibus: NCBI gene expression and hybridization array data repository.* Nucleic Acids Research, 2002. **30**: p. 207-210.

109. Barrett, T., et al., *NCBI GEO: archive for high-throughput functional genomic data.* Nucleic acids research, 2009. **37**: p. D885-90.

110. Parkinson, H., et al., *ArrayExpress update--from an archive of functional genomics experiments to the atlas of gene expression.* Nucleic Acids Research, 2009. **37**: p. D868-72.

111. Ikeo, K., et al., *CIBEX: center for information biology gene expression database.* Comptes rendus biologies, 2003. **326**: p. 1079-82.

112. Ball, C.A., et al., *Standards for microarray data.* Science, 2002. **298**: p. 539.

113. SBML. *Software Guide*. 2011; Available from: http://sbml.org/SBML_Software_Guide.

114. Calvert, J., *Ownership and sharing in synthetic biology: A 'diverse ecology' of the open and the proprietary?* BioSocieties, 2012: p. 1-19.

115. Deshpande, A. and D. Riehle, *The Total Growth of Open Source*, in *Proceedings of the Fourth Conference on Open Source Systems*2008, Springer Verlag.

116. Hammond, W.E., *Downloaded from content.healthaffairs.org by Health Affairs on May 22, 2012 at UNIV OF WASHINGTON SCHOOL.* 2005. **5**: p. 1205-1213.

117. Galdzicki, M., et al. *BBF RFC 31: Provisional BioBrick Language (PoBoL)*. 2009; Available from: http://hdl.handle.net/1721.1/45537.

118. Howell, J.M. and C.A. Higgins, *Champions of Technological Innovation.* Administrative Science Quarterly, 1990. **35**: p. 317-341.

119. Ash, J., *Organizational factors that influence information technology diffusion in academic health sciences centers.* Journal of the American Medical Informatics Association: JAMIA, 1997. **4**: p. 102-11.

120. Galdzicki, M., et al., *Synthetic Biology Open Language (SBOL) Version 1.0.0*, 2011. p. 1-27.

121. Field, D., et al., *'Omics Data Sharing.* Science, 2009. **326**: p. 234-236.

122. Galdzicki, M., et al., *Synthetic Biology Open Language (SBOL) Version 1.1.0*, in *BBF RFC #87*2012. p. 1-26.

123. *BIOFAB Electronic Datasheets*. 2012; Available from: http://biofab.org/data.

124. Madsen, C., et al., *Design and Test of Genetic Circuits using iBioSim.* Design & Test of Computers, IEEE, 2011(99): p. 1-1.

125. Myers, C.J., et al., *iBioSim: a tool for the analysis and design of genetic circuits.* Bioinformatics, 2009. **25**(21): p. 2848-2849.

126. Beal, J., T. Lu, and R. Weiss, *Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks.* PLoS One, 2011. **6**(8): p. e22490.

127. Ham, T.S., et al., *Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools.* Nucleic acids research, 2012.

128. *Clotho*. 2012; Available from: http://clothocad.org.

129. Cai, Y., M.L. Wilson, and J. Peccoud, *GenoCAD for iGEM: a grammatical approach to the design of standard-compliant constructs.* Nucleic Acids Research, 2010. **38**: p. 2637-2644.

130. Villalobos, A., et al., *Gene Designer: a synthetic biology tool for constructing artificial DNA segments.* BMC Bioinformatics, 2006. **7**(1): p. 285.

131. Villalobos, A., M. Welch, and J. Minshull, *In silico design of functional DNA constructs.* Methods in molecular biology (Clifton, NJ), 2012. **852**: p. 197.

132. SBOL. *Team - Synthetic Biology Open Language*. 2012; Available from: http://sbolstandard.org/.

133. Nandagopal, N. and M.B. Elowitz, *Synthetic biology: integrated gene circuits.* Science, 2011. **333**(6047): p. 1244-1248.
134. Mungall, C.J., C. Batchelor, and K. Eilbeck, *Evolution of the Sequence Ontology terms and relationships.* Journal of biomedical informatics, 2010.
135. Bray, T., et al. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. in *W3C Recommendation*. 2008. Cambridge, MA.
136. Beckett, D. *RDF/XML Syntax Specification (Revised)*. in *W3C Recommendation*. 2004.
137. OMG. *UML Version 2.0*. 2005; Available from: http://www.omg.org/spec/UML/2.0/.
138. Berners-Lee, T., R. Fielding, and L. Masinter, *RFC 2396-uniform resource identifiers (URI)*, 1998, IETF RFC, August 1998. http://www.ietf.org/rfc/rfc2396.txt.
139. Tennison, J., *RDF and XML Q&A: Which should I use?* Jeni's Musings, 2008: p. 1-2.
140. Sleight, S.C., et al., *Designing and engineering evolutionary robust genetic circuits.* Journal of Biological Engineering, 2010. **4**(1): p. 1-20.
141. Johnson, J., M. Galdzicki, and H.M. Sauro, *Standard for the Electronic Distribution of SBOLv Diagrams.* 2010.
142. McEntyre, J. and J. Ostell, *The NCBI handbook.* 2002.
143. Hillson, N.J., R.D. Rosengarten, and J.D. Keasling, *j5 DNA Assembly Design Automation Software.* ACS Synthetic Biology, 2011. **1**: p. 14-21.
144. Galdzicki, M., et al. *Provisional BioBrick Language (PoBoL)*. BioBricks Foundation Request For Comment (BBF RFC 31) 2009 2010 Aug 20]; Available from: http://hdl.handle.net/1721.1/45537.
145. *Protégé* 2003 2010 Aug 20]; 89-123]. Available from: http://protege.stanford.edu/.
146. Eilbeck, K., et al., *The Sequence Ontology: a tool for the unification of genome annotations.* Genome Biol, 2005. **6**(5): p. R44-R44.
147. Galdzicki, M., et al., *Standard Biological Parts Knowledgebase.* PLoS ONE, 2011. **6**: p. e17005.
148. BIOFAB, *Data Access Web Service.* 2011.
149. Smith, B., et al., *The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration.* Nature biotechnology, 2007. **25**: p. 1251-1255.
150. Gibson, D.G., et al., *Enzymatic assembly of DNA molecules up to several hundred kilobases.* Nat Methods, 2009. **6**: p. 12-16.
151. Hearst, M.A., *Search user interfaces.* 2009: p. 404.
152. Gruber, T.M. and C.a. Gross, *Multiple sigma subunits and the partitioning of bacterial transcription space.* Annual review of microbiology, 2003. **57**: p. 441-66.
153. Barkai, N. and S. Leibler, *Biological rhythms: Circadian clocks limited by noise.* Nature, 2000. **403**: p. 267-268.
154. Vilar, J.M.G.J.M.G., et al., *Mechanisms of noise-resistance in genetic oscillators.* Proc Natl Acad Sci U S A, 2002. **99**: p. 5988-5992.
155. Johnson, J., H.M. Sauro, and D. Chandran, *WikiDust : a TinkerCell Plugin to Annotate and Share Network Models*, 2010. p. 1-2.
156. Prud'hommeaux, E. and A. Seaborne. *SPARQL Query Language for RDF*. W3C Recommendation 2008; Available from: http://www.w3.org/TR/rdf-sparql-query/.
157. Kelly, J.R., et al., *Measuring the activity of BioBrick promoters using an in vivo reference standard.* Journal of biological engineering, 2009. **3**: p. 4.
158. Chandran, D., F.T. Bergmann, and H.M. Sauro, *Computer-aided design of biological circuits using TinkerCell.* Bioeng Bugs, 2010. **1**: p. 276-283.

159. Hillson, N., *Personal Communication: VectorEditor v1.6.9 now supports SBOL XML import/export*, M. Galdzicki, Editor 2012.

160. Platt, D., *Personal Communication: Thanks! [re: San Francisco SBOL meeting]*, 2012: sbol-dev@googlegroups.com.

161. SBOL. *Publications Citing SBOL - Publications*. 2012; Available from: http://www.sbolstandard.org/publications.

162. Hillson, N. *j5 SBOL XML <-> GenBank conversion utility*. 2012; Available from: http://j5.jbei.org/bin/sbol_converter_entry_form.pl.

163. W3C. *Semantic Web Health Care and Life Sciences (HCLS) Interest Group*. 2012; Available from: http://www.w3.org/blog/hcls/.

164. Bader, G., Demir, E. *BioPAX - Biological Pathways Exchange Language*. 2008; Available from: http://www.biopax.org/.

165. Luciano, J. and R. Stevens, *e-Science and biological pathway semantics.* BMC Bioinformatics, 2007. **8**(Suppl 3): p. S3.

166. Sirin, E., et al., *Pellet: A practical owl-dl reasoner.* Web Semantics: science, services and agents on the World Wide Web, 2007. **5**(2): p. 51-53.

167. Wang, X., H.J. Hamilton, and Y. Bither, *An ontology-based approach to data cleaning.* Regina: Dept. of Computer Science, University of Regina, 2005.

168. Oliveira, P., F. Rodrigues, and P. Henriques. *An ontology-based approach for data cleaning*. in *11th International Conference on Information Quality*. 2006. MIT, Boston.

169. Fürber, C. and M. Hepp, *Using SPARQL and SPIN for Data Quality Management on the Semantic Web*, in *Business Informations Systems (BIS 2010)*2010: Berlin.

170. Sirin, E., et al., *Pellet: A practical owl-dl reasoner.* Web Semantics: science, services and agents on the World Wide Web, 2007. **5**: p. 51-53.

171. Motik, B., R. Shearer, and I. Horrocks, *Hypertableau reasoning for description logics.* J Artif Intell Res, 2009. **173**: p. 1275-1309.

172. Tsarkov, D. and I. Horrocks, *FaCT++ description logic reasoner: System description*, in *Proc of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)*, U. Furbach and N. Shankar, Editors. 2006. p. 292-297.

173. SBOL. *Extensions*. 2012; Available from: http://www.sbolstandard.org/specification/extensions.

174. Cai, Y., et al., *Modeling Structure-Function Relationships in Synthetic DNA Sequences using Attribute Grammars.* PLoS Comput Biol, 2009. **5**: p. e1000529.

175. Sahoo, S.S., et al., *An ontology-driven semantic mashup of gene and biological pathway information: application to the domain of nicotine dependence.* J Biomed Inform, 2008. **41**: p. 752-765.

176. Bergin, J., *Synthetic Biology: Emerging Global Markets.* BCC Research, Wellesley, MA, USA, 2009.