A THREE VIEWPOINT MODEL FOR SOFTWARE ECOSYSTEM

John D. McGregor, J. Yates Monteith
Clemson University
Clemson, SC, USA
{johnmc, jymonte}@clemson.edu

ABSTRACT
An organization that produces software-intensive products participates in a network of relationships with other organizations that supply needed assets, that consume the organization’s products, and that compete for the same customers. Strategic decision makers in that organization must understand these relationships to make optimal decisions. The modeling approach presented in this paper creates three viewpoints on the network to separate concerns and reduce complexity for decision makers. A case study illustrates the effectiveness of the 3 viewpoints and shows the results of applying analyses in the context of those views.

KEY WORDS
Ecosystem; Software Ecosystem; Business Ecosystem; Metamodel

1. Introduction
Strategic software engineering supports the executive decision makers in an organization by providing technical information about the business and engineering aspects of the organization’s products. One of those important aspects is the ecosystem in which an organization operates. The ecosystem comprises the organizations that supply needed assets, that consume the organization’s products, and that compete for the same customers [23]. The ecosystem also includes the software products that are produced by these organizations and that have interdependencies with other software products. When the interrelationships are so complex that they obscure the total effects of a decision, a model of the ecosystem can reduce that complexity to manageable levels.

Our experience with modeling numerous ecosystems in a variety of domains led to the observation that most existing modeling techniques were not sufficient because they addressed only a single dimension, either the organizations or the software-intensive products. Our important questions involve more than just business or just software but some combination of both. For example, a strategic decision maker may be interested in the differences between the software architectures used by organizations with different business models, such as open source or closed source. It is from these combinations that many interesting new ideas emerge.

Different domains present different opportunities. Commercial organizations are making decisions about investing in new products or new features while a government organization often seeks to influence development in a significant portion of the ecosystem in particular directions. A useful model reflects the needs of decision makers in that domain. Much of the value of that model is in the relationships it captures. Relationships among organizations, relationships between organizations and software products, and relationships among the software products provide a rich variety of information useful to decision makers such as who among my suppliers also supplies my competitors.

The contribution of this work is an approach to ecosystem modeling that provides multiple domains of analysis, using viewpoints to tailor knowledge to the needs of a specific stakeholder. The views, instantiated from different viewpoints, are bound together with the relationships between model elements such as a software product being mapped to the organization that produces and/or sells the product.

In the next section we provide information that may be helpful in understanding the main theme of the paper. Then we present the three viewpoints used to model the ecosystem followed by a description of our modeling technique. In section 5 we apply this ecosystem modeling technique to an international research project spanning multiple domains and perform an example analysis on the model. In section 6 we evaluate the modeling technique and in section 7 we compare our work to that of others and then we conclude with a summary of results and a discussion of future directions.

2. Background

2.1 Ecosystems
An ecosystem is “a system formed by the interaction of a community of organisms with their environment” [21]. The term most often refers to a context in which the organisms are living entities such as plants or animals. The ecosystem in which an animal thrives is believed to contain the elements needed to sustain life. The use of the term ecosystem in our context replaces the living organisms with entities such as commercial enterprises, government agencies, and others.

Numerous definitions of software ecosystems are available in the literature [9, 7]. Each of these definitions
replaces the term organism with some form of software-related entity, whether it be a software product or an organization that produces software. Many researchers define a software ecosystem as revolving around a software platform available to all members of the ecosystem [23].

While other authors have provided significant work in modeling ecosystems [4, 5, 6, 7, 9] centered around a platform product, we take a more general view of a platform to be a basis for collaboration such as the core asset base of a software product line. Furthermore, we conduct our analysis focusing on an arbitrary entity in the ecosystem which we refer to as the “hub” of the ecosystem and which may not be the owner of any platform.

2.2 Meta-Modeling

Meta-modeling is the area of modeling that pertains to creating a structure for the creation of models. While a model contains entities that are connected through relationships, a meta-model contains constraints on how entities can be connected to each other through relationships. Rather than specifying what each entity or relationship is, a meta-model provides a mechanism for describing the behavior of each element that exists within a model. The power in utilizing a meta-model comes from the ability to constrain how model elements are composed in addition to the ability of other modelers to extend the meta-model to represent more complex scenarios and situations.

Meta-models have been used across disciplines in software engineering. Notable meta-models include ISO/IEC 24744 for defining development processes [24], the Object Management Group’s Meta-Object Facility [25], and the Object Management Group’s Software Process Engineering Framework [26], among others.

In order to facilitate study and evaluation of our approach, we have created a meta-model, shown in Figure 2, which captures many of the ecosystem relationships and exposes them for analysis. Using the meta-model we create a model that describes the ecosystem of interest. Data captured in the model is used to derive supporting information for strategic decisions. Because of their interplay we include both the concept of an ecosystem and an ecosystem model in the meta-model.

2.3 Models, Viewpoints and Views

Models are intended as abstractions that communicate essential information while eliminating unnecessary details. For complex systems these models can become complex themselves. As a method for managing the complexity of these models, we have adapted the concept of architectural viewpoints presented in [24]. For our purposes, a model expresses an ecosystem of interest, identifying a series of stakeholders and concerns and contains at least one viewpoint from which a view is instantiated. A viewpoint is a part of the ecosystem model that frames the concerns of one or more stakeholders and governs a set of views. A view frames one or more stakeholder concerns, is governed by exactly one viewpoint and composes at least one kind of model element. One may liken this deconstruction through a programmatic notion. The model is the programming language, the viewpoint is a type, and a view is an instantiation of that type. For information, see ISO/IEC 42010 [23].

For the purpose of ecosystem modeling, a view is a subset of the information from a model that presents the entities and relationships of a specific aspect of the model. By decomposing a model into views, a more manageable representation for examining the model is provided while allowing the modelers to construct each view to target a specific set of stakeholders.

Views have been widely used in the software architecture community as a method of delivering the appropriate set of architecture documentation to the right stakeholder. Architectures contain a significant number of entities that document the information flows, software dependencies, hardware dependencies, connections between elements and more. The focused scope of a view limits the set of analytic techniques that need to be applied in that context. [15, 16, 17].

The notion of a view of an ecosystem is useful for multiple reasons. As mentioned in the introduction, the ecosystem contains elements of both the business organization and the software products. While these two notions are related, the business side of the ecosystem often differs based on the market and domain of the business. By dividing our representation of the ecosystem into its constituent parts and providing relations between the views, our model and documentation of the ecosystem is still intact, but can be more accurately delivered to the stakeholders that can best use each view. Furthermore, ecosystems often encompass very large data sets, and, by presenting the ecosystem through a set of views, we make understanding these models more manageable.

These definitions facilitate defining methods for modeling. In the next section we will define the three viewpoints – business, software, and innovation - that we use for each ecosystem model.

3. Three Viewpoints of Software Ecosystems

Our early ecosystem models many mixed types of entities and relationships. Each manager who used the model was only interested in a subset of the model and was often confused by the rest of the model. We made the model more complete modular by taking the three different types of ecosystems in the literature – business, software, and innovation – and, through analysis of each ecosystem, making a separate viewpoint for each one [1, 2, 8]. Each viewpoint targets a set of stakeholder perspectives based on each of the types of ecosystems. The terms are still evolving and there are multiple names and definitions for each “type” of ecosystem. We will avoid some of this confusion by describing the viewpoints and their characteristics, remembering that our primary purpose is to model reality but at a manageable level.

Each ecosystem model consists primarily of entities and relationships covering business or software entities within a
specified domain. Each ecosystem model is tailored for a specific investigation but there are standard elements in each viewpoint. Examples of standard features include domain standards, risks, and threats. Features such as security classifications are useful in some domains but not others. Each view is based on a specific viewpoint whose content is specific to the type of ecosystem.

The three viewpoints are not disjoint. For example a software view contains software products each of which is produced by an organization in the business view and there is a “produces” relationship between the entities in the two views. Most of the relationships are bi-directional such as “produced by” as the other perspective on “produces.”

### 3.1 Business Viewpoint

A business ecosystem is defined as the “loose networks of suppliers, distributors, outsourcing firms, makers of related products or services, technology providers, and a host of other organizations” [2]. Our business viewpoint on an ecosystem includes all types of organizations and the relationships among them. We say “among them” because in ecosystem analysis some of the most interesting relationships exist between suppliers rather than between a supplier and the hub of the analysis. This information is not always available but any that is available makes for a much more insightful analysis.

The business ecosystem is modeled through the use of two tools: Porter’s Five Forces for Strategy Development [18] and Transaction/Transfer Analysis. Porter’s model includes the five competitive forces, shown in Figure 1, that guide the development of a comprehensive business strategy. The business viewpoint uses these categories to identify organizations that play each of the roles. This model includes organizations that may not have a direct transactional relationship with the hub and its buyers and suppliers, but rather has an indirect influence, e.g. competitors.

The Transaction/Transfer Analysis is used to develop models of the supply chains and value chains found in business ecosystems. In Baldwin’s work, transfers are seen as the no-cost exchange of goods, services or information, while transactions are transfer for which a cost is associated [20]. Transfers can be used to model the use of open source software, but we often use the transaction relationship since the total cost of ownership of open source software is non-zero.

By synthesizing these business analysis techniques, we are able to provide a more in depth examination of the business ecosystem than through any single method alone. By applying Porter’s force analysis, we may strategically evaluate the organizations related via Baldwin’s transfer and transaction relations.

### 3.2 Software Viewpoint

A software ecosystem consists of the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions [8]. Our software view represents general uses relationships between software entities. For this high-level view we do not need to know about lower level relationships such as generalization or aggregation. The two dependencies of importance are “shared uses,” where component A calls component B but other components could also call component B, and “exclusive uses” where component A essentially contains component B and no other component has access to it. In many cases we cannot know for certain about this relationship and so the generalized “uses” is also provided in the meta-model.

The software architecture is often a major structuring element in this viewpoint. This is particularly true when much of the software complements a central platform and must obey the constraints of the platform. Standard interfaces are also a major element in the viewpoint since they define and constrain the relationships with compatible components.

The power of the viewpoint construct can be seen in a recent experience [22]. We modeled an ecosystem in which there were isolated clusters of software with no direct relationship to each other. There were connections among all of the organizations in the ecosystem. We created a separate software view from the software viewpoint to model each of the software clusters. We were able to show the disjointedness of the software elements and contrast that to the continuity of the business relationships.

### 3.3 Innovation Viewpoint

The innovation ecosystem is the set of organizations and products that are needed for innovation to be successful [1]. One of the relationships in this view is the link between an anticipated innovation and those organizations and products which must be aligned for the innovation to occur. For example, an innovative plug-in may require a change in the platform at the next release before it can be deployed. A link between the platform and the software plug-in in the software view indicates the dependency, but one that can be realized only after the new release. There is also a link to the organization that created the plug-in within the business view from the node representing the innovation.
The process for creating this viewpoint is less well-defined in terms of a definitive set of steps and requires stakeholder interviews and perspectives. However, Business Week has identified four types of innovation: product innovation, process innovation, customer experience innovation, and business model innovation [19]. These categories guide the search for innovations. Each innovation is then described using the categories of Strategy, Process, Climate, Structure, and Competency.

4. Meta-model and Modeling Process

4.1 Meta-Model

The ecosystem meta-model combines two types of information – domain and modeling. The View stereotypes are model management entities used to form groupings in the model based on our viewpoints. The Organization, Software Product, and Innovation stereotypes denote the actual content being modeled. We have kept the model simple to allow the modelers of actual ecosystems as much freedom as possible. It is also our expectation that the meta-model will continue to evolve. Due to limitations of space we have collapsed the nodes in the model so that their attributes are not visible.

4.2 Modeling Process

Creating the model of the ecosystem that surrounds a hub organization follows a broad brush process but remains a creative activity that addresses the unique aspects of each ecosystem. The process tasks cut across all of the entity types:

1. Gather information – The information needed for the model is scattered and in some cases has never been assembled before. Some of the information is proprietary and the model may become a protected work product that few get to see.

2. Identify entity and relationship types – Build the entity types in the network using the meta-model. Create a notional view of the ecosystem in which each type is represented by a single node and only the general shape of the ecosystem is captured.

3. Capture actual entities and relationships – Use the modeling tool to create both the notional model and an exact model. Modeling the individual entities can require large data modeling tools such as Gephi [10]. In one recent model one entity type alone had approximately 10,000 unique entities [22].

4. Conduct analyses to answer questions – Once the model is populated, analyses identify clusters of entities that constitute the supply chain for each software product and link an innovation to its sources. To conduct the analyses it is often necessary to create sufficient instances of the entity types to represent the actual ecosystem.

These four steps summarize the steps of the STRategic Ecosystem Analysis Method (STREAM), explained in [22].
Table 1: Use Cases for Ecosystem Analysis

<table>
<thead>
<tr>
<th>Actor</th>
<th>Use Case</th>
<th>Viewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineer</td>
<td>Analyze single points of failure within software supply chain</td>
<td>Software</td>
</tr>
<tr>
<td>Technical Manager</td>
<td>Analyze trade-offs in selecting from competing suppliers for software components</td>
<td>Software</td>
</tr>
<tr>
<td>Technical Manager</td>
<td>Analyze feature addition opportunities from existing complementors and competitors</td>
<td>Business</td>
</tr>
<tr>
<td>Strategic Manager</td>
<td>Strategic acquisition and licensing planning</td>
<td>Business</td>
</tr>
<tr>
<td>Strategic Manager</td>
<td>Entry barrier analysis</td>
<td>Business</td>
</tr>
<tr>
<td>Strategic Manager</td>
<td>Ensure conformity between business model and existing innovations within ecosystem</td>
<td>Innovation</td>
</tr>
</tbody>
</table>

4.3 Analysis Use Cases

Table 1 contains a generic set of analyses, stated as use cases for the ecosystem model, that can be performed on the ecosystem utilizing different viewpoints from the perspective of three stakeholders used in section 4.2: an engineer, a technical manager and a strategic manager. We will illustrate these in a case study in the next section.

5. Case Study: The Leducq Single Ventricle Modeling Project

To illustrate the three-viewpoint modeling technique for describing ecosystems, we present a case study based on the research and development community surrounding adolescent congenital cardiac conditions, the Leducq Trans-Atlantic Single Ventricle Modeling (SVM) project.

SVM is a five year interdisciplinary project funded by the Leducq Foundation that seeks to determine new and better methods in supporting clinical decision making when treating children and adolescents who suffer from hypoplastic left heart syndrome (HLHS) [12]. One of the end goals of this project is to develop a product line of surgical planning assistants encompassing products that enable more sophisticated clinical support for determining which procedures patients require to treat HLHS. In addition to the release of a product line of tools, the SVM project is actively engaging in innovating improvements to the domain of biomodeling tools. The complexity of interactions and its blend of activities make the SVM project a good candidate for representation using our model.

The SVM cluster of research institutions will be the hub of our analysis. This cluster is a good starting point for analysis given that it encompasses two computing platforms from which products are constructed. The SVM researchers are creating computational models used for modeling human physiology to predict surgical outcomes (SimVascular platform) and for educating interested users (Drupal/C++ platform). The ecosystem model is shown in Figure 3. This model only shows the entities and relationships within the SVM project. It does not show competing and alternative clusters of organization and products within the same domain. Entity definitions include the stereotypes that map the model to the ecosystem meta-model.

Figure 3. SVM Partial Ecosystem Model
5.1 Business View

We reviewed an extensive set of papers and ecosystem documents, in addition to conducting stakeholder interviews in order to obtain the analysis in this section; however, providing a detailed analysis of the conclusions drawn in this section is beyond the scope of this paper. To construct the business view we first analyze the ecosystem using Porter’s 5 Forces:

- **Suppliers** – The SVM ecosystem includes the SimVascular cardiovascular simulation environment supplied by Simtk.org, an open source project with a large number of participating organizations [12]. Three libraries are supplied by commercial software vendors and numerous open-source plug-ins provided by users. The C/C++ and FORTRAN95 compilers are required by SimVascular and are acquired from Intel. Open source projects provide the Drupal content management system [14] and MySQL, the database behind the web site.

- ** Buyers** – Currently the buyers in this ecosystem are largely users of the research information although there are several commercialization possibilities from the research. The SVM research group hosts a conference each year and also presents at other conferences in order to present results and to encourage others to participate and use the research.

- **Substitutes** – There are open source projects that provide functionality similar to SimVascular, but none could be directly substituted for SimVascular due to missing features and non-standard interfaces. In an open source environment more users of the products leads to a more vibrant community, giving better access to contributions and support. There are a large number of substitutes for the content management and database systems. One interesting note: the open source community is working to produce substitutes for some of the mathematical libraries used in the project.

- **Competitors** – There are other research groups that compete for grant dollars, open source contributors, and commercial groups that provide nearly comparable software. For a more in depth analysis, competitors could be identified by noting the other attendees at conferences giving related presentations as well as those publishing similar studies.

- **Potential entrants** – We are unable to identify any potential entrant companies or products.

There are several transactional relationships that embody the supply chains among the organizations in the ecosystem as shown in the graph representation in Figure 4a. We show that the Leducq Foundation supplies funds for the cluster to operate. The hospitals supply the research institutions with data from patients. Interactions among the various research institutions are transfers, given the intellectual property agreements within the research group.

5.2 Software Views

Figure 4b shows the elements in the graph representation of the software view. There are several software products, including the content management system (CMS) and the Drupal web portal framework, that operate on data but that do not interact with any other software. In addition to the

![Figure 4a. SVM Business View](image-url)
CMS there is the SimVascular cluster, which includes several products support SimVascular. The CMS allows clinicians to upload data taken before and after the cardiovascular procedures via a number of diagnostic procedures. In addition to uploading hand-recorded data, the website also stores large-sized DICOM and other formatted files encoding the information collected in each of the diagnostic procedures. Files are retrieved by the bioengineers and mechanical engineers for comparison of actual and experimental results. The web site also hosts a set of model simulations that allow users to experiment with simulations being developed by the project. Users can enter their own data into a model and receive predictions about the actual physical results.

A suite of image processing tools, Mimics, are used to enhance the medical images to facilitate their comparison to future images and to the experimental results from a set of predictive models [13]. This view also includes a blend of proprietary and open source vascular modeling programs, such as SimVascular [12]. Bioengineers utilize SimVascular as the basis for creating cardio-vascular models that seek to match the data and images recorded by the clinicians. This cluster includes 3 commercial numerical models that seek to match the data and images recorded by the clinicians. This cluster includes 3 commercial numerical models that seek to match the data and images recorded by the clinicians. This view includes improvements to research models, research software, and new medical products. TSVM integrates researchers with surgeons to produce innovations grounded in experience. The SVM activities have produced two primary types of innovations: extensions to proprietary and open source software and new computational models of the human circulatory system have been implemented and tested. A medical description of these innovations is beyond the scope of this paper. This view will grow as publication of research results allows more information to be revealed.

5.3 Innovation View

The innovation view is potentially very rich when the ecosystem includes numerous research facilities. This view includes improvements to research models, research software, and new medical products. TSVM integrates researchers with surgeons to produce innovations grounded in experience. The SVM activities have produced two primary types of innovations: extensions to proprietary and open source software and new computational models of the human circulatory system have been implemented and tested. A medical description of these innovations is beyond the scope of this paper. This view will grow as publication of research results allows more information to be revealed.

5.4 SVM Ecosystem Analysis

The SVM Ecosystem model was created to support strategic decision makers while they are charting new research directions. The model not only shows connections among groups, but also shows other groups that are not as connected. Metrics such as centrality measures can be used to identify nodes on each end of the connectivity spectrum. This information can be used to identify areas of potential over or underutilization.

The centrality measure identifies “clusters” where several nodes are interconnected as around SimVascular and the SVM cluster. By examining nodes such as SimVascular, we can consider whether to reduce long-term expenses by expending time and short-term resources for providing open source versions of the commercial libraries.

The scenarios in Table 1 provide the basis for defining useful analyses of the SVM ecosystem. Given that the SVM ecosystem has a business model driven by research needs and a more focused competitive environment, many of the scenarios need to be refined. For example, given the highly specialized nature of the ecosystem there is seldom a need to select among vendors for software components. For the same reason, a useful scenario to tailor and apply is the one in which we investigate the health of our suppliers, the failure of any one of which would leave a hole in the supply chains. The relatively small size of the ecosystem means the strategic manager needs to continually evaluate the scenario concerning innovations to be certain that resources are focused on the goals of the research.

The scenarios in Table 1 provide the basis for defining useful analyses of the SVM ecosystem. Given that the SVM ecosystem has a business model driven by research needs and a more focused competitive environment, many of the scenarios need to be refined. For example, given the highly specialized nature of the ecosystem there is seldom a need to select among vendors for software components. For the same reason, a useful scenario to tail and apply is the one in which we investigate the health of our suppliers, the failure of any one of which would leave a hole in the supply chains. The relatively small size of the ecosystem means the strategic manager needs to continually evaluate the scenario concerning innovations to be certain that resources are focused on the goals of the research.

There are several risks and constraints associated with the entities in the business view. The conduct of the hospitals involved in the cluster is governed by a number of standards of care, government privacy laws and medical regulations and processes. The research personnel are, likewise, governed by a variety of laws, regulations and professional ethics. This information is captured as attributes in the nodes under the heading of constraints. These governance issues are compounded by the international nature of the ecosystem meaning that multiple systems of regulations and laws will be encountered.

If we examine the software components used within the ecosystem, we see there exists single points of failure within both views of the ecosystem. In the web portal side, if either the Drupal API or the open source database shifts to a commercial license, this shift would require modification of several aspects of the ecosystem but would not be a major problem due to similar products providing implementations of the same standard interfaces. For example, Drupal provides a standardized database abstraction interface that can work with MySQL, MS SQL, Oracle or others. On the other hand, if, as pointed out above, one of the three commercial library providers went out of business, researchers could be severely limited in their operations since these do not have standard interfaces and would require considerable rework. As a result of the analysis we raise risks associated with the commercial libraries.
6. Evaluation of Modeling Method

In this paper we have focused on modeling, but the point of capturing data in a model is to be able to analyze the data and draw conclusions and answers to questions more readily. We have developed ecosystem models for organizations that have been used to generate scenarios for describing the details of system interactions and for architecture and system evaluations as well as supporting strategic decision making.

- A strategic decision maker wants to think ahead several steps to ensure there are no surprises as the result of a decision. By tracing across the appropriate relationships in the model a decision maker can identify the affected parties and reason about potential impact. For example, by tracing a transaction thread through the model a specific supply change is identified and can be investigated to determine the cause of poor quality.
- A strategic decision maker wants a modeling technique that is sufficiently flexible to meet changing needs without learning a new process. By taking a meta-model based approach the method presented in this paper can quickly be revised to provide additional stereotypes specific to problem concepts. Additional viewpoints can also be defined.

Our method should produce models that are accurate and comprehensible. We have no empirical evidence to support this assertion, but we do have a growing body of successful experience that supports the statement. The model should be comprehensible since there is a one-to-one mapping with entities in the real ecosystem and we use the real names in the model. The model should be accurate because the one-to-one mapping ensures we know what to model. We intend to continue efforts to evaluate the method.

7. Related Work

There has been much interest recently in the ecosystem perspective on organizations and their communities; however, we know of no other modeling techniques that use a viewpoint approach to modularize and simplify the model. There has been much interest recently in the ecosystem perspective on organizations and their communities; however, we know of no other modeling techniques that use a viewpoint approach to modularize and simplify the model. Of note, the authors in [6, 7, 9] have pioneered much of the work in ecosystem literature. In [7], the authors propose a two-dimensional chart for suggesting and evaluating activities and companies that seek to open up their organization in order to embrace software ecosystems. Along the X-Axis are organizational categories, with the Y-Axis representing activities at the strategic, tactical and operational level. The intersection of any two points results in a set of activities that can be performed in order to open up a company’s strategic and operational activities in an embrace software ecosystem development and management.

In [9], the authors provide a novel meta-model for instantiating models of software ecosystems. They provide three such levels of modeling, the software supply network level, the software ecosystem level and the software ecosystems level to show varying levels of granularity. Furthermore, their model makes a distinction between the product deployment context and the software supply network context, modeling both the business interactions and software composition duality found in software ecosystems.

In [4], Pettersson et al. provide a meta-model that is based off the Software Process Engineering Meta-model (SPEM) to instantiate models that facilitate the creation of an ecosystem based around eLearning and the development of community learning activities within a common platform. In [5], authors Yu and Deng provide an example of how the i* framework can be applied to software ecosystems to model the strategic goals of actors involved within the ecosystem.

8. Future Work

While we propose a novel way of providing a three-viewpoint perspective to related ecosystems, there is still additional work that can be done to further the analysis and modeling of ecosystems. Several future activities are possible including creating a reasoning framework for providing ecosystem analysis, or perhaps a taxonomy of standardized measures for each view of the ecosystem, or perhaps simply a taxonomy of observable qualities given topological structures in each of the ecosystem. While we have provided a real case study to demonstrate our modeling technique, it is important to note that the size of the SVM ecosystem makes the analyses seem trivial; however, within a large enough ecosystem, the sheer volume of entities inhibits the modeler’s ability to perform these analyses without the aid of the model. Yet another barrier to maturing our process of modeling ecosystems is the acquisition of data in order to model such large ecosystems. We must ensure that our meta-model is robust enough to handle a wide variety of ecosystems by applying it to a variety of situations. The criteria of comprehensiveness, maturity and traceability will be applied to the meta-model.

9. Conclusion

In this paper, we have presented a model for ecosystems based on a three viewpoint meta-model. Our model is capable of representing the business, software and innovative practices and relationships within an ecosystem. In our work with commercial, governmental, and academic organizations we have found a need for three types of viewpoints: a business viewpoint, a software viewpoint, and an innovation viewpoint; however, our model allows for any number of viewpoints to be accommodated. We have applied this model to a real world research project that spans domains and technologies and both produces and utilizes innovations in order to facilitate its research.
References