Journal on Interactive Systems, 2022, 13:1, doi: 10.5753/jis.2022.2602
This work is licensed under a Creative Commons Attribution 4.0 International License..

A Tool for Supporting the Teaching and Modeling of Software Ecosystems Using SSN Notation

Francisco Victor da S. Pinheiro [**Federal University of Ceara** | ***victor.pinheiro.ce@alu.ufc.br***]
Emanuel Ferreira Coutinho [**Federal University of Ceara** | ***emanuel.coutinho@ufc.br***]
Italo Santos [**Northern Arizona University** | ***italo_santos@nau.edu***]
Carla I. M. Bezerra [**Federal University of Ceara** | ***carlailane@ufc.br***]

Abstract

Software Ecosystems (SECO) are a set of actors and components that work as a unit, which establish relationships based on common interest to provide solutions or services for the software industry. As a company or organization expands its relationships and begins to interact with external actors, a network is formed and SECO includes both the actors and the involved artifacts. However, SECO is not commonly taught in the Software Engineering disciplines. The activity of modeling a SECO can assist in a better visualization and understanding of relationships. However, there is no official modeling standard for SECO, and the notations are quite varied. In this scenario, SSN (Software Supply Network) notation emerged to try to fill this gap. An eminent problem in the literature is the lack of support for modeling using SSN notation, the lack of available models and maintenance of models in general. In this context, this work aims to present a tool to support the teaching and modeling of SECO using the SSN notation, and thus alleviate the problem of the lack of specific modeling tools for SECO. In this work, the tool was applied in Software Engineering classes, where students filled out a questionnaire and a qualitative analysis was performed on the results. In general, the tool pleased both in terms of usability and understanding of SECO. Qualitative analysis revealed that the tool collaborates for SECO modeling, but it can improve usability and design, and there is a need for documentation and support for SECO teaching and modeling.

Keywords: *Software Ecosystems, Software Engineering, Teaching, Tool, Model*

1 Introduction

Software Engineering (SE) community has advanced theoretical and applied research to deal with a range of industry demands. Topics such as Software Product Line (SPL), Systems-of-Systems (SoS), Software Ecosystems (SECO), and Internet of Things (IoT) came to address technical, economic, and social issues of SE (dos Santos et al., 2013).

Software Ecosystems (SECO) is defined as a set of solutions that allow automating activities and transactions by the actors in the associated social or business ecosystem and the organizations that offer these solutions (Bosch, 2009). SECO also consists of a set of actors interacting as a unit with a distributed market between software and services, along with the relationships between the most varied entities (Jansen et al., 2009a). We reinforce that SECO is a software-intensive system (SIS), whose focus is broader, and can aggregate as part of itself, depending on the analysis perspective.

Modeling is one of the essential activities for describing SECOs in a system. It involves different levels of technologies, notations, and abstractions. However, despite the initial advances in SECO research, there are few analytical models, real case studies, and integrated support for tools (Manikas, 2016). Models are part of the state of the art and practice of software engineering (Graciano Neto et al., 2019). Models are built to provide a better understanding of systems or environments (Coutinho et al., 2017b). A significant barrier to the evolution of SECO, in the sense of aiding decision-making in industry, is the lack of support for SECO modeling (Coutinho et al., 2019). Jansen et al. (2015) argued that SECO modeling is essential to provide ideas from representations, in addition to enabling the analysis and comparison of "static" ecosystems based on a key concept (e.g., organizations, relationships, and flows) and existing methods (e.g., socio-technical networks and software supply networks).

A difficulty pointed out by Coutinho et al. (2017a) is that the representation of SECO is still not standardized, so a diversity of notations (e.g., class diagram, flowchart, architectural images) is used for the representation and modeling of a SECO. In addition, the use of modeling tools is essential for a better understanding and overview of a SECO. A way to standardize SECO modeling was suggested by Boucharas et al. (2009), using the SSN notation (Software Supply Network), the most popular notation used by the community of SECO for modeling. SSN is a series of software, hardware and service organizations, which cooperate to meet market demands (Costa et al., 2013). The graphical elements of the notation help to represent the actors and their behaviors and interactions within the ecosystem.

Technological evolution has contributed to the emergence of new paradigms and trends that meet the demands of a dynamic market (Ferreira et al., 2018), and in this context, teaching SE becomes a challenge (Ferreira et al., 2021). The traditional applied methodology (e.g., expository classes, textbooks, and assessments) is often not capable of transferring knowledge about higher education due to difficulties related to reconciling theory and practice, students' lack of interest in purely expository classes, and the process of evaluation not always appropriate or effective (Ferreira et al., 2021). In addition, emerging topics are even more complex to apply in the classroom, with SECO being one of these topics (Ferreira et al., 2018).

Courses related to systems modeling, such as Requirements, Systems Design Analysis (SDA), and Software Engineering (SE), use diagrams to represent information. There are several notations, such as i* and UML, many of which are suitable for modeling SECO. However, some elements and characteristics of SECO can not be represented when applying another modeling language (Coutinho et al., 2017b). For SECO elements that can not be modeled with existing notations, it is noteworthy that the notation can be used, but that it does not cover specific SECO situations. For example: (i) software providers with problems to distinguish specific SECOs in which they are active; and (ii) problems to use this SECO to their strategic advantage, partnership models from the perspective of the participants and how they are affected. Jansen et al. (2015) indicated the lack of a universally accepted set of modeling methods hinders the advancement of SECO research. In addition, the lack of standardization in the notation for teaching SECO makes it challenging to apply in practice, as there is a need for training in notation, occupation in the course load of the discipline for teaching notations, and adequacy of the notation for SECO.

Another situation that occurs for a more effective teaching of SECO is that there is a scarcity of examples of SECO models (Coutinho et al., 2019). In addition to a lack of standardization, images of SECO models, for example in SSN notation, are not so easy to identify. Furthermore, maintaining these models is not an easy task, as platforms evolve, and consequently SECO models change. For the teaching of SECO, examples would help students a lot to have a more applied and real vision. Several systematic mappings and reviews have already been published by the SECO community and few documents dealing with tools were found (Alencar et al., 2020). In this sense, there is a lack of tools that act as a repository of SECO models.

Once SECO concepts are incorporated into the disciplines, a simple analysis of an application can help on teaching SE, focusing on SECO modeling, allowing a global view of suppliers, customers, intermediaries, and your relationships. This broadens the perspective of several factors that can impact software development, including maintenance and evolution. Therefore, the teaching of SECO becomes a transversal concept to the different areas of higher education. Furthermore, given the low dissemination of SECO in higher education, its application is essential in the educational context.

The problem that this work addresses is the lack of support for SECO modeling in teaching. In our case, support is considered the lack of tools that help in specific modeling for SECO, with its own notation and the possibility to extend concepts related to SECO, such as SECO quality, metrics an health. The difficulty in teaching SECO is due to the lack of specific tools, and current notations do not fulfill all needs. Software modeling is an essential aspect in SECO, as it is possible to have a global view of all SE areas and all those involved, such as clients, intermediaries, and suppliers. A modeling tool capable of meeting these requirements efficiently would respond to this gap in the literature about the lack of modeling support.

The general objective of this work is to present a tool to support the SECO teaching and modeling with SSN notation. For this, we developed a web tool. Our tool was used and evaluated in SE courses. A survey was applied and a qualitative analysis was conducted. In general, the tool enabled the understanding of SECO concepts and pleased usability, and the qualitative analysis showed that the tool collaborates for SECO modeling, needing to improve usability, visuals, and documentation, both in the tool and in the notation.

This paper is divided into the following sections: Section 2 presents a theoretical framework on SECO and related works; Section 3 describes the SECO *Modeling* tool; Section 4 describes the methodology applied in this work; Section 5 presents the results, discussions and limitations of the research; and finally, Section 6 presents the conclusions and future work.

2 Theoretical Background

This section presents some concepts about SECO, a brief summary about the SSN notation, some related work, and a brief vision of SECO modeling in industry.

2.1 Software Ecosystems (SECO)

SECO can be defined as a set of actors functioning as a unit that interacts with a distributed market between software and services (Jansen et al., 2009a). These relationships are supported by a technological platform or a common market and carried out through exchanging information, artifacts, and resources. Hanssen and Dybå (2012) defined SECO as a community of actors and organizations in a network, which support their relationships on a common interest in the development and use of a core software technology.

The ecosystem metaphor reinforces external or unknown actors contributing to developing a common technology platform, moving the traditional organization-centric value chain to a software delivery network where multiple components developed on different platforms coexist and affect the buyer's business (Boucharas et al., 2009). Finally, Manikas (2016) updated the definition of SECO as being: "the interaction of software and actor concerning a common technological infrastructure, which results in a set of contributions that directly or indirectly influences the ecosystem".

According to Campbell and Ahmed (2010), SECOs can be seen in three dimensions: (i) Architecture: involves the platform (technology or infrastructure) on which SECO will be inserted, as well as issues of software architecture, Software Product Lines and SE processes; (ii) Business: involves knowledge about the market, decisions that actors must make about business models, the definition of the SECO product portfolio, licensing and sales strategies; and (iii) Social: defines how the network of actors will relate within SECO to achieve its objectives and foster the growth of SECO through a proposal where everyone can obtain gains.

In addition to dimensions, SECO have key roles that are mandatory and essential for an SECO. Table 1 shows each of the roles and their descriptions within a SECO. Some examples of SECOs are MySQL/PHP, Eclipse, Microsoft and iPhone SECO (Jansen et al., 2009a; dos Santos et al., 2013).

These SECO can be closed or open, implying the level of freedom that external organizations and the community have in SECO.

Table 1. SECO roles (Hanssen, 2012)

Role	Description
Keystone	A description or small group that somehow drives the development of the core software technology
End-users	A key role for core software technology as it represents who needs it to run their business, of whatever type
Third-party organizations	Use core software technology as the basis for producing related solutions or services (includes external developers)

2.2 SSN Notation

One of the problems faced in modeling SECO is the lack of standardization. Boucharas et al. (2009) proposed a way to standardize 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 (Costa et al., 2013). SSN represents the main actors and their interaction within a SECO using key elements. Figure 1 shows the elements used in the SSN notation for modeling a SECO.

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.

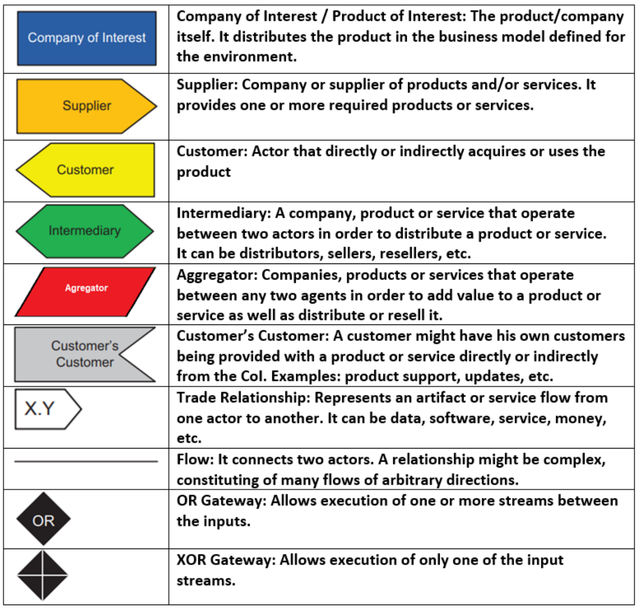

Figure 1. SSN notation (Boucharas et al., 2009) with extension of Costa et al. (2013).

2.3 Related Work

Some works in the literature developed or motivated the use of tools for teaching SECO. However, reports are still scarce. Alencar et al. (2020) identified a series of related work to the teaching of SECO. Among them, a prominent criterion was the lack of specific modeling tools for SECO.

Boucharas et al. (2009) formalized a definition of standards for modeling supply chains and software products, called SSN, describing the relationships between elements. Some uses of the notation were presented with a study of software and third parties in SSN and interviews. The authors reported problems in modeling SECOs.

Coutinho et al. (2019) researched on the SECO teaching in SE, with a qualitative analysis on the experience after the SECO concepts exposition. The idea was that students had a global view of SECO modeling, preferably with SSN. Highlight for the absence of a specific tool for SECO modeling. As results related to modeling, the eminent problem in the literature is highlighted: lack of modeling support, lack of SECO description, and lack of modeling tool. However, the authors concluded that it is beneficial to teach SECO in the higher education discipline, even with the lack of modeling support and SECO modeling tool.

Alencar et al. (2020) analyzed the lack of SECO models in the literature and the impact on the SECO dissemination and teaching. They presented a tool to support the SECO teaching (ARIEL), with model inclusion and query. However, the tool does not have modeling functionality. The work evaluated the user experience. One of the main conclusions was the need for a SECO modeling tool.

Ferreira et al. (2021) presented a digital game to support the SECO concepts teaching (SECO Tree). The game was built following the ENgAGED educational game development process and verified with students using the MEEGA+ assessment model. The dimensions usability, challenge, satisfaction, fun, focused attention, relevance, and learning perception were analyzed. According to the students' perception, the game helped in the teaching-learning process of basic concepts about SECO. For those who do not play so often, they found the game to be appropriate and a fun teaching method.

Amorim et al. (2018) investigated architectural practices aimed at orienting and training newcomers in their first contacts with SECO, particularly open ecosystems. This includes software architecture and how educational artifacts were built to achieve a healthy ecosystem. As results, architectural practices disseminated by some training resources and analysis of how these practices contribute to achieving a healthy ecosystem. A qualitative study with data obtained from different training sources was conducted and the findings showed connection between existing education for newcomers and its potential impact on achieving healthy open source ecosystems.

Table 2 presents a comparison between related work. The comparison criteria are: (i) the work performs SECO modeling, or if in the related work some SECO modeling is done; (ii) the work introduces SECO and SSN notation concepts; (iii) the work uses some approaches to support SECO modeling; (iv) the work uses SSN notation for modeling; and (v) the

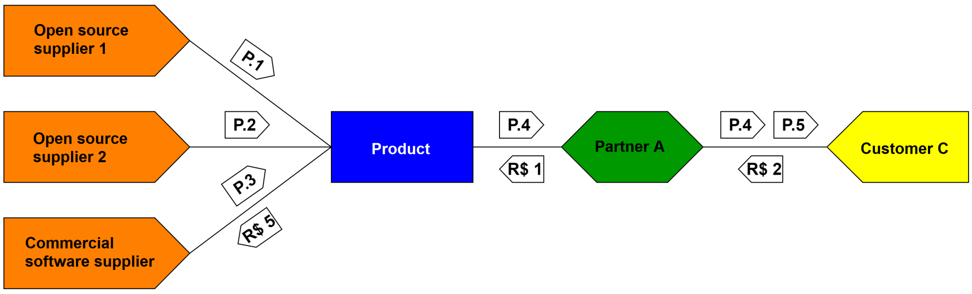

Figure 2. Generic SECO model in SSN notation (Boucharas et al., 2009).

Table 2. Related work comparison

Research	Modeling	Concepts	Approach	SSN	Tool
Boucharas et al. (2009)	Yes	Yes	Yes	Yes	No
Coutinho et al. (2019)	Yes	Yes	Yes	Yes	No
Alencar et al. (2020)	No	Yes	Yes	Yes	No
Ferreira et al. (2021)	No	Yes	Yes	No	No
Amorim et al. (2018)	No	No	No	No	No
This work	Yes	Yes	Yes	Yes	Yes

work models in its tool for SECO or has a modeling tool. The comparison criteria are items considered essential to support SECO teaching in SE. They were based on studies in the literature, which used similar criteria. Most of the related work follow the selected criteria, but it is clear that there is still little support for SECO modeling. It is also noticed that the lack of a modeling tool in SECO is imminent and a tool that supports the diffusion of the literature in the area of SECO. The purpose of our research meets all the criteria.

2.4 SECO Modeling in Industry

SECO is an approach that investigates the complex relationships that exist between companies in the software industry (Barbosa and Alves, 2011). Companies work competitively or cooperatively to achieve their strategic objectives. Thus, they must engage in a new perspective, considering both their own business and the business of others. Inspired by properties of natural and business ecosystems, a SECO encompasses the technical and commercial aspects of software development, as well as partnerships between companies.

The SECO approach can increase the value of core service offerings to existing users, increase the attractiveness of the ecosystem to new users, increase platform adherence, accelerate innovation through open innovation in the ecosystem, and reduce innovation and maintenance costs (Bosch, 2009).

In addition to the benefits of the SECO approach, companies face challenges such as gaining insight into ecosystems, identifying survival strategies within ecosystems and disclosing intellectual property (Jansen et al., 2009b). The competition is usually about who has the best platform strategy and the best SECO to support it (Cusumano, 2010). As the challenges become more understandable and manageable, the realization of benefits becomes more likely. In the literature, there is still a fragmented view of support for management practices for SECO (Viljainen and Kauppinen, 2011).

Software supply networks pose particular risks in the innovation process (Viljainen and Kauppinen, 2011). There are additional challenges in coordinating quality assurance across organizations and synchronizing release time (Jansen et al., 2009b). Often, external development teams can not be subjected to standardized process models, tools and ways of working (Bosch, 2009).

Technology asset management covers how technology is organized, stored and revealed by a company (Viljainen and Kauppinen, 2011). Furthermore, a company can choose to open interfaces offering APIs or open source code such as a requirements engineering process, roadmaps, release times, customer and supplier information, bug repositories or market research (Jansen et al., 2009a).

All these aspects can be minimized with SECO modeling, in the appropriate abstractions. In this way, it is possible to have a global view around the central platform and possible impacts of changes.

3 *The ECOS Modeling Tool*

The *ECOS Modeling* tool was developed according to a client-server architecture, being a web application for modeling SECO with SSN notation designed to fill one of the shortcomings in the literature, which is the lack of a proper environment and support for SECO modeling. The word "ECOS" means "Ecossistema de Software" in Brazilian Portuguese. Its English translation is "SECO".

The technologies Vue.js and Mxgraph.js were used in the development of the tool. Its main features are: create the SECO model using SSN notation, model editing functions (resize, move, copy, paste, remove, group), save/export the model in image formats (PNG, SVG), export the model in formats that allow the model to be imported into the tool for possible changes and/or maintenance (XML, JSON) or in other tools for possible integrations, and printing.

Figure 3 presents the main and unique screen of the SECO modeling tool, proposed with a minimalist and straightforward design, to make it as easy as possible for the user to understand during use. The main screen has a top menu with icons that allow the user to manipulate the components and

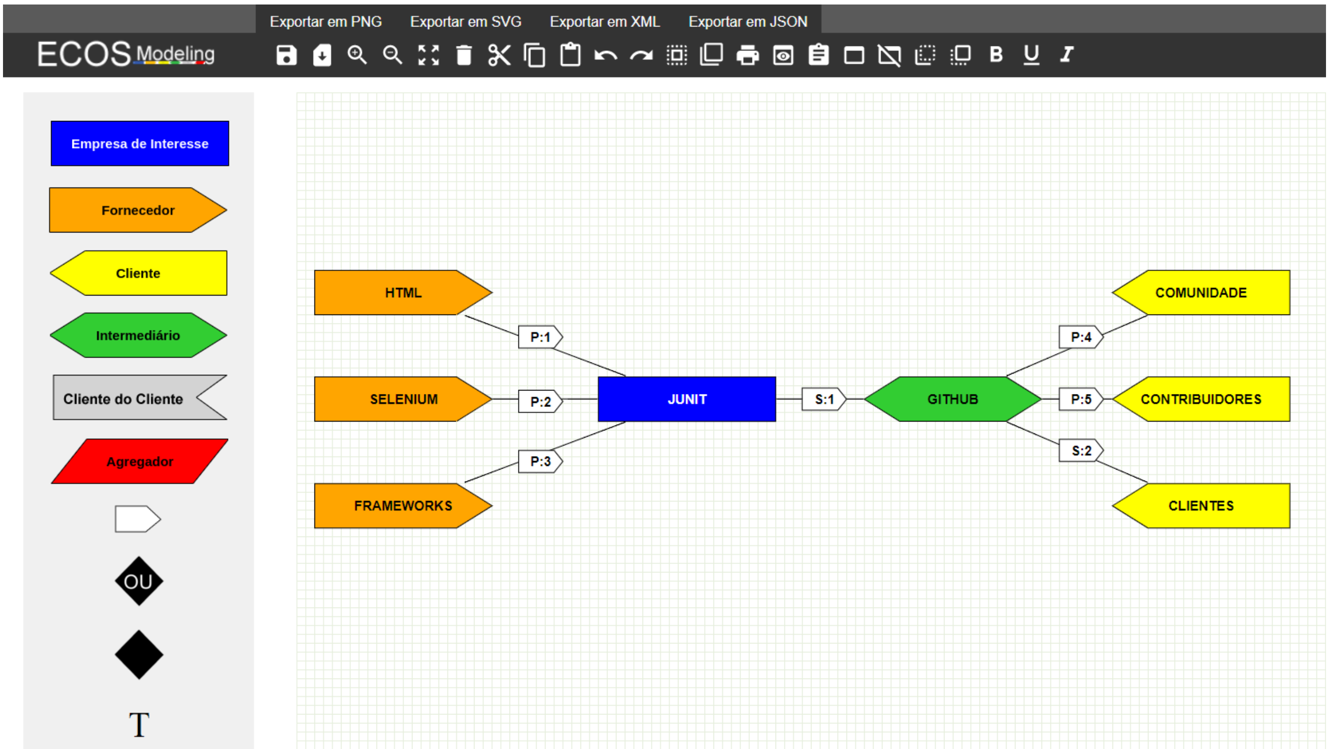

Figure 3. *ECOS Modeling* tool screen (in Brazilian Portuguese).

the created SECO model, in addition to options for exporting the model. The left side menu provides the components of the SSN notation: company of interest, supplier, customer, intermediary, customer of the customer, aggregator, and relationships. In the center is the drawing area, where the user can develop their models and edit them.

The central area is where models are created using SSN notation. The ability to click and drag components facilitates modeling, editing, and model maintenance. Defining relationships between components is also possible. There are no limits to the number of model elements, and the degree of detail is a decision of the modeler.

Figure 3 presents an example of a simplified SECO model for the JUNIT testing tool and a brief analysis. In the example, components from suppliers and customers are arranged around the central platform and an intermediate element and relationships between the elements with products and services that travel on SECO. As providers were modeled HTML, Selenium, and Frameworks. At this level, it is clear that the number of suppliers can be much higher, for example, the number of frameworks that JUNIT uses, but the level of abstraction is a decision of the modeler. You can also model only what is considered necessary for the context of the model. As customers, there are Community, Contributors, and Customers. Again these are generic elements to illustrate, but a particular customer can be directly specified. The importance of the Community element in the model is an aspect to be analyzed for the evolution of SECO JUNIT, as it can be a community of developers, researchers, or industry. As an intermediary, only GITHUB is the actor that makes JUNIT available to the community. Relationships generally have source code or application executables as data to be shared, but it varies a lot depending on the type of actor and the service provided.

A simple analysis like the one previously described can help in teaching both SE and systems analysis, with a focus on SECO modeling, allowing a global view of suppliers, customers, intermediaries and their relationships.

4 Methodology

This work presented the ECOS Modeling tool. For its evaluation, a consultation with students was designed, in order to obtain information about the tool and thus promote its evolution.

We adopted the following steps to meet the objectives: (i) Class on Concepts and Modeling in SECO; (ii) SECO Modeling Exercise with the Tool; (iii) Application of the Evaluation Questionnaire; and (iv) Consolidation and Analysis of Results.

4.1 Class on Concepts and Modeling in SECO

We conducted a class on SECO topics in SE subjects in this step. The purpose is to expose to students concepts and definitions related to SECO and modeling, specifically in the SSN notation, its applications in general, and its importance in the educational context, given the low dissemination of SECO in education. The *ECOS Modeling* tool is presented to the students, with the exposition of the functionalities and modeling of a SECO in SSN.

4.2 SECO Modeling Tool Activity

Students were asked to model a SECO following a previously established script. The script is presented in Table 3. It was composed of a sequence of tasks to be performed in the tool.

Basically they are tasks for creating a model in SSN notation and tasks for testing the other functionalities of the tool. All students verify that the tool's functionalities are working correctly. The models generated by the students must meet the prerequisites of the SSN notation; that is, the user must model a SECO in the proposed tool, using the notation correctly, listing the actors, their roles, and relationships.

Table 3. Tool evaluation script

1. Create an SSN model of anything you think is necessary, analyzing the modeling and relationships between components of the notation. For that:
a. Drag the figures from the sidebar to the drawing area, and check that the figures in the sidebar match the figure dragged to the drawing area.
b. Make connections between the components, pass the mouse arrow over a component, check if an arrow appears in the center of the figure, click and drag to another figure.
c. Put the business relationship between the binding of the components.
d. Change the component label by double-clicking on the figure label.
2. Testing the features:
a. Copy, paste, cut, delete the component, both with the menu icons and with the keys (ctrl C, ctrl V, ctrl X, Delete).
b. Undo and redo with both menu icons and keys (ctrl Z and ctrl Y).
c. Zoom in, zoom out, current zoom both on the menu icons and on the keys (ctrl 0 - current zoom), mouse down and up (zoom in and out) and the respective icons in the menu.
d. Select all objects, deselect, group and ungroup on menu icons.
e. Place the figure behind or in front of the other in the respective icons in the menu.
f. Change the stylization of the figure label, text in bold, italics and underline in the respective icons in the menu.
g. Save the created SSN template in the formats available in the tool: PNG, SVG, XML and JSON.

4.3 Survey Application

In this step, students answer a survey with questions related to the respondent's profile and the experience of using the tool. The objective is to evaluate the features and functionalities that the modeling tool proposes and to measure the user experience. The survey was divided into 4 sections: free and informed consent form, demographic data of the participants to obtain the profile of those who participated in the study, technical questions about the user's experience in using the tool, and open questions to investigate participant's opinion and suggestions for improvement.

At first, the students who would answer the questionnaire would be students from the SE class, due to the fact that they already have knowledge in systems modeling because of the curriculum. This course is part of the curriculum of several courses on campus, so it was assumed that all of them attended the prerequisites. The survey was planned so that the student had a first contact with the tool. Once he had a foundation in SECO, due to the previously taught class, he would try out the tool following a previously defined script that would cover modeling and editing functionality. All students from some classes were invited to respond to the survey, considering that it was a planned class with the propfessors of these classes, and that they should practice.

Although there is content exposure, strictly speaking this work would not be a case study, as there is no related hypothesis, and no debate between student and teacher was stimulated, as only data were collected through of the questionnaire. Even with the use of the tool by the student, the modeling of a specific situation was not part of the script, which was free for the student. Also, no methodology for the notation or tool was proposed, only its use. Due to these aspects, we consider it as a survey. The survey is characterized by the direct questioning of people whose behavior you want to know (Gil, 2022). Basically, information is requested from a group about a problem, followed by a qualitative analysis to obtain conclusions corresponding to the collected data.

4.4 Analysis of the Results

The results obtained from the assessment carried out by the students are consolidated and analyzed quantitatively and qualitatively. In the quantitative analysis, descriptive statistics were used to represent and describe the data characterization of the participants.

For the qualitative analysis, procedures of the Grounded Theory (Corbin and Strauss, 2014) method were used, inspired by the procedures presented by Ferreira et al. (2018). Grounded Theory aims to create a theory from systematically collected and analyzed data, is composed of three phases: (1) open coding, (2) axial coding, and (3) selective coding. In open coding, data is broken down, analyzed, compared, conceptualized, and categorized (Corbin and Strauss, 2014). In the first phases of this codification, meticulous reading of the collected answers was carried out, and each relevant fragment receives an expression, phrase, or word, forming codes and categories. In axial coding, categories are related to their subcategories, forming more dense, developed, and related categories. And finally, the selective coding phase originated the category or central idea of the study, that is, the central category of the theory in which all are related.

The coding process ends when no new data adds new knowledge to the categorization analysis process. Despite the Grounded Theory proposal being theory building, Strauss and Corbin explain that the researcher can use only a few steps to reach his research objective (Corbin and Strauss, 2014). For example, when researchers need to understand a particular phenomenon or situation. In this research, only phases 1 and 2 of the coding were used to analyze the data to identify emerging topics, difficulties, strengths, and weaknesses. This type of analysis involves researcher creativity, experience, and bias. Because of this, we sought to verify the analyses carried out, which were conducted by another researcher, to increase the quality of the results.

5 Results, Analysis and Discussions

The *ECOS Modeling* tool evaluation was carried out from a survey with students of the SE courses belonging to undergraduate and graduate courses in computing at the Federal University of Ceará, Campus Quixadá, to provide students an overview of SECO concepts and modeling using the SSN notation. The survey was applied in March 2021 over two weeks. A total number of 17 respondents was obtained, 11 undergraduate students and 6 graduate students.

Concepts related to SECO, its applications in SE, concepts related to SSN notation, and examples of SECO modeling using the suggested notation were discussed during the class. Finally, the modeling of a SECO was carried out in the proposed tool to demonstrate the functionality and put into practice the subject addressed during the class.

After the presentation, the students answered an online survey (Table 4). The survey questions were divided into four sections, the first presented the consent form, the second addressed demographic data (D), the third was composed of technical questions (T), and the fourth and last section consisted of open questions (O).

Demographic questions were relevant to be able to profile the participating students, with information about their course, entry semester, whether they have already taken the SDA course. Moreover, questions about the use of modeling tools in general, about the level of knowledge in modeling before the SECO class, they had already heard something about SECO, concepts, and applications.

The technical questions address topics about the degree of difficulty and satisfaction of users in using the tool, if the user would recommend the proposed tool for other people to know and/or model if the look of it is pleasant and easy to use if it is useful, if it meets the requirements it proposes to implement and what is the level of mental effort of the user in carrying out a SECO modeling. These questions have the objective of verifying if the tool meets what it proposes if the user feels comfortable using it, and the degree of mental effort during the modeling.

The open questions are related to the students' opinions about the strengths that the user highlighted in the modeling tool, the weaknesses, and possible suggestions for improvement for the proposed tool to perceive potential inconsistencies that were not captured in the previous questions.

The survey also aimed to draw a profile of the participants about their level of knowledge concerning tools and systems modeling, to verify if the tool meets the problem that it proposes to solve efficiently and correctly, if the functionalities implemented in the tool are consistent, and analyze/perceive users' experience in using it, seeking consolidation.

Table 4. Survey Questions (demographic, technical, opinion)

ID	Description
D01	What is your undergraduate or postgraduate course?
D02	What is your entry semester for the course?
D03	Have you already taken the SDA course (or equivalent)?
D04	Have you ever used any systems modeling tool?
D05	How knowledgeable do you consider yourself to be in systems modeling?
D06	Before the presentation, had you heard about Software Ecosystems?
T01	How difficult is it to use the tool?
T02	How satisfied are you with using the tool?
T03	Would you recommend the tool to someone who wants to know / model Software Ecosystems?
T04	Does the tool's interface look nice?
T05	What is your level of mental effort to perform a model?
T06	Is the tool useful to the user?
T07	Does the tool meet all the requirements it proposes?
O01	What are the strengths of the tool?
O01	What are the weak points of the tool?
O03	What suggestions for improvement for the tool?

5.1 Demographic Questions

The objective of the demographic questions was to obtain data from the participants' profile (Figure 4).

D01 and D02 correspond to the participants' course and year of entry. Most of the participants were from the postgraduate course in Computer Science and the Information System (IS) course, followed by the Computer Science (CS) and SE courses. Although the research was applied in the SE course, enrollment is open to other courses, so it is common to have CS and IS students. Most students joined their respective courses in the first semester of 2018.

The remaining questions correspond to the participants' modeling and modeling tools skills. D03 displayed results about whether students took the SDA course or equivalent, with 94.1% answering yes and 5.9% no. In general, SDA is a prerequisite of SE. D04 showed results on the students' experience using systems modeling tools, with 88.2% answering yes, who have already used some modeling tool and 11.8% answering no. These questions verified if the students have any previous knowledge about the analysis and design of systems and modeling. According to the obtained results, we can observe that most students both attended the course and already used some modeling tool. D05 presented the results on the level of knowledge in systems modeling, and it was observed the majority answered that they understand little or are neutral concerning the subject. D06 showed results on whether students had already heard some concept or application of SECO to verify if they had already seen or studied something related to SECO, with 76.5% of students responding no and 23.5% answering yes.

Specifically concerning to SDA, it was found that only one student had not attended SDA, and this student was a master's student. The student's course was Computer Networks, and there is no SDA in its curriculum. However, the student pointed out the easy difficulty level. In addition, 4 students reported difficulties, 2 from the master's degree (e.g. 1 from IS and 1 from SE) and 2 from the undergraduate degree. The profile of the 24% of students who found it difficult to use the tool corresponded to 2 students from the master's degree in computing, 1 from IS, and 1 from SE. All coursed SDA, all with knowledge and modeling tools, 3 with level understand little modeling, and none had heard of SECO.

5.2 Technical Questions

Technical questions aimed to capture feedback from students about tool features, SECO modeling, and their experience/satisfaction in using the proposed tool (Figure 5).

T01 and T02 correspond to the difficulty and satisfaction degree in using the tool. The degree of difficulty in using the tool was 29.4% answering very easy or easy, 23.5% answering that it was "somewhat easy" and 17.6% neutral. The

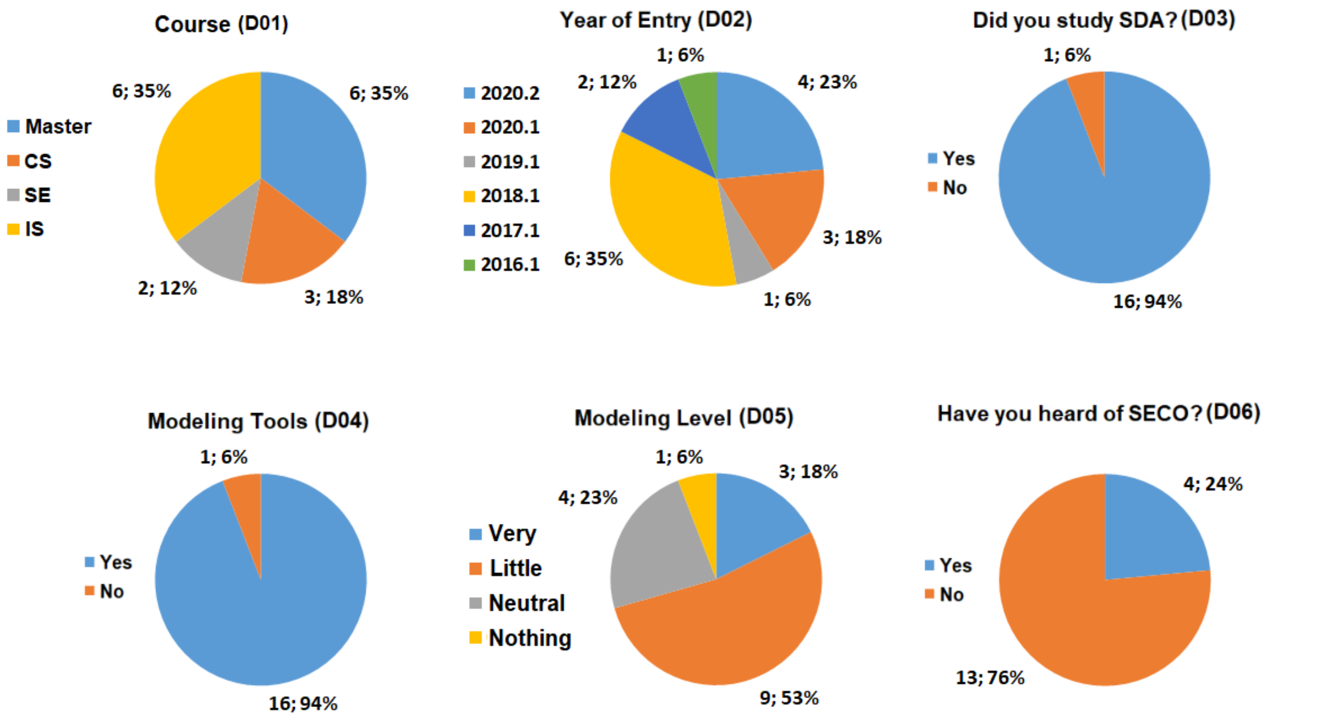

Figure 4. Demographic Questions.

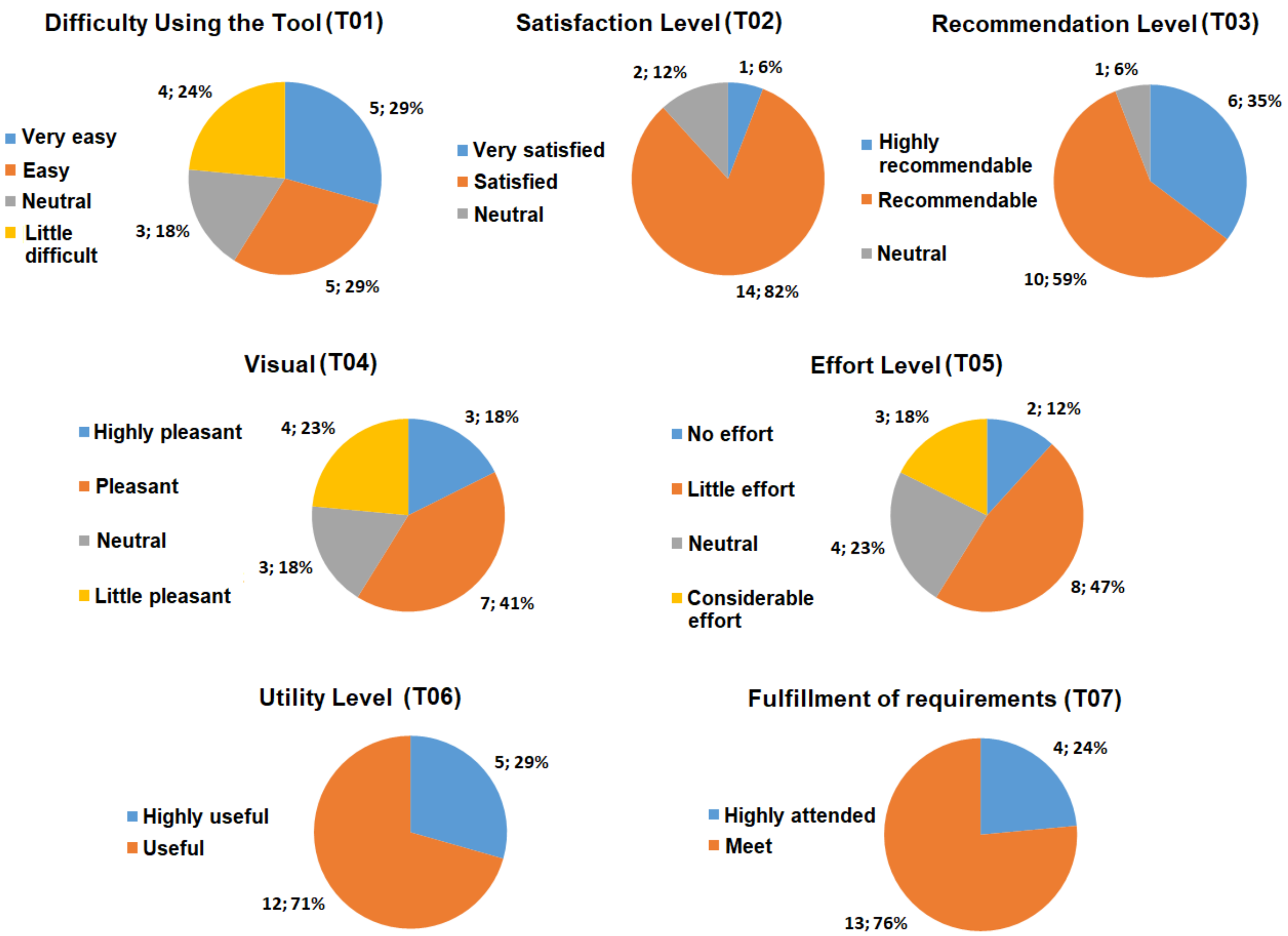

Figure 5. Technical Questions.

degree of student satisfaction in using the tool was 82.4% satisfied, 5.9% responded very satisfied, and 11.8% were neutral.

For the recommendation level (T03) of the tool to someone interested in knowing and/or modeling a SECO, 56.3% responded the tool is recommended, 37.5% responded it is highly recommended and 6.3 % were neutral.

T04 showed how pleasant the tool looks, with 37.5% responding that the interface is pleasant, 18.8% answered it was highly pleasant, 25% answered it was not very pleasant, and 18.8% were neutral. The user's level of mental effort when performing modeling in the tool (T05) showed that 43.8% of the participants responded they needed a little effort, 18.8% answered they required considerable effort, 12.5% required no effort, and 25.0% were neutral. This result indicates that it takes a little mental effort to model SECO and that their application is not trivial.

T06 reflected the degree of usefulness of the tool for the user, with 68.8% of the students responding that the tool is useful and 31.3% answered it is highly useful, a promising result to consolidate the tool. T07 indicated the degree of compliance with the requirements proposed by the tool, with 81.3% of the participants responding that the proposed requirements were highly met and 18.8% answered they were met. This result is essential for the research, as it concerns the proposed tool's functionalities to alleviate the lack of support for SECO modeling in the literature.

The results showed the tool does what it proposes to do. It deals with the previously mentioned problem, which is the lack of support for SECO modeling in the literature, necessary for an adequate application for teaching. Another critical point is the evolution of the proposed tool, as it lacks some complementary functionalities to be implemented to improve it, expand it, and make it available in the literature. In general, the students evaluated the tool well regarding its features, interface, usability, and level of mental effort to create SECO models and their degree of satisfaction in using it.

5.3 Open Questions

For the qualitative analysis, we used Grounded Theory procedures. We also chose to display more generic, larger categories, just to illustrate the relationship between the smaller categories, which were the ones that were actually identified by reading the students' responses. In this way, we generated three figures (Figures 6, 7 e 8), where each one is representing the strengths, weaknesses and suggestions of tool improvement. The relationships between the smaller categories reveal the use of the tool by the students. As a consequence of this choice, the figures have few categories and are somewhat generic.

During the qualitative analysis, 8 categories were identified, with the following frequency of citations in descending order: Visual (33), Functionality (22), Modeling (10), Documentation (8), Comprehension (7), Characteristic (6), Product (4), and Technical (3). Table 5 displays the description of each category. Seven relationships between categories were identified. Figures 6, 7 e 8 show categories and relationships. The dashed line has no semantics, just to reinforce the meta categories (strengths, weaknesses or improvements) and not relationships between categories. Some quotes from participants were captured to highlight aspects of the opinion questions, with participants identified from P1 to P17 to preserve anonymity.

Table 5. Categories identified in the qualitative analysis

Categories	Description
Characteristic	any quality or defect of the tool
Comprehension	indication of goal or understanding
Documentation	user help on the tool and SECO theory
Functionality	presence or absence of functionalities, with or without defects
Modeling	SECO general modeling elements
Product	technical feature of the tool
Technical	technical characteristics related to hardware or software
Visual	tool design and usability elements

5.3.1 Tool Strengths

The categories identified as strengths were: Characteristic, Comprehension, Modeling, Product, and Visual. Figure 6 displays the categories and relationships for the strengths.

Features such as modeling cause emerged as something that the tool had that provided a better, more fluid, and adequate modeling to the idea of SECO. **P13** highlighted this relationship with "*Practicality, in the sense that it already provides modeling options already according to the SSN standard; simple and easy to use*".

Comprehension associated with the visual conveys the idea that the interface helped both in understanding and in the use of the notation, highlighted by **P14** in "*the interface components are easy to understand as well as the identity of the process components*". Also, comprehension associated with modeling was presented as an ease of the tool to support the creation of SECO models, lack of literature, reinforced by **P14** with "*In my opinion the tool completely meets (at least as far as I know) the issue of modeling software ecosystems*".

The tool's interface contributed to better modeling, because the ease of use or by the simplicity of the components, pointed out by **P6** with "*I found the interface very simple, it helps when modeling*" and by **P7** in "*Very interactive interface, with several modeling options*".

In order to expand the strengths identified in the qualitative analysis, the tool can increase modeling functionality, either for better visualization and use, or by adding metadata or attributes to the elements of the SSN notation.

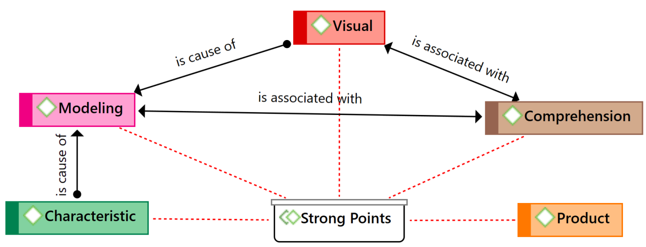

Figure 6. Categories and relationships for tool strengths.

5.3.2 Tool Weaknesses

The categories identified as weak points were: Documentation, Functionality, Modeling, Technical and Visual. Most of the mentions were about the features and the look of the tool. Figure 7 displays the categories and relationships for weaknesses. As many students were unfamiliar with SECO and SSN, documentation issues with the notation were reported.

Features associated with the tool's interface were reported by the respondents, mainly about design issues and small details that could interfere in modeling. However, most of the comments were about improvements to the tool, not a weakness in itself. **P14** highlighted the aspect of design with "*The tool has some design flaws such as the fact that in some cases, the connector does not touch the edge of the component despite the connection being made, in addition, the mouse select function does not display the selection area when pressing and holding the left mouse button and finally, in the side box where the components are, a line appears at the bottom.*".

Also the functionalities associated with the documentation were commented, mainly in relation to the lack of knowledge of SECO concepts because they are not commonly taught in SE classes. **P2** highlighted the lack of documentation for the tool in "*It doesn't have a help function that takes the documentation and how to use it*" and **P7** highlighted the lack of documentation of the notation with "*I believe there was a little bit of specification about each graphic on the left side, and in the formatting, when creating an entity for example, the name is very minimized relative to the size of the square*". The documentation was also related to modeling, where it was noticed there was a need for more concepts about SECO or more classes with modeling practices, mentioned by **P16** with "*I can't identify well, because the difficulties I had were in relation to ecosystem modeling itself (knowledge in the area) and not about the tool*".

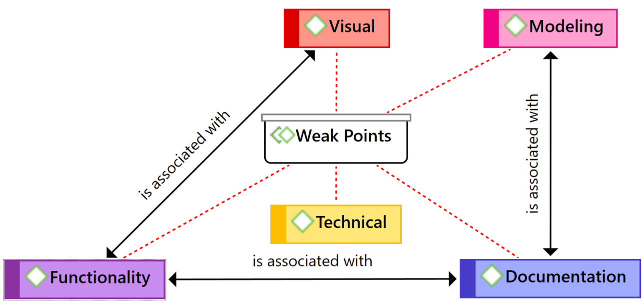

Figure 7. Categories and relationships for tool weaknesses.

To improve the use of the tool, a usability study would be interesting. An improvement in the design and refinement of some functionality can promote a better use of the tool. In addition, including tips on how to use and document SECO in the tool itself can contribute to teaching and modeling in SECO.

5.3.3 Improvement Suggestions

The categories identified to improve the tool were: Documentation, Functionality, Modeling, and Visual. Figure 8 displays the categories and relationships.

Many suggestions were concerning the interface's look, as highlighted by **P10** with "*Place a balloon informing the description of the element, theme options (it doesn't matter so much), improving the tool's visuals, and better organizing the items within the screen for the user to drag*". It is natural to propose improvements in the tools, and as the students' experience was much more in use than in the concepts themselves, many suggestions were in this line, such as the comment of **P4** with "*More number of entities possible. This can be partially solved through the possibility of changing the colors of the entities*". However, comments of this kind hurt the SSN notation.

Only one relationship stood out for the improvement suggestions, which was the association of the documentation with the modeling, highlighted by **P13** "*As the Software Ecosystem idea is still new, for me at least, and despite the class, I received from the professor on the subject, when I stop in front of the software I have a little difficulty in knowing what each of the elements means, I don't know if it is an improvement of the system itself, but if I could somehow provide on-screen information about what each element should mean in modeling, with some examples I think would make it even easier to use*".

Documentation for those who does not know the SSN notation and mainly examples are needed for a better application in the classroom. This aspect became clear from the comments.

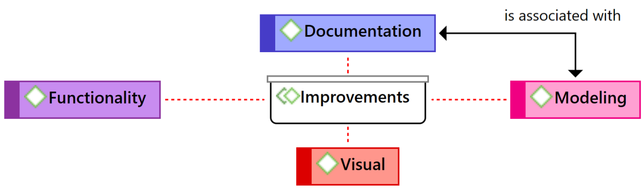

Figure 8. Categories and relationships for tool improvement.

Suggestions for improving the tool's interface, facilitating a better understanding of the components and functionalities, were identified in the analysis. This could be a consequence of the weaknesses, as there was a coincidence with the improvement of the visuals and documentation. As one of the research ideas is to promote the dissemination and teaching of modeling in SECO, these aspects are essential for a better understanding and productivity.

5.4 Discussion

The students' experience after using the tool and the survey filling made it possible to publicize the SECO area and obtain several feedbacks from the tool and notation.

For the SE course, SECO usually is not taught. Regarding the question if the students before the SECO class already knew the subject, where only 4 answered yes. This result is understood (as expected) because the contents about SECO do not belong to the SE disciplines syllabus. For this reason, it is not widespread among undergraduate and graduate students in computing. Given the limited dissemination of the literature, this experience also aimed to present SECO to students, their applications and concepts, and their use and importance in the academic and business spheres. By presenting SECO concepts and applying them in the tool, there is the benefit of having a global view of the elements around a central platform or company of interest, component of the SSN notation. As there are suppliers, intermediaries, and customers, a view of the relationships between SECO actors can describe the flow of information, dependencies, and an impact analysis of changes and the entry or exit of participants. This view is not normally presented in the SE course. The qualitative analysis had some indications that SECO concepts are interesting, but there is a need to go deeper.

Most of the students who answered the survey were students of the master's degree in computing, where they have a more significant academic maturity and have already taken modeling disciplines in general. There was a greater acceptance and less difficulty in using the modeling tool. A relevant result for the tool was about use satisfaction and visual. Most students responded they were satisfied with using the tool, and most also responded the tool's interface is useful and pleasant for the user. These results indicated despite requiring usability improvements, the tool can motivate its use for SECO teaching. Further, 100% of the answers were the tool is highly useful or useful and that the requirements and functionalities proposed by it are fully met, being of value to the literature.

Regarding the open questions about the tool's strengths, weaknesses, and improvements, some quality points of the tool were reported, such as: what should be maintained and/or improved later, corrections, interface, and suggestions for implementation of more features to enhance the tool. It was also noticed that, in general, the tool is useful for users, being easy to use, pleasant interface and that it meets all the requirements that it proposes to do, listing only a few points for improvement, to facilitate the user during use and achieve the expansion of the literature, both in educational and in research and modeling work.

The tool intends to be freely available, thus providing its growth together with the community, to expand the number of research related to the literature, explicitly concerning SECO modeling. The results indicated the proposed tool partially addresses the aforementioned problem of lack of support for SECO modeling, that is, lack of a modeling tool that provides the SSN notation correctly in its purpose, allowing the scientific community to have its environment of SECO modeling. For full service, the qualitative analysis suggested the tool can improve usability and visual, and there is a need for documentation and support in the teaching of SECO and modeling. It is to be expected this need for support conceptually and in notation occur, considering that SECO is not much explored in SE disciplines, and with the dissemination of the SECO area, this conceptual problem can be minimized.

From the viewpoint of the results identified by the qualitative analysis, the tool can improve in some aspects, mainly in the improvement of the interface, facilitating a better understanding of the components and functionalities, and documentation. These aspects would make it possible to improve the use of the tool both for research on SECO and for teaching in a more practical and applied way.

The tool may have some implications for practitioners in the field of SECO, both researchers and teaching. For SE professors, a tool helps in understanding concepts, practical application, and contextualizing scenarios, in addition to promoting different types of exercises for students. For students, it is more motivating to know and practice a particular subject when it has tool support, enabling better learning and practices. For researchers in SECO, with a free tool, new research and collaborations can arise. The aggregation of new SECO concepts and functionalities, such as quality criteria for SECO, a SECO health view, SECO simulation, and model formalization, can promote much research and collaborations between researchers.

As theoretical implications of the research, there is scope for integration and collaboration with different areas of SE. Considering the evolution and maintenance of systems, the SECO area provides a global view and relationships between the components of a system or company of interest. This makes it possible to evaluate impacts, costs, effects of the entry and exit of suppliers and customers. Aligned to other areas such as SPL and SoS, modeling can make changes easier to understand. More technological areas such as Cloud Computing and Internet of Things can also benefit from SECO. As they are areas very close to the industry, the possibility of studies and analysis is high, and modeling a SECO of platforms can collaborate for a global vision of use, business possibilities and integrations between companies.

As the scope was only to evaluate the tool, no post-use evaluation was carried out to assess learning in SECO. This has been included as future work. However, it was noticed that some students were interested in the area, which was one of the research objectives (dissemination of the SECO area).

5.5 Research Limitations

This research may be affected by several factors that may invalidate its main conclusions, consisting of threats to the validity (Wohlin et al., 2012) and limitations of the work, presented below.

The number of responses was small concerning expectations, even with several requests to students. This harms the potential of generalization of results because with more answers we could get more feedback about the tool and the students' opinion about SECO. One aspect that may influence the quality of the answers was the lack of experience of those who answered, as SECO was not known to most students, which may have affected the quality of the answers. In general, SECO is not part of the SE course curriculum. One way to minimize this aspect, and one of the objectives of this research, is the dissemination of SECO with lectures and minicourses. Some questions were subjective and open, so the answers were not necessarily with good quality or adequated. To minimize this effect, different researchers reviewed the qualitative analysis.

The survey was applied in only two class, and this was not enough to generalize results, given that some class had few answers. As the use of the tool was not so intense, consisting only of one practice, it was not possible to explore real models. Consequently, it was not possible to deepen the knowledge of SECO or evaluate the learning. The ideal would be to apply SECO concepts throughout the course, taking advantage of its characteristic of the transversality of concepts.

An observation about the students is that they had no experience in SECO, as expected. This was in fact reinforced by the demographic question D06 "Before the presentation, had you heard about Software Ecosystems?". In this aspect, the evaluation of the tool at a deeper level of SECO and its modeling in SSN would not be possible, given the inexperience of most students. One way to avoid this situation for future editions of research or training of specialists in SECO would be its introduction into disciplines such as Requirements, Systems Design Analysis (SDA), and Software Engineering (SE), and thus create the culture of SECO.

The SECO learning was not evaluated, as the main focus was on the tool. This does not invalidate the work, as an adequate tool facilitates a better understanding and application of the concepts. A more detailed study should be designed and applied to more classes with a greater theoretical and practical workload for SECO in the SE discipline to assess this aspect.

6 Conclusion

This work aimed to present a tool to support SECO teaching, enabling the modeling of SECO according to the SSN notation. Thus, the need for SECO models and material for study and teaching can be minimized. The proposed modeling tool was evaluated in the classroom by undergraduate and graduate students using the tool, application of a survey, and qualitative analysis with Grounded Theory. Students considered the tool useful for teaching and modeling SECO, although few knew the concepts. The results showed that the tool meets what it proposes to do, that is, to support the modeling of SECO.

The main conclusions of the qualitative analysis were: (i) the tool collaborates for SECO modeling but can improve usability and visuals, and (ii) there is a need for documentation and support in SECO teaching and modeling. Regarding the contribution of the tool and study to the scientific community, it is dealing with a gap identified in the literature, which is the lack of SECO models and a SECO modeling tool with the SSN notation. All these items are necessary for teaching SECO. Thus, education in SECO can be better worked in the classroom and broaden the global view of the other macro activities of SE.

As future work, we intend to evolve the tool, considering other specified functionalities. One of the intentions is to invest in SECO health-related functionality and SECO quality metrics. The tool is available to the SE community, which can collaborate with its expansion through new functionalities or an application, especially for teaching and modeling SECO. It is also intended to reapply the modified tool in other disciplines and obtain results about learning in SECO and its impacts on SE. Finally, provide a base of SECO models built in the tool to support SECO teaching in SE.

Acknowledgements

The present work was carried out with the support of the Coordination of Higher Education Personnel Improvement (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) - CAPES - Brazil - Process Number 88887-617302/2021-00.

References

Alencar, I. R., Coutinho, E. F., Moreira, L. O., and Bezerra, C. I. M. (2020). A tool for software ecosystem models: An analysis on their implications in education. In *XXXIV Brazilian Symposium on Software Engineering*, New York, NY, USA. ACM.

Amorim, S. d. S., McGregor, J. D., de Almeida, E. S., and von Flach G. Chavez, C. (2018). Educating to achieve healthy open source ecosystems. In *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, ECSA '18, New York, NY, USA. Association for Computing Machinery.

Barbosa, O. and Alves, C. F. (2011). A systematic mapping study on software ecosystems. In *Proceedings of the Third International Workshop on Software Ecosystems - IWSECO@ICSOB*, volume 746 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Bosch, J. (2009). From software product lines to software ecosystems. In *Proceedings of the 13th International Software Product Line Conference*, SPLC '09, page 111–119, USA. Carnegie Mellon University.

Boucharas, V., Jansen, S., and Brinkkemper, S. (2009). Formalizing software ecosystem modeling. In *Proceedings of the 1st International Workshop on Open Component Ecosystems*, IWOCE '09, pages 41–50, New York, NY, USA. ACM.

Campbell, P. R. J. and Ahmed, F. (2010). A three-dimensional view of software ecosystems. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 81–84, New York, NY, USA. ACM.

Corbin, J. and Strauss, A. (2014). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, 4 edition.

Costa, G., Silva, F., Santos, R., Werner, C., and Oliveira, T. (2013). From applications to a software ecosystem platform: An exploratory study. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems*, MEDES '13, pages 9–16, New York, NY, USA. ACM.

Coutinho, E. F., Santos, I., and Bezerra, C. I. M. (2017a). A software ecosystem for a virtual learning environment: Solar seco. In *Proceedings of the Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems*, JSOS '17, page 41–47. IEEE Press.

Coutinho, E. F., Santos, I., Moreira, L. O., and Bezerra, C. I. M. (2019). A report on the teaching of software ecosystems in software engineering discipline. In *XXXIII Brazilian Symposium on Software Engineering*, SBES 2019, New York, NY, USA. ACM.

Coutinho, E. F., Viana, D., and dos Santos, R. P. (2017b). 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)*, MISE '17, pages 47–53.

Cusumano, M. (2010). Technology strategy and management the evolution of platform thinking. *Commun. ACM*, 53(1):32–34.

dos Santos, R., Werner, C., Alves, C., Jorge Santos Pinto, M., Cukierman, H., Oliveira, F., and Tania Cohen Egler, T. (2013). *Ecossistemas de Software: Um Novo Espaço para a Construção de Redes e Territórios envolvendo Governo, Sociedade e a Web*, pages 337–366. Letra Capital.

Ferreira, T., Viana, D., and dos Santos, R. P. (2021). Árvore de ecos: Um jogo para ensino de conceitos de ecossistemas de software. *Brazilian Journal of Computers in Education - RBIE)*, 29:273–300.

Ferreira, T., Viana, D., Fernandes, J., and Santos, R. (2018). Identifying emerging topics and difficulties in software engineering education in brazil. In *XXXII Brazilian Symposium on Software Engineering*, SBES '18, page 230–239, New York, NY, USA. Association for Computing Machinery.

Gil, A. C. (2022). *Como elaborar projetos de pesquisa*. Atlas, 7a. edition.

Graciano Neto, V. V., Basso, F., Pereira dos Santos, R., Bakar, N. H., Kassab, M., Werner, C., Oliveira, T., and Nakagawa, E. Y. (2019). Model-driven engineering ecosystems. In *2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems (SESoS) and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES)*, pages 58–61.

Hanssen, G. K. (2012). A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. *Journal of Systems and Software*, 85(7):1455–1466.

Hanssen, G. K. and Dybå, T. (2012). Theoretical foundations of software ecosystems. In *IWSECO@ ICSOB*, pages 6–17.

Jansen, S., Brinkkemper, S., and Finkelstein, A. (2009a). Business network management as a survival strategy: A tale of two software ecosystems. In *First International Workshop on Software Ecosystems (IWSECO-2009), colocated with the 11th International Conference on Software Reuse*.

Jansen, S., Finkelstein, A., and Brinkkemper, S. (2009b). A sense of community: A research agenda for software ecosystems. In *2009 31st International Conference on Software Engineering - Companion Volume*, pages 187–190.

Jansen, S., Handoyo, E., and Alves, C. (2015). Scientists' needs in modelling software ecosystems. In *Proceedings of the 2015 European Conference on Software Architecture Workshops*, ECSAW '15, New York, NY, USA. Association for Computing Machinery.

Manikas, K. (2016). Revisiting software ecosystems research: A longitudinal literature study. *Journal of Systems and Software*, 117:84 – 103.

Viljainen, M. and Kauppinen, M. (2011). Software ecosystems: A set of management practices for platform integrators in the telecom industry. In Regnell, B., van de Weerd, I., and De Troyer, O., editors, *Software Business*, pages 32–43, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media, Springer Heidelberg New York Dordrecht London.