ECOS Modeling: A Modeling Tool, Repository for Models and Evolution Analysis of Software Ecosystems

Francisco Victor da S. Pinheiro
Universidade Federal do Ceará
Quixadá, Ceará, Brazil
victorpinheiro@ufc.br

Emanuel F. Coutinho
Universidade Federal do Ceará
Quixadá, Ceará, Brazil
emanuel.coutinho@ufc.br

Ronier da S. Lima
Universidade Federal do Ceará
Quixadá, Ceará, Brazil
ronier.lim@alu.ufc.br

Maria Erilane L. da Silva
Universidade Federal do Ceará
Quixadá, Ceará, Brazil
mariaerilane12@alu.ufc.br

Carla I. M. Bezerra
Universidade Federal do Ceará
Quixadá, Ceará, Brazil
carlailane@ufc.br

Rossana M. C. Andrade
Universidade Federal do Ceará
Fortaleza, Ceará, Brazil
rossana@ufc.br

Abstract

Context: Software Ecosystems (SECO) are a set of actors and components that function as a unit, relating to each other based on a common interest in providing solutions or services to the software industry. **Problem:** For a better visualization and understanding of SECOs, recent studies propose modeling the network formed by them. However, there is still no formalized modeling standard for this area, and there is a lack of tools, approaches, and notations that assist modeling. **Solution:** In this context, this work aims to present an evolution of the SECO modeling tool, called *ECOS* Modeling, in its version 4.0. In this version, some new features were added, such as saving the registration and login in the tool, saving and sharing the model in the repository, generating an analysis report of the SECO evolution, viewing the numerical data of the model and making the tool available in other languages. **IS Theory:** General systems theory, specifically the interfaces between different parts of systems and solutions that communicate, exchanging information. These parts can be systems from different institutions that require integration, consisting of Information Systems. **Method:** The tool evaluation was planned based on the TAM (Technology Acceptance Model), widely used to measure technology acceptance. The process involved the participation of 49 experts with experience in the areas of software and systems engineering. The selection of experts sought to ensure diversity of profiles, including academics and industry professionals, to broaden the scope and relevance of the results. **Summary of Results:** The evaluation results demonstrated strong validation of the tool, with experts rating it as highly useful in terms of usability, functionality, and alignment with requirements. The majority (73%) found the tool very useful, 71% stated that it fully met the proposed requirements, and 69% reported ease in finding information and resources. The tool's navigation was considered clear and easy by 55%, while 49% always knew their location and next steps within the tool. In addition, 55% of experts found the interface visually appealing and 45% reported low mental effort when using it. The study also revealed that 55% used modeling tools frequently and 63% worked with tools saving models in repositories, reinforcing the relevance and suitability of the tool for its target audience. **Contributions and Impact in the IS area:** As an emerging area, the results contribute to Information Systems by providing a practical tool for modeling, analyzing and evolving SECO. The main contribution is to fill the gap of a formal standard for SECO modeling, allowing the visualization and sharing of ecosystem dynamics, in addition to offering detailed reports that support the understanding of its evolution and identification of deficiencies.

CCS Concepts

• **Software and its engineering** → **Software development techniques**; • **Information systems** → *Information system applications*; • **Computing methodologies** → Model development and analysis; • **Human-centered computing** → Visualization.

Keywords

Software Ecosystem, Software Supply Network, Modeling

1 Introduction

Increasingly, software companies are realizing that they can no longer compete solely on the basis of product excellence. The ecosystems that surround platforms, such as operating systems, business applications and even social networks, are undeniably responsible for much of a platform's success. With this vision, software producing organizations need to design tools and strategies to improve their ecosystems and reinvent tools that others have already invented many times [12].

Software ecosystems (SECO) are an important concept for collaborative software development, and studies on their development are fundamental to understanding the underlying dynamics and modeling their behavior [21]. SECO is an effective way to build large software systems on top of a software platform, composing components developed by internal and external actors [12]. SECO can be defined as a collection of businesses that function as a unit and interact with a shared market of software and services, along with the relationships between them. In an ecosystem, competing companies can also act as partners to collaboratively develop software products and platforms for a shared customer base [14, 22].

To understand the SECI behavior, it is important to understand the structuring and roles of the actors and their relationships, as well as the central platform where they operate. One of the essential activities for describing SECO in a system is modeling, which involves different levels of technologies, notations and abstractions. However, despite initial advances in SECO research, there are few analytical models, real-world case studies, and integrated support for tools [17].

Models are built to provide a better understanding of systems or environments. A major barrier to the evolution of SECO towards decision-making in real industry is the lack of support for SECO modeling. Some work in the literature argues that SECO modeling is important for providing insights from representations, as well as enabling the analysis and comparison of "static" ecosystems, based on key concepts (e.g., organizations, relationships and flows) and existing methods (e.g., socio-technical networks and software supply networks) [7, 8, 15].

A way to standardize SECO modeling was suggested by Boucharas et al. [4], using SSN (Software Supply Network) notation, the most popular modeling method used by the SECO community. SSN is a series of interconnected software, hardware and services organizations that cooperate to meet market demands [5]. The graphic elements of the notation help to represent the actors and their behaviors and interactions within the ecosystem.

Motivated by the lack of SECO modeling in the literature and the absence of standardized models, this work presents the SECO evolution modeling and analysis tool, called ECOS Modeling, with functionalities associated with model maintenance, evolution analysis and repository for SECO models. This work also brings an evaluation of the tool by experts, with the aim of evaluating its functionalities.

This work is organized into the following sections: background on SECO and SSN notation in Section 2, some related work are described in Section 3, the *ECOS* Modeling tool is presented in Section 4, Section 5 describes an evaluation of the tool, Section 6 presents the relationship between work and information systems and their challenges, and finally, Section 7 presents the final remarks.

2 Background

This section presents some concepts about SECO, a brief summary about SSN notation, a brief overview of SECO modeling in industry.

2.1 Software Ecosystem

Ecosystems are groups that, although distinct from each other, interact in a mutual and disciplined manner. When a company takes its software products beyond its organizational limits, making its platform available and interacting with actors external to its organization, a SECO is formed [3]. SECO is a SE metaphor that has been applied to understanding the software supply chain dynamics centered on software platforms in the last decade [7]. SECO can be characterized by a set of elements, which can be: the actors, inside and outside the organization, the main software product, the software support platform and SECO assets. These elements are treated in an integrated manner, taking into account the interactions, information exchanges and artifacts [16].

A SECO can be understood as a subtype of business ecosystem [23], portraying an analogy from the early 90s to describe a new way of looking at business networks [19]. On the other hand, a SECO can be defined as a set of actors functioning as a unit, interacting with a market distributed among software and services, along with the relationships between them, which are usually supported by a technological platform or a common market, functioning through the exchange of information, resources and artifacts [13].

Such ecosystems have three elements: a hub, a platform or technology/market and a set of niche agents, where the hub is the owner of the platform and niche agents can use it to generate value for themselves and for the platform [11]. A human ecosystem is a set of actors and their connections and activities, such as transactions carried out around these connections, where physical and non-physical factors are considered, and can be discerned between commercial and social ecosystems [3].

In a commercial or business ecosystem, the actors are the businesses, suppliers and customers, the factors are the services and the transactions are, in addition to the financial ones, the sharing of knowledge and information, pre and post sales contacts [18].

2.2 Software Supply Network

One of the problems faced in modeling SECO is the lack of standardization. Boucharas et al. proposed standardizing SECO modeling using the Software Supply Network (SSN) strategy. As one of the most famous SECO modeling notations, SSN is a series of linked software, hardware, and service organizations which cooperate to meet market demands [5]. SSN represents the main actors and their interaction within a SECO using key elements.

The notation graphical elements aid to represent the actors, their behaviors and interactions within the ecosystem. SSN can be used to illustrate the structures of software supply chains in SECO [10]. SSN allows to reason about the business model of a software organization, showing its dependencies and flow [4]. With it, it is possible to explain the commercial relationships between the elements of a SECO in terms of input and output flows between the actors, having a terminology based on the terms used in software development activities, making it easily understandable for developers [22]. Figure 1 shows the elements used in the SSN notation. Figure 2 presents a brief example of SECO modeling using the SSN notation, showing the relationships of suppliers and intermediaries with the company of interest, in addition to the customer. Each element relationship (there may be more than one) can consist of a financial value, a data type, or any by-product that is being passed between elements. Actors, Trade Relations, Flows and Gateways are the essential elements of SSN modeling. Thus, an actor will be an organization or company that participates in a SECO, and may be a Company of Interest, Supplier, Customer, Intermediary or Customer of Customer. What will connect two actors will be a Business Relationship, which can be formed by one or more Flows [22].

Based on a review of modeling techniques used in SECO, Sadi and Yu [22] reported that among the selected techniques (UML, Graph representation, PDC, i*, etc), only SSN is specifically proposed to model SECO, while the others are generic and widely used in other areas. Although SSN does not support describing contributor activities, most techniques provide little support for describing contributor constraints, attributes, and interactions.

2.3 Software Ecosystem Modeling

Three important points for using SECO modeling are: it is the most significant way to understand SECO, regardless of the type to which they refer (e.g., open, commercial, social); analysis is performed most satisfactorily through modeling; and the prediction of how

Company of Interest	**Company of Interest / Product of Interest: The product/company itself. It distributes the product in the business model defined for the environment.**
Supplier	**Supplier: Company or supplier of products and/or services. It provides one or more required products or services.**
Customer	**Customer: Actor that directly or indirectly acquires or uses the product**
Intermediary	**Intermediary: A company, product or service that operate between two actors in order to distribute a product or service. It can be distributors, sellers, resellers, etc.**
Agregator	**Aggregator: Companies, products or services that operate between any two agents in order to add value to a product or service as well as distribute or resell it.**
Customer's Customer	**Customer's Customer: A customer might have his own customers being provided with a product or service directly or indirectly from the CoI. Examples: product support, updates, etc.**
X.Y	**Trade Relationship: Represents an artifact or service flow from one actor to another. It can be data, software, service, money, etc.**
	Flow: It connects two actors. A relationship might be complex, constituting of many flows of arbitrary directions.
OR	**OR Gateway: Allows execution of one or more streams between the inputs.**
	XOR Gateway: Allows execution of only one of the input streams.

Figure 1: SSN notation [4] with extension of [5].

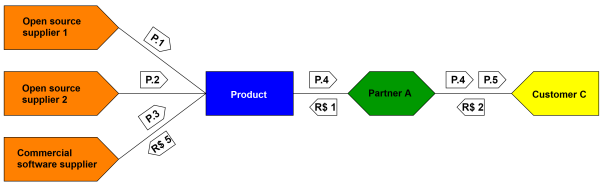

Figure 2: Generic SECO model in SSN notation [4].

the ecosystem is pending certain decisions, which are made more efficiently with the support of modeling [15]. Based on the literature and interviews, Jansen et al. [15] found that the modeling objectives are common: (i) to provide information, for researchers an image in a document is able to provide an overview of the SECO aspect, facilitating the identification of actors with key roles, such as the keystone and providing information about relationships; (ii) analyze static ecosystems, with the real analysis of SECO it is possible to identify key relationships, density of relationships, connection within SECO and differences in the size of the importance of the actors; and (iii) compare the ecosystem, through visualization it is convenient to analyze the maturity, development of SECO dynamics, and see how the forms, connectivity and objects in SECO models differ. The following elements are observed in ecosystem models: (i) organizations and their types, and the entity on which the model revolves; (ii) relationships, which may be component dependencies, business relationships and collaborations; and (iii) flows, which can be the amount of code that passes from one project to another, the value that passes from a software reseller to a platform provider, knowledge flows and dependencies in a social network, however, this entity ends up not getting much attention.

3 Related Work

The literature presents a limited number of approaches, methods, notations and tools focused on the SECO context, as highlighted by [20], who highlighted this gap through a systematic mapping of the literature. Despite this, some tools and initiatives that seek to motivate and facilitate SECO modeling were described in subsequent works, offering specific contributions to the area.

Franco-Bedoya et al. [9] the authors performed a systematic mapping to assess the state of the art in SECO, investigating how knowledge about SECO is represented, including notations, techniques and modeling tools. The results classified existing modeling and analysis techniques, revealing that most adapt existing methods or use *ad hoc* models, without proposing new techniques or guidelines. The most used notations include ad hoc, tabular and conceptual maps, while some representations used class diagrams, metamodels or mathematical models. Most works do not perform analysis on SECO models, and those that do use mathematical, visual or statistical techniques for specific cases.

Ambrósio et al. [2] described ESECO (E-Science Software ECOsystem), a platform to support collaboration between geographically distributed researchers, supporting interoperability and data provenance solutions. The work aims to enhance collaboration between researchers throughout the lifecycle process of scientific experimentation on a SECO scientific platform. The research used an approach to support the scientific software ecosystem integration with external scientific platforms. A common idea in ECOS Modeling is a collaboration between research, but in Ambrósio et al. [2], there was no SECO modeling.

Coutinho and Bezerra [6] discussed the dynamic variability of the SOLAR educational software ecosystem and software modeling. As an example, aspects of the dynamic variability of the resource model of the SOLAR VLE discussion forum functionality, one of the most used services within SOLAR SECO, were discussed. As the main conclusion of this work, it was identified that the use of the contextual characteristics diagram allows the study of the dynamic aspects of a system, even more supported by tools to support the automatic collection of measurements.

Alencar et al. [1] presented the ARIEL tool, aiming to mitigate the gap in support for SECO teaching in Software Engineering. A user experience assessment was designed using the DECIDE framework, with undergraduate and postgraduate students. As a result, participants stated that the tool was easy to use and they made few errors; the majority felt satisfied when using the tool, and the level of mental effort to carry out the activities was low. However, the tool does not have SECO modeling functionality, it only uploads model images, requiring models to be generated outside the environment, without standards and compatibility.

In this sense, there is a lack of tools that allow the creation of SECO models in a standardized notation, in this case, the SSN is no longer treated as a gap. Related work served as motivation to create and improve the *ECOS* Modeling tool that meets SECO requirements for modeling and model sharing, inspiring functionalities for knowledge sharing or collaboration. Table 1 presents a comparison between related works and the proposed study, highlighting the unique contributions of this work in relation to previous ones.

Table 1: Comparison between related works and the proposed study

Work	Objective	SECO Modeling	Notations/Techniques Used	SECO Analysis	Unique Contributions of the Proposed Study
Franco-Bedoya et al. [9]	Systematic mapping of techniques and tools for SECO modeling.	Partial	Ad hoc maps, tabular, conceptual diagrams, among others.	Limited analysis (specific cases)	Integrated tool for SECO modeling with standards and sharing.
Ambrósio et al. [2]	Support collaboration in geographically distributed scientific software ecosystems.	No	No focus on notations; support for interoperability.	No	Standardized modeling and sharing in scientific SECOs.
Coutinho and Bezerra [6]	Study of dynamic variability in the SOLAR educational software ecosystem.	Yes, focused on dynamic variability	Contextual characteristic diagram.	Yes	Broad applicability in SECOs with support for different contexts.
Alencar et al. [1]	Support SECO teaching in Software Engineering through tools.	No	Image uploads without standards or compatibility.	No	Integrated modeling within the environment with standardization and compatibility.
Proposed Study	Develop a tool for SECO modeling with standardized notations and collaboration functionalities.	Yes	SSN notation, standardized models.	Yes, integrated with modeling	Modeling standards, sharing, and collaboration support.

4 *ECOS* Modeling

The *ECOS* Modeling tool in its version 4.0[1] is a significant evolution of the *ECOS* Modeling web tool. Now, it offers a robust environment that allows you to perform SECO modeling using SSN notation, maintain models collaboratively with a model repository functionality, analyze SECO evolution, simulate SECO behaviors and manage user registration functionalities and Login.

The tool was developed using Vue.js and mxGraph.js on the front-end, and Java with the Spring ecosystem on the back-end. It employs the PostgreSQL relational database management system to manage the database and the MinIO file server to store the models in the repository. The license for the tool is Apache Open Source [2]. The tool has a modern, simple and minimalist screen design. Very intuitive and easy to use, in order to facilitate the user's understanding of its functionalities. In the following sections, we will present the tool architecture and functionalities as well as main characteristics.

4.1 Architecture

An overview of the tool architecture can be seen in Figure 3, which has a client-server structure with the server on which *ECOS* Modeling is hosted, receiving requests for resources from its clients. The development stack used to implement the application was Vue.JS + Mxgraph.JS for the front-end and *Java* 11 with Spring ecosystem for the back-end.

The architecture components are SECO API, Relational Database, Spring, Vue.JS, MinIO, and Swagger, which communicate via the http protocol. The components are described below: **SECO API:** is

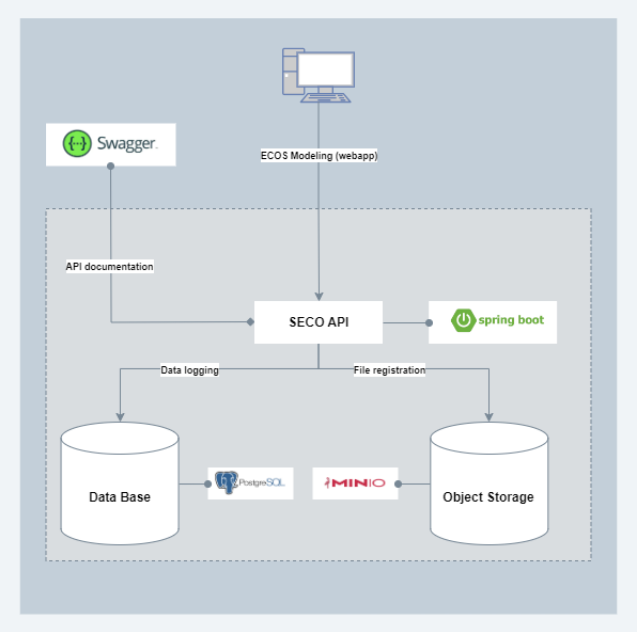

Figure 3: *ECOS* Modeling architecture.

the back-end structure of the application made available in Restfull API format, which focuses on the main business rules, all security and user access control and the interface between the client, the database and the saved files; **PostgreSQL:** relational database, where users, models and references to their respective files are

[1] http://200.129.43.193:33380/
[2] https://www.apache.org/licenses/LICENSE-2.0

saved; **MinIO:** is a file server based on the S3 protocol, where files relating to models are saved; and **ECO Modeling webapp:** is what makes up the modeling system and the repository for models.

4.2 Features

In this section, the main features and functionalities available in the *ECOS* Modeling tool are described. Figure 4 shows the tool home screen with the top menu with features (editor, models, evolution, registration, and login), a brief description of the tool, and some of the most accessed models from the model repository. These elements are designed to facilitate navigation and quick access to the tool's main functionalities.

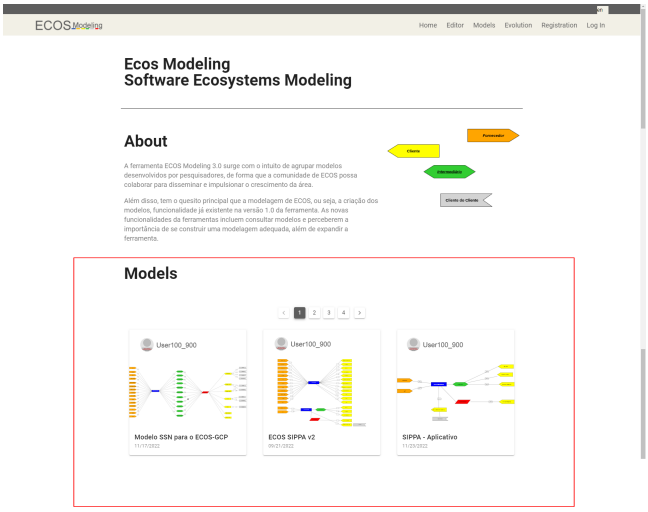

Figure 4: Tool home screen.

The modeling functionality is divided into six areas: (i) Side menu on the left, where the components of the SSN notation are located; (ii) Central area that allows modeling and editing of models; (iii) Toolbar on the right that has model visualization, statistics and text formatting functionalities; (iv) Top toolbar that contains features for exporting and importing models, zooming, editing models (such as copying, pasting, cutting, redoing and undoing); (v) Environment internationalization functionality that allows changing the interface language; and (vi) login if the user is not logged in or profile where the user can access their registration data. The main areas have functionalities that are described below in Figure 5.

Registration and login: These features allow new users to register with the tool and existing users to log into their accounts. Registration requires basic information such as name, email and password, while login allows authenticated users to access and manage their models.

Model Repository: The functionality allows users to store, share and access SECO models centrally. This repository facilitates collaboration between users, enabling the import and export of models, in addition to maintaining a version history. With this, users can reuse existing models, promote knowledge sharing and ensure project continuity.

Evolution analysis: The functionality allows users to analyze SECO changes and development over time through SSN models. This tool provides metrics, visualizations and data on the evolution of a SECO through models, identifying trends, patterns and possible improvements. With evolution analysis, users can evaluate the impact of modifications, make informed decisions and ensure the quality and efficiency of the models developed.

Internationalization: Enables the internationalization of the tool. The functionality allows the translation of the tool in three languages: Portuguese, English and Spanish.

Modeling: Functionality that allows the user to manipulate the components of the SSN notation, from the left side menu to the modeling area, allowing the creation of the SECO model. It also has infinite drawing area functionality, allowing the creation of large-scale models.

Export model: It allows the export of the model created in some extensions, both as an image (PNG) as well as in JSON and XML formats. This functionality allows the user to later use the model created in the tool.

Import model: It allows the import of the model in XML format in the tool for maintenance and subsequent evolution, such as the addition of new components and relationships to SECO.

Manipulate model: Series of functionalities that help in the creation of the model, its structuring and subsequent maintenance and evolution. The manipulation functionalities are: zoom in and out, standard zoom, delete component, cut, copy, paste, redo, undo, select all and deselect, print, component forward, backward, bold, italic and underline.

Properties: The functionality displays the XML properties of the model created at the time of modeling to the user. This functionality allows to view the properties of the model.

Statistics: The functionality generates a statistical report of the model, on the SECO components and relationships. The functionality allows data extraction for future quantitative analysis of the model, evolution, versions and maintenance.

Figure 6 presents the repository of SECO models generated in the tool and shared by users. This centralized repository allows users to store, access and collaborate on various SECO models. A template details functionality is highlighted, allowing users to view information such as the title, description, creation date, and last edited date for each template. This functionality facilitates the management and organization of models, promoting better understanding and traceability of changes made over time.

In the evolution analysis functionality, the user selects two or more SSN models of an SECO in the model list and clicks on "generate evolution report". A tab opens with the selected SECO report, containing graphs, tables, metrics, percentage data, averages and absolute data. The comparison between SECO versions includes graphs with numerical and percentage comparisons, total components and relationships, numerical and percentage variation between the first and last version, as well as a general table with numerical, percentage and average differences. Figure 7 illustrates the SSN models selection page for evolution analysis, the mentioned graphs and the comparison table with quantitative metrics results.

Figure 8 presents an example of SECO modeling using the *ECOS* Modeling 2.0 tool. It consists of a simple example, illustrative only, focused on the presentation and use of the tool. The model only

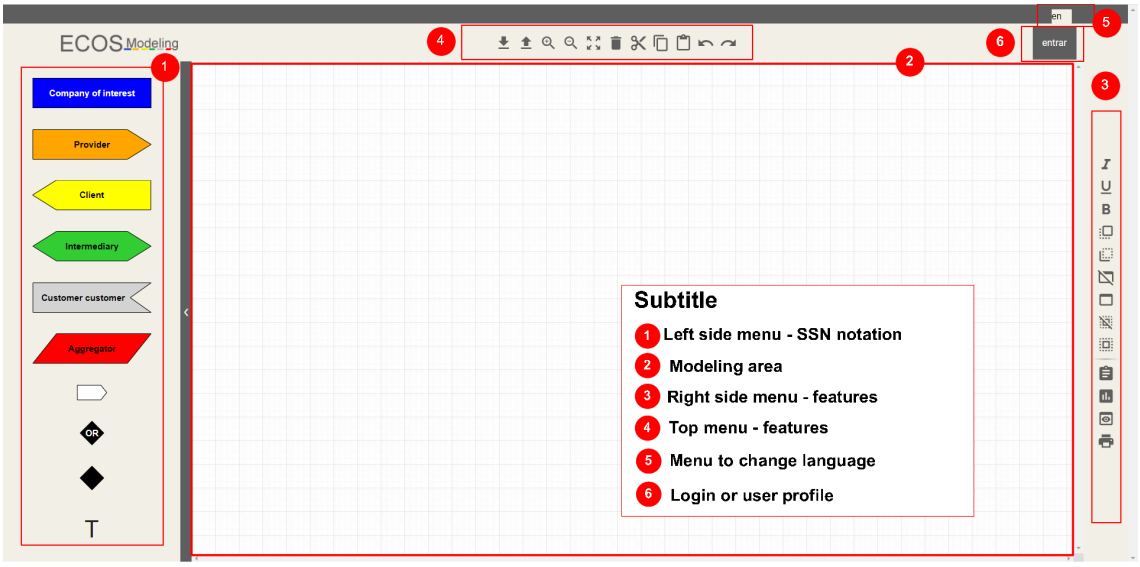

Figure 5: Modeling screen.

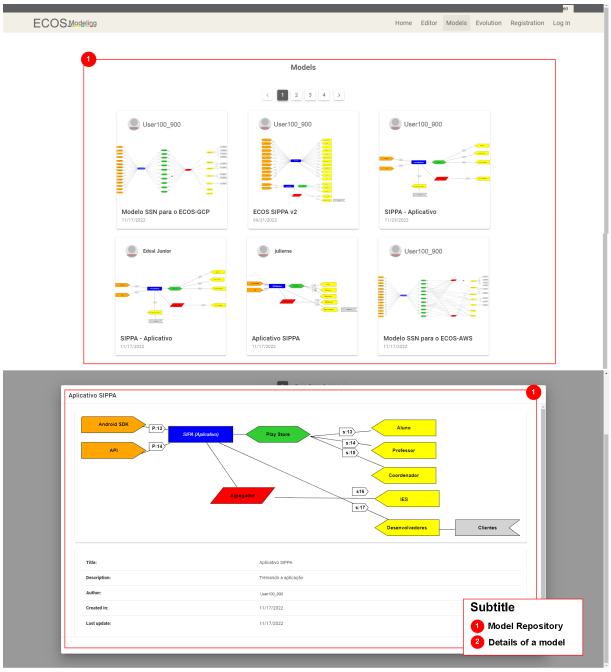

Figure 6: Model repository screen.

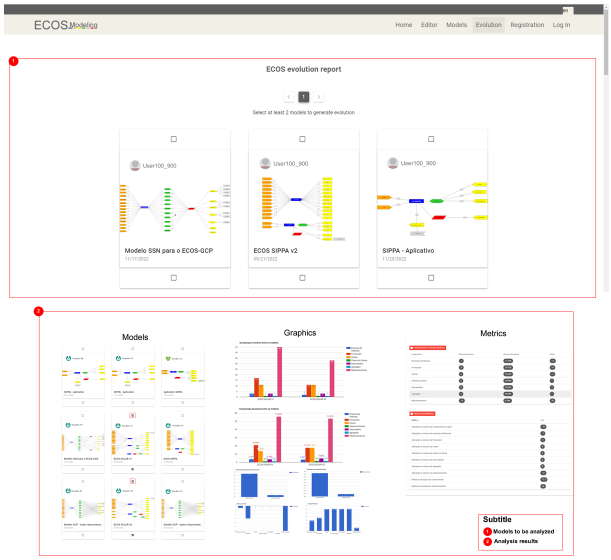

Figure 7: Evolution analysis screen.

presents the use of SSN components in the modeling area. The model at the time of creation/editing in the tool is fully responsive, and has a high image quality.

Another example is the static reporting functionality of the model, shown in Figure 9. The functionality provides a numerical overview of SECO, in relation to its actors, roles and relationships. The generated report is presented to users both on screen and can also be exported in PDF format. The report generated on the model shows the number of elements present in the SECO, how many companies of interest, how many customers, how many suppliers, how many intermediaries, how mnany aggregators, the total of relationships and the total of SSN components in the SECO. This report is important for future studies related to the health, quality and evolution of SECO, such as how much the model has evolved in terms of quantity of elements over a given period of time. This feature is the beginning of a series of statistical features about the model that will later be implemented in the tool.

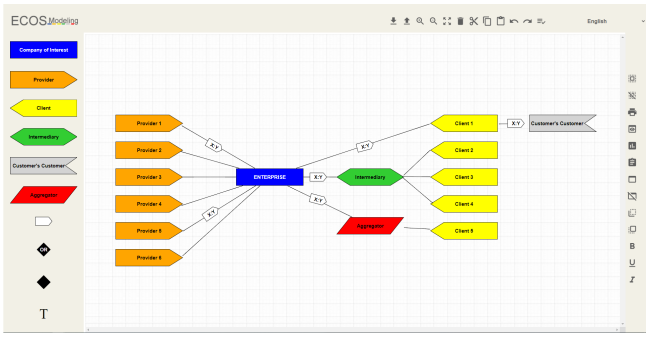

Figure 8: Modeling example in *ECOS* Modeling.

In the example of Figures 8 and 9 we have 1 company of interest, 5 customers, 6 suppliers, 1 intermediaries, 1 aggregator, 1 customer' customer, 13 relationships and a total of 15 actors in this SECO.

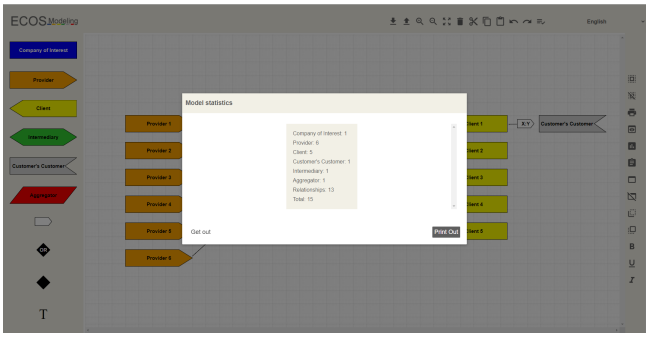

Figure 9: SECO model statistical report.

Another example is illustrated in Figure 10, of the export functionality, which is to save the created model and import, which is to open the model in the tool for editing and maintenance. These are features that allow exporting the model in some extensions (XML, SVG, PNG and JSON) and importing the model into the tool itself in XML format. Import functionality is important for maintaining the model over time, reusing the old model, rather than recreating it from scratch, just making small edits, adding or removing components and relationships to the model.

5 Tool Evaluation

First, the tool evaluation was done by a specialist, who followed a pre-defined script to guide the use of *ECOS* Modeling resources. Then, a questionnaire based on the Technology Acceptance Model (TAM) with closed questions and open-ended questions in free text for comments, strengths and weaknesses, was completed by 49 experts to report their impressions.

5.1 Evaluation Scenarios

The scenarios consists of using the tool to create a SECO model, saving the model in the repository and analyzing the evolution of a SECO, according to an action script, to guide the evaluator in testing the available functionalities. The script has a guiding

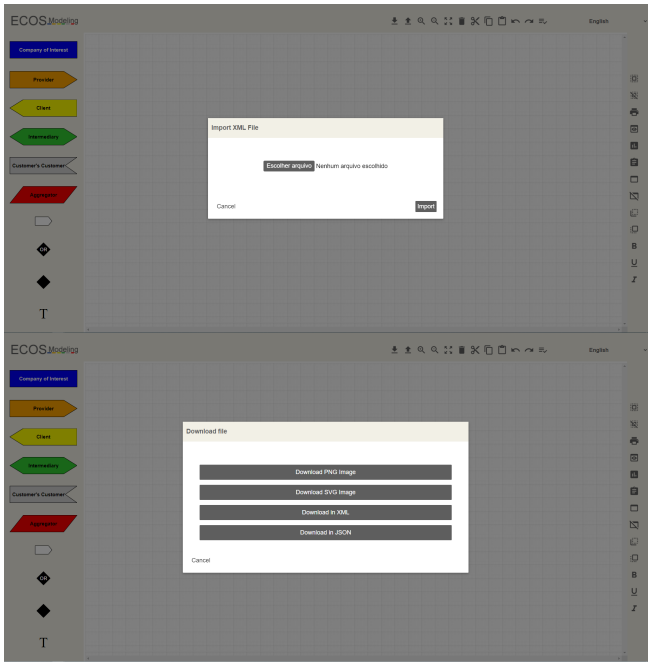

Figure 10: Import and export features.

nature and does not interfere with the expert's actions during the evaluation.

5.1.1 ***Script:*** The tool usage script is divided into two sections. The first helps the evaluator to register, log in and create a SECO model belonging to any domain, testing the functionalities of: changing language, dragging, dropping, connecting, renaming SSN components testing the functionalities of copying, pasting, cutting, deleting the component, undo and redo, zoom in, zoom out, default zoom, select all objects, deselect, group and ungroup, move picture backward or forward, change picture label style, bold, italic or underlined text, export the model in PNG, SVG, XML and JSON formats, import the model into the tool in XML format, view the properties, model statistics and save the model in the repository. The second section refers to the test with SECO evolution analysis, the evaluators can choose two or more UM SECO models in different versions already existing in the tool, generate the evaluation report, observing the results shown by the tool, both in graphs, tables and metrics.

5.1.2 ***Questionnaire:*** Table 2 presents the evaluation questions, divided into two groups: technical and opinion questions. Opinion questions were divided into "yes or no" and on a five-level scale, for example "Not at all pleasant", "Not pleasant", "Neutral", "Pleasant" and "Very pleasant". The opinion questions were free text.

5.2 Results

Based on the results, experts could obtain some impressions about the tool, allowing a robust validation of the tool. In terms of technical issues, experts considered the tool pleasant, very useful, highly suitable and with several other interesting features, shown in Figure

Table 2: Technical (T) and opinion (O) questions.

ID	Description
T1	How frequently do you use systems modeling tools?
T2	Have you ever performed systems modeling in a tool that saves the model in a repository?
T3	In the *ECOS* Modeling tool I always know where I am and how to get where I want
T4	The navigation features of the *ECOS* Modeling tool are all clear and easy to find
T5	The tool has an attractive/pleasant look/interface
T6	The tool is useful for the user
T7	The tool meets the requirements it proposes
T8	The tool makes it easy to find the information and features I want
T9	My level of mental effort in using the tool was
O1	Strong points
O2	Weak points
O3	Improvement suggestions

11. Practically all evaluations are at the most positive values on the scale. **Frequency of use of systems modeling tools:** 55% of experts used modeling tools frequently or very frequently. **Modeling with a repository:** 63% of experts performed modeling in a tool that saves the model in a repository. **Orientation and navigation:** 49% of experts always knew where they were and how to get to what they wanted in the tool. **Clarity and ease of navigation:** 55% of experts considered that the tool's navigation features were clear and easy to find. **Appearance and interface:** 55% of experts found the tool visually attractive and pleasant. **Usefulness:** 73% of experts found the tool very useful. **Compliance with requirements:** 71% of experts considered that the tool fully meets the proposed requirements. **Ease of finding information and features:** 69% of experts found it easy to find the desired information and features. **Mental effort:** 45% of experts reported that the level of mental effort when using the tool was low.

Regarding the strengths, it is worth highlighting the following comments: "*The tool offers a simple and practical way of adopting the SSN notation for SECO modeling, being easy to learn and contributing to its use in initial disciplines of modeling courses in the software ecosystem*", *"The tool is very consistent and well presented. It has interesting and robust features. Regarding the study approach, it is well implemented and well presentable, easy to understand and intuitive. I really liked the graphs and tables presented".*

Regarding weaknesses, experts pointed out some technical defects, but these did not impact the modeling. Finally, as suggestions for improvement, many of them were related to the look of the tool, usability and connection of elements, which could be better refined, but a comment can be highlighted about a good suggestion for improvement: *"To facilitate first interaction with the user, onboarding could be added. Further helping early users to use the tool".*

5.3 Discussions

The experts' assessment provided valid feedback from someone who knows the field of SECO, has worked with SECO modeling and SSN notation, and has known SECO for at least 2 to 7 years. Through the use of the tool, some improvements were identified, both from a technical and usability point of view. However, the SECO modeling activities themselves, model repository and evolution analysis could be carried out without problems.

Although the tool received a good opinion from experts, this latest version has not yet been applied in the classroom, which is one of the main ideas of the research: promoting the dissemination of SECO through modeling. The possibility of generating different searches increases as the tool is used more. This also increases the dissemination of SECO and collaboration with research in the area, making the tool a central point for studies.

One aspect that the tool intends to explore is investment in the quality and health of SECO. Once these items are met, it is believed that it will be possible to research the simulation and evolution of SECO models, which may include an element of quality and software maintenance in a more global way, which is the vision that a SECO enables. Finally, it is understood that the tool's target audience extends beyond SECO researchers, interested in having a global view of relationships between suppliers and customers from a central platform, and what happens between them. From an educational point of view, teachers and students of Software Engineering, Systems Analysis and Design and Software Quality can benefit from the tool during classes with laboratory practices.

5.4 Research Limitations

The limitations identified in the work and in the evaluation of the tool highlight aspects that may influence its adoption and applicability. First, the tool presents a limited focus on the user experience. Although these are aesthetic and interaction issues, they directly impact acceptance and ease of use, especially for new users. The absence of an onboarding process aggravates this scenario, as it makes the learning curve more difficult and may discourage use by those who are not familiar with the tool. In addition, its application has not yet been validated in different scenarios, such as the educational environment or with more diverse audiences, such as students, which limits the generalization of the results obtained.

Another relevant point is the incomplete exploration of advanced functionalities, such as simulation and SECO model evolution analysis. These features are promising for improving the quality and health of software ecosystems, but they have not yet been sufficiently developed or evaluated. In addition, the lack of quantitative analysis of technical performance, including aspects such as response time and scalability, raises questions about its applicability in higher-demand environments. These limitations point to opportunities for future improvements, both in the user experience and in the technical and pedagogical capabilities of the tool.

6 Relationship to Information Systems

Integrating areas in the software ecosystem is essential to align different disciplines and knowledge, allowing for a more collaborative and efficient approach to developing solutions. The tool allows integrating different areas of the software ecosystem, providing support for modeling that covers both technical and organizational aspects. This integration facilitates collaboration between engineering, business and data analysis teams, promoting a unified view of the ecosystem and allowing strategic decisions to be made based on consistent and accessible information.

Communication between systems is a central element in software ecosystems, ensuring interoperability and efficient and secure data sharing. The tool contributes to communication between systems

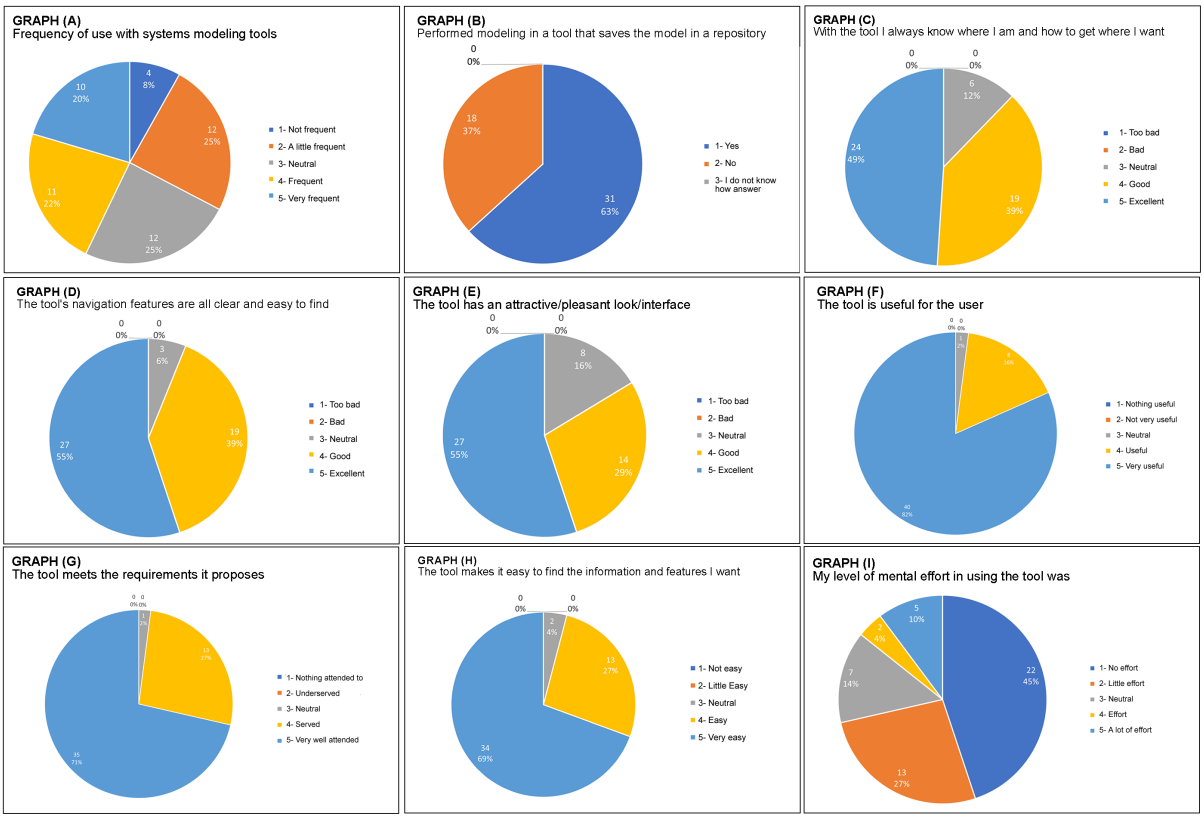

Figure 11: Graphs with evaluation results.

by providing a centralized environment for data modeling and analysis. It allows identifying and documenting integration points between systems within the ecosystem, facilitating interoperability and information exchange. This ensures that connected systems in the ecosystem can operate in a coordinated and efficient manner.

SECOs are comprised of a variety of roles, including developers, customers, suppliers, and administrators. Understanding and managing these relationships is crucial to the success of the platform, as each role brings unique perspectives and demands. The tool models and visualizes the relationships between the various roles in the ecosystem, such as developers, suppliers, customers, and administrators. This helps map interactions and responsibilities, promoting a better understanding of internal dynamics and enabling the alignment of interests between the involved parties.

The tool provides a global view of the ecosystem by allowing users to visualize and analyze the interaction between people, systems, and technologies. This makes it possible to identify dependencies and optimize processes, creating a more cohesive environment that is prepared to deal with changes and challenges. This holistic approach makes it easier to identify interdependencies, solve problems, and optimize processes, promoting a balanced and sustainable ecosystem in the long term.

The evolution of the core platform is a key aspect of the ecosystem's survival and growth. It involves adapting to new market demands, incorporating emerging technologies, and continually improving the user experience. A well-managed core platform can serve as a catalyst for innovation and ecosystem expansion. The tool supports the evolution of the core platform by providing capabilities to track the evolution of models and the impacts of changes in the ecosystem. It allows for tracking changes, forecasting scenarios, and planning improvements, helping the core platform adapt to new demands and technologies in a strategic way.

Throughout its life cycle, a software platform generates a series of financial impacts that must be carefully monitored and managed. These impacts include development and maintenance costs, revenue generation, and return on investment. With the tool, it is possible to model and monitor the costs and benefits related to the maintenance and evolution of the ecosystem. This includes the analysis of required resources, ROI (Return on Investment) estimates, and identification of areas that can be financially optimized, providing support for more informed and strategic decisions. Understanding these factors helps to make strategic decisions to maximize the economic value of the ecosystem.

The concept of ecosystems of ecosystems is explored by enabling the modeling of interactions not only within an ecosystem, but also between different ecosystems. This helps to identify synergies, dependencies and opportunities for collaboration between multiple ecosystems, expanding the scope of analysis and the impact of

the solutions developed. This interconnection creates complex networks of collaboration and dependency, allowing ecosystems to share resources, knowledge and innovation.

The tool plays an important role in the Information Systems area by facilitating the modeling and management of software ecosystems. It helps organize information, promote interoperability between systems and improve collaboration between different stakeholders. In doing so, it provides a solid basis for the analysis and evolution of platforms, enabling more informed decision-making and contributing to efficiency and innovation in complex contexts.

7 Conclusion and Future Work

This work presented the ECOS Modeling tool, designed to enable SECO modeling using the SSN notation, provide a repository for SECO models, and facilitate the analysis of SECO evolution. These features aim to support its application in Software Engineering or related disciplines, allowing for both study and practice in the SECO domain. The tool assists in critical activities of the software development process, such as specification, design, and evolution, offering a comprehensive view of systems and their associated actors.

The evaluation of the tool highlighted its strengths, with experts praising its usefulness, compliance with requirements, and ease of finding information and features. The tool's usability and visual design were well-received, with most ratings reaching positive values on the evaluation scale. Notably, 73% of experts found the tool very useful, 71% considered it fully compliant with the proposed requirements, and 69% highlighted its ease of finding desired information. However, areas for improvement were identified, such as usability refinements, onboarding features for new users, and a more polished visual design. These insights emphasize the potential for enhancing the tool to better meet user needs.

The *ECOS* Modeling tool contributes significantly to the SECO community and Software Engineering by enabling the creation and maintenance of standardized models, fostering collaboration among researchers and professionals, and advancing knowledge in software ecosystems. Its functionalities, such as model evolution analysis and repository management, position the tool as a valuable resource for both practical and educational purposes.

As future work, we aim to expand the tool with features focused on assessing SECO quality and health and implement metrics for SECO simulation. Additionally, we plan to conduct further studies on the tool's usability and its impact on SECO learning in classroom settings, strengthening its relevance and applicability in academic and professional environments.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES) – Finance Code 001 - Process Number 88887-617302/2021-00.

References

[1] Igor R. Alencar, Emanuel F. Coutinho, Leonardo O. Moreira, and Carla I. M. Bezerra. 2020. A Tool for Software Ecosystem Models: An Analysis on Their Implications in Education. In *Proceedings of the 34th Brazilian Symposium on Software Engineering* (Natal, Brazil) *(SBES '20)*. Association for Computing Machinery, New York, NY, USA, 405–414. https://doi.org/10.1145/3422392.3422486

[2] L. M. Ambrósio, P. Marques, J. M. N. David, R. Braga, M. A. Ribeiro Dantas, V. Ströele, and F. Campos. 2019. An Approach to Support Data Integration in a Scientific Software Ecosystem Platform. In *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. 39–44.

[3] Jan Bosch. 2009. From software product lines to software ecosystems. In *2009 13th International Software Product Line Conference*, Vol. 9. 111–119.

[4] Vasilis Boucharas, Slinger Jansen, and Sjaak Brinkkemper. 2009. Formalizing Software Ecosystem Modeling. In *Proceedings of the 1st International Workshop on Open Component Ecosystems* (Amsterdam, The Netherlands) *(IWOCE '09)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/1595800.1595807

[5] Gabriella Costa, Felyppe Silva, Rodrigo Santos, Cláudia Werner, and Toacy Oliveira. 2013. From Applications to a Software Ecosystem Platform: An Exploratory Study. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems* (Luxembourg, Luxembourg) *(MEDES '13)*. ACM, New York, NY, USA, 9–16. https://doi.org/10.1145/2536146.2536159

[6] Coutinho and Bezerra. 2020. A study on dynamic aspects variability in the SOLAR educational software ecosystem. *Journal of the Brazilian Computer Society* 26, 1 (2020), 1–19.

[7] Emanuel F. Coutinho, Italo Santos, Leonardo O. Moreira, and Carla I. M. Bezerra. 2019. A Report on the Teaching of Software Ecosystems in Software Engineering Discipline. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (Salvador, Brazil) *(SBES 2019)*. ACM, New York, NY, USA, 130–139. https://doi.org/10.1145/3350768.3351302

[8] Emanuel Ferreira Coutinho, Davi Viana, and Rodrigo Pereira dos Santos. 2017. An Exploratory Study on the Need for Modeling Software Ecosystems: The Case of SOLAR SECO. In *9th International Workshop on Modelling in Software Engineering (MISE)* (Buenos Aires, Argentina) *(MISE '17)*. IEEE Press, Piscataway, NJ, USA, 47–53. https://doi.org/10.1109/MiSE.2017.3

[9] Oscar Franco-Bedoya, David Ameller, Dolors Costal, and Xavier Franch. 2017. Open source software ecosystems: A Systematic mapping. *Information and software technology* 91 (2017), 160–185.

[10] Eko Handoyo, Slinger Jansen, and Sjaak Brinkkemper. 2013. Software ecosystem modeling: the value chains. In *Proceedings of the Fifth International Conference on Management of Emergent Digital Ecosystems*. 17–24.

[11] Marco Iansiti and Roy Levien. 2004. *The keystone advantage*. Harvard Business Press.

[12] Slinger Jansen. 2020. A focus area maturity model for software ecosystem governance. *Information and Software Technology* 118 (2020), 106219.

[13] Slinger Jansen, Sjaak Brinkkemper, and Anthony Finkelstein. 2007. Providing transparency in the business of software: a modeling technique for software supply networks. In *Working Conference on Virtual Enterprises*. Springer, 677–686.

[14] Slinger Jansen, Sjaak Brinkkemper, and Anthony Finkelstein. 2009. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. IWSECO@ ICSR 2009 (2009).

[15] Slinger Jansen, Eko Handoyo, and Carina Alves. 2015. Scientists' needs in modelling software ecosystems. In *Proceedings of the 2015 European Conference on Software Architecture Workshops*. 1–6.

[16] T. M. P. Lima. 2015. Uma Abordagem Socio-técnica para Apoiar Modelagem e Análise de Ecossistemas de Software. (2015), 83.

[17] Konstantinos Manikas. 2016. Revisiting software ecosystems Research: A longitudinal literature study. *Journal of Systems and Software* 117 (2016), 84 – 103. https://doi.org/10.1016/j.jss.2016.02.003

[18] John D McGregor. 2010. A method for analyzing software product line ecosystems. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*. 73–80.

[19] J Moore. 1996. The death of competition. *Harper Paperbacks* (1996).

[20] Francisco Victor Pinheiro, Emanuel Coutinho, Maria Erilane Silva, and Carla Bezerra. 2024. A Systematic Mapping of Health, Quality, Evolution, Simulation and Modeling in Software Ecosystems. In *Proceedings of the 20th Brazilian Symposium on Information Systems* (Juiz de Fora, Brazil) *(SBSI '24)*. Association for Computing Machinery, New York, NY, USA, Article 26, 10 pages. https://doi.org/10.1145/3658271.3658297

[21] Konstantinos Plakidas, Daniel Schall, and Uwe Zdun. 2017. Evolution of the R software ecosystem: Metrics, relationships, and their impact on qualities. *Journal of Systems and Software* 132 (2017), 119–146.

[22] Mahsa H Sadi and Eric Yu. 2015. Designing software ecosystems: How can modeling techniques help? In *Enterprise, Business-Process and Information Systems Modeling*. Springer, 360–375.

[23] Ivo van den Berk, Slinger Jansen, and Lútzen Luinenburg. 2010. Software Ecosystems: A Software Ecosystem Strategy Assessment Model. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (Copenhagen, Denmark) *(ECSA '10)*. Association for Computing Machinery, New York, NY, USA, 127–134. https://doi.org/10.1145/1842752.1842781