A Proposal for Software Ecosystems Engineering

Rodrigo Pereira dos Santos and Cláudia Maria Lima Werner

System Engineering and Computer Science Department
COPPE/UFRJ – Federal University of Rio de Janeiro
P.O. Box 68511 – Rio de Janeiro, 21945-970, Brazil
{rps, werner}@cos.ufrj.br

Abstract. Software Ecosystems (SECOs) have emerged as an approach to improve software reuse in industry considering relations among companies and stakeholders. Companies and organizations have opened up their platforms and artifacts to others, including partners and third-part developers around the world. This changes the traditional software industry because it requires mature research in software architecture, component-based software engineering and software product line in a market and business environment. In this sense, this paper presents an initial proposal for SECOs engineering in order to outline a set of steps that combines three different dimensions of SECOs and joins different perspectives in SECOs research literature through a survey. In this paper, the focus is on the first dimension, that is, architecture. A preliminary analysis done by the Brazilian Software Reuse Lab's SECOs at COPPE/UFRJ points out that several concepts presented at IWSECO 2009 and IWSECO 2010 can be connected in a broader SE approach.

Keywords: Software Ecosystems, Component-based Software Engineering, Value-Based Software Engineering, Software Reuse.

1 Introduction

Software Ecosystems (SECOs) represent a phenomenon in the Software Engineering (SE) field considering their rapid evolution in this decade, though the first researches in this topic were done by Business Schools in the 90's [18][19]. SECOs studies in the SE community were motivated by the software product lines (SPLs) approach aiming to allow external developers to contribute to hitherto closed platforms [4]. However, different research directions indicated by literature and industrial cases reinforce a lot of important perspectives to be explored, such as architecture, social networks, modeling, business considerations, mobile platforms and organizational-based management [14]. Besides, SECOs need a multidisciplinary treatment, including Sociology, Communication, Economy, Business and Law.

These studies are also motivated by the software vendors' routine since they no longer function as independent units that can deliver separate products, but have become dependent on other software vendors for vital software components and infrastructures, for example, operating systems, libraries, component stores, and platforms [6]. So, software vendors resort to virtual integration through alliances to create and keep networks of influence and interoperability, generating SECOs. Nevertheless, some challenges are emerging in this process [15]: (i) software vendors have to be aware of SECOs; (ii) they want also to be aware of survival strategies that exist among SECOs stakeholders; and (iii) they need an overview of possible ways to opening up the organization's platform without exposing the intellectual property.

In order to understand these challenges, Jansen *et al.* [15] model SECOs in a three-level perspective. At the first level, *organizational scope level*, the objects of study are the actors and their relationships in the context of an organization in the SECO. Performance and evolution should be analyzed as aspects that depend on the SECO entrepreneurs. Also, the opening process is the target issue of the organization and involves sharing knowledge, research, market and technology with its partners. At the second level, *SECO scope level*, the objects of study are the software supply networks (SSNs) as well as their relationships that include all stakeholders (i.e., suppliers, customers, distributors, third-part developers) and the internal characteristics related to the SECO health and stability (i.e., size, types, roles, connectedness etc.).

Finally, at the third level, *SECOs scope level*, the objects of study are the SECOs themselves and their relationships. The SECOs life cycles are analyzed through four phases: (1) the establishment of a market relationship with a dominant and focused organization; (2) the emergence of a preliminary network; (3) the reduction of the dominant (and focused) organization's power, and the stimulus of new communities of practice; and (4) the existence of a community of creation, where no dominant organizations exist and the power is distributed. In this context, the SECOs should have well defined frontiers, even overlapped, such as a market, a technology, an infrastructure or an organization, and also geographic restrictions, component specifications, license availability, their age and history.

From this overview, understanding and realizing time and space dimensions of SECOs were pointed out by Jansen *et al.* [15], and explored by Hunink *et al.* [12] and van der Berk et al. [25]. Hunink *et al.* [12] propose a method to create a SECOs domain specific taxonomy in a wide and complete way, allowing software vendors, scientists and government to have insights and identify the gaps between needed and shared information. In turn, van der Berk *et al.* [25] present a model to describe the SECO key characteristics, aiming at evaluating the SECO status and observing how decisions can impact its performance, or generate strategic advantages based on the experience (i.e., past). In both research works, the concern with theoretical basis from Business Schools is frequently mapping (or instancing) concepts to the SE context.

The related works summarized in the last paragraph explore distinct directions: one of them develops a process to create a taxonomy, and the other one creates an evaluation method. Despite these efforts are motivated by the lack of methods, techniques and tools to maximize the awareness in SECOs, no link between a process and a model has been established, aiming at understanding the SECOs life cycles from their birth to maturity (or disappearance, eventually). This is an orthogonal problem to the SECOs and requires exploring information extracted from a set of parameters and behaviors existent in software industry and real cases reported by academy. Trying to explore the mentioned problem, this paper presents an initial proposal for SECOs engineering to outline a set of steps that combines three different dimensions of SECOs and joins different existing perspectives in SECOs research literature through a survey. This paper focuses on the first dimension, that is, architecture. Also, some results of a preliminary analysis of the Software Reuse Lab's SECOs at COPPE/UFRJ are used to exemplify the first dimension of the proposal.

Besides this section related to the introduction and background, the paper is organized in the following: Section 2 presents an overview of the proposal for SECOs engineering, based on steps to contemplate a SECO tridimensional view during its life cycle; Section 3 explores the architectural dimensions; and Section 4 concludes the paper, discussing ways to detail the SECOs engineering approach.

2 The Proposal for Software Ecosystems Engineering

Understanding SECOs from a three-level division discussed in Section 1 [15] requires focusing on the SECO *scope*. Each level has different research challenges starting from the effect of SECO architectural changes to develop general metrics and measure the SECO health. These challenges can be articulated through the definition of general properties of target objects (in organizational, SSN, or SECO scope level) such as health, interaction, performance, inputs, outputs, competition, value sharing and coordination methods. Beyond the scope, different *dimensions* that cross the SECO levels should be considered in order to represent the pillars extracted from literature researches [4] [7] [14][15][16][22]: (i) software; (ii) networks and social business; and (iii) actors, organizations and business ecosystems. In other words, many organizations play with their older and newer SE process models in the market that are mixed with their business models, their involvement with third-part developers and their open product architecture or platform.

These views are also observed from a classification of 15 papers published at the First and Second IWSECO[1] in three categories:

- **SECO architecture**: five papers explore decision-making aspects and architectural properties maintenance, for example, using design and code visualization [1][2][5][9][21];
- **SECO strategies and tactics**: seven papers explore analysis of SECOs perspectives, make analogies with other ecosystems types, and also provide methods and models for organizing, classifying and evaluating SECOs [7][10][12][15][17][22][25];
- **SECO social networks**: three papers explore nets focused on stakeholders (i.e., SECO community management) and artifacts (i.e., knowledge management in requirements, for example), as well as their combinations [8][11][24].

From the SECO challenges discussed at IWSECO 2010, such as "why do SECOs appear and disappear?" and "how to define and monitor SECO scope, types, roles and characteristics", and the current lacking of researches in this direction, this research defines a proposal for SECO engineering, initially focused on researches presented at IWSECO. The goal is to understand SECOs generated by different SSNs throughout their life cycle phases (from their birth to their death or impairment) considering their three levels of scope and allowing the identification of new SECOs. So, the proposal

[1] Available at: <http://iwseco.wordpress.com/>.

is structured in a set of related steps classified according to a SECO tridimensional view, initially distinguished by Campbell & Ahmed [7].

- **Architectural dimension** focused on the SECO platform (i.e., market, technology, infrastructure or organization) through platform domain engineering process (establishing its life cycle), commonalities and variabilities management (defining platforms features), and developed SPL architecture (treating the platform as a SPL);
- **Business dimension** focused on knowledge flow, that is, artifacts, resources and information, through a business (establishing goals and action plans by programs and projects, e.g., MPS.BR program[2]), innovation (linking a SECO to a market, e.g., MPS Model was developed by SOFTEX Brazilian Society to help Brazilian micro, small and median companies to get quality in software processes and products) and strategic planning (understanding how, when, where and who will perform the goals, e.g., the involvement of government, university and industry in developing and maintaining MPS Model) views;
- **Social dimension** focused on SECO stakeholders through balancing proposition and realization of utility (why stakeholders integrate, extend and modify knowledge in a SECO, and interact to each other), promotion (how stakeholders' capabilities and engagement are implicit and explicitly recognized) and knowledge (what collaboration, open source development and other social network opportunities contribute to stakeholders).

The next section discusses the goal and steps of the first dimension. Examples from a preliminary analysis of four Software Reuse Lab's Brazilian free software projects at COPPE/UFRJ are presented to illustrate some concepts. These projects are: (i) *Odyssey*[3]: a large Java SE project, which is a standalone IDE to support domain engineering and software reuse, involving a platform kernel (Odyssey-light) with several plug-ins (subprojects) in evolution to support software process lines, developed since 1997; (ii) *Brechó*[4]: a medium Java EE project, which is a components and services web library to support reuse management processes and value-based component markets and environment based on open source frameworks (Struts and Hibernate), being a self-contained platform in a kernel/plug-ins refactoring phase, developed since 2005; and (iii) *EduSE Portal*[5]: a medium Java EE project, which is a collaborative web environment for empirical research on SE education (systematic reviews, surveys and bodies of knowledge management) based on open source frameworks (JBoss Seam), being a self-contained platform in development phase, developed since 2009; and (iv) *RPP Portal*[6]: a medium Joomla project, which is a collaborative web environment for social sciences research on public politics, being a self-contained platform in development phase, developed since 2010, which will support 10 academic groups that want to share research components as web contents (videos, pictures, interviews, thesis, reports and their combinations).

[2] MPS.BR is a program to improve the Brazilian companies' process model. Details in [20].

[3] Available at: <http://reuse.cos.ufrj.br/odyssey>.

[4] Available at: <http://reuse.cos.ufrj.br/brecho>.

[5] Available at: <http://lab3D.coppe.ufrj.br/portaledues>.

[6] Available at: <http://lab3D.coppe.ufrj.br/rpp>.

3 Dimension #1: Architecture

The first dimension is related to the organizational and SSNs scope levels (internal point of view) more than the SECO scope level (external point of view) since it focuses on the *platform* element. This dimension aims at knowing how SE is applied to the platform conception, development and maintenance, considering three steps:

Step 1: *contextualize platform project and development* – corresponds to the platform analysis phase done via three activities. The involved concepts can be matched to van den Berk *et al.*'s concepts relationship domain model [25]:

Activity 1: *select platform* – represents a decision point in choosing a platform in order to study it, depending on the SECO boundary (i.e., market, technology, infrastructure or organization). In the Software Reuse Lab, Odyssey and Brechó platforms were selected for a preliminary analysis, as distinct SECOs. Despite, some examples mention EduSE Portal and RPP Portal in the following steps.

Activity 2: *identify roles* – aims to define who are the SECO actors, considering the different roles previously pointed out by the business ecosystem literature since a SECO is a specialization of a business ecosystem [13]. These roles are grouped in two categories: *hubs* and *niche players* [15][25]:

- The hubs can be *keystones* (responsible for creating and sharing value with all SECO actors, e.g., bachelors, master and PhD students who work in the Software Reuse Lab's platforms), or *dominators* (responsible for extracting a value as he/she can assimilate or eliminate it from the SECO, e.g., Eclipse SECO is attracting students to develop their researches out of Odyssey or Brechó's platforms). In turn, the niche players use the platform to create value to it, developing and improving its capabilities and differentiation by themselves.
- The niche players can be *influencer* (SECO participant who influences keystone, e.g., Quality and Empirical SE Groups at COPPE/UFRJ, and also users from different Brazilian regions who report issues and make suggestions), *hedger* (SECO participant who wants to belong to the concurrent SECOs to minimize risks, e.g., SE groups from other Brazilian universities such as UFF, PUC-RS and IFF that work with the Software Reuse Group) and *disciple* (SECO beginner who wants to expose opinions about the platforms, e.g., SE Group from UFLA, integrated to EduES Portal and Brechó through a national research project approved in 2010).

Activity 3: *analyze health* – consists in quantifying and qualifying some health indicators related to the SECO state from its platform. The main three indications are [10][15]:

- *productivity*: describes the SECO activity level, i.e., how much business is created, how much value is earned and how many actors are joined. For example, in May 2008, the Brechó SECO has a special mark with the most intense period of development as reported by the

StatSVN[7] tool (Fig. 1). Other interesting measures in this case are: developed use case points, flow of component production, number of reports, quality of thesis and dissertations etc.

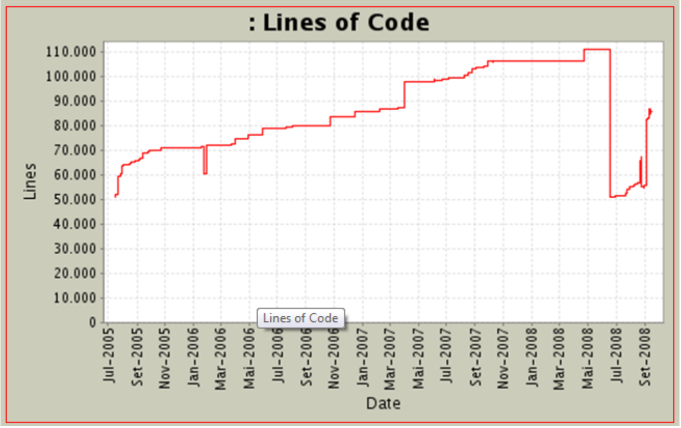


Fig. 1. SVN Commits in LoC (Brechó SECO)

- *robustness*: describes how a SECO can recover from a major stress by itself, such as keystone loss, the "death" of some niche players, or a technological advance that affect the major part (community and/or platform) of the SECO. For example, the effective exit or loss of master and PhD students due to industry opportunities or federal concourses reduce the number of members developing and leading new Odyssey plug-ins or Brechó extensions. Other interesting measures in this case can be: quality of research (publishing impact) and quality of platforms' software product according to international standards such as ISO 25000[8].
- *niche creation*: describes the SECO capability in creating opportunities to new and old actors to explore new business chances. For example, the continuous search for national and regional agencies for financial support (e.g., CNPq, CAPES, FAPERJ in Brazil) to join new research groups (e.g., SE Group at UFLA, and IPPUR – *Regional and Urban Plan Research Institute* – at UFRJ) and strengthen new Software Reuse Lab's SECOs (EduSE Portal and RPP Portal, respectively). Another example is the marketing used to promote Brechó and EduSE Portal SECOs through short courses and papers presentation in national and international events. Other interesting measures in this case are the number of new collaborators and users in the platform.

Step 2: *plan the process of opening the platform architecture* – corresponds to the platform design phase done through three activities:

[7] StatSVN retrieves information from a SVN repository and generates various tables and charts describing the project development. Available at: <http://www.statsvn.org>.

[8] ISO 25000 is a standard for SE – Software product Quality Requirements and Evaluation, at: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35683>

Activity 1: *specify levels* – aims to identify and separate modules or components from the platform, exploring the layer-based architecture benefits and making its opening process easier, because it can be organized in tasks and subtasks workgroups where each actor leads with a particular abstraction level, based on [2]. For example, Brechó Library and EduSE Portal web platforms in their SECOs, as well as Odyssey IDE desktop platform in its SECO, have three layers: (i) *extended applications* developed by external developers (other SE groups in Brazil) and installed by users in their target systems or infrastructure (researchers and professors who use the platforms); (ii) *native applications* developed by the Software Reuse Lab and, in some cases, do not modified; and (iii) *kernel* developed by keystones, which represents the platform hearth and treats low-level components such as device drivers, security, framework etc.

Activity 2: *delineate factors* – consists in defining platform extension and accessibility mechanisms from making the conditions that govern the access to different layers and components explicitly, based on [2]. For example, three actions are used to make clear the notion of architecture opening in Brechó SECO: (i) *integrate*: allows using components from an existing layer in an application via API, service call, code inclusion, shared data objects or other software extension mechanisms, e.g., a project that integrates the tool for supporting software development process Microsoft Team Foundation Server (TFS)[9] to Brechó [26]; (ii) *extend*: allows enhancing the functionalities of components in a layer, e.g., the evolution of Brechó platform from a component repository to a component marketplace environment, generating Brechó-VCM SECO [23]; and (iii) *modify*: allows replacing or modifying components in a layer, e.g., the evolution of a component trade mechanism [26] to support pricing models [22].

Activity 3: *define licenses* – tries to facilitate and restrict the participation of the SECO actors over the platform through rights and obligations that govern the process of opening the architecture [1]. Aspects related to components evolution and replacing, architecture evolution, component license evolution, and modification of wished rights and acceptable obligations should be considered. This happens through a process called platform “co-evolution” because it shows the interdependence among software vendors (keystones) and suppliers (niche players), and the need of communication and coordination mechanisms in this scenario. Besides, the emergence of licenses should consider some kinds of common elements in software architecture in a platform such as source code components, executable components, web services, APIs, software connectors, connection methods, and systems and subsystems configured architectures. For example, Software Reuse Lab always requires allowance to integrate, extend or modify entities in Brechó platform, as shown in Table 1.

Step 3: *balances architecture modularity*[10] *and transparency*[11] *in SECO platform* – corresponds to platform implementation phase done through four activities:

[9] Available at: <http://msdn.microsoft.com/en-us/tfs2008/default.aspx>.

[10] Modularity consists in applying the traditional engineering principle related to decomposing a system in manageable modules, minimizing the technical coupling among its parts [9].

[11] Transparency consists in making all kinds of development information available, including design and code, development tasks, defects and interactions among SECO stakeholders [9].

Table 1. Comparison among architecture opening strategies in the Software Reuse Lab' SECOs, based on [2]. *P* = Possibility, *L* = License status, *Po* = Possible, *Pc* = Possible for some components, *Np* = Not possible, *Pn* = Permission is not needed, *Ps* = in some cases, permission is needed, and *Pa* = Permission is always needed. Specially, Brechó's kernel is being studied in order to derive components (critical), and EduSE Portal architecture is in a good stage to start thinking about this (alert).

	Brechó		***Odyssey***		***EduSE Portal***	
FACTOR	**P**	**L**	**P**	**L**	**P**	**L**
Integrate *extended applications*	Po	Pn	Po	Pn	Po	Ps
Extend *extended applications*	Po	Pn	Po	Pn	Po	Ps
Modify *extended applications*	Po	Ps	Po	Ps	Po	Ps
Integrate *native applications*	Po	Ps	Po	Ps	Po	Ps
Extend *native applications*	Po	Ps	Po	Ps	Po	Ps
Modify *native applications*	Po	Ps	Po	Ps	Po	Ps
Integrate *kernel*	Po	Pa	Po	Pa	Po	Pa
Extend *kernel*	Po	Pa	Po	Pa	Po	Pa
Modify *kernel*	Po	Pa	Po	Pa	Po	Pa

Activity 1: *capture context and establish strategies* – its objective is to detail the SECO platform scope, classifying it according to the *abstraction level* (e.g., reusable asset in requirement, design, code or executable level, resource, information etc.) and *type* (e.g., functionalities, components, crosscutting concerns etc.) of knowledge manipulated in SSN, as well as the *actor profile* (e.g., platform manager, requirement engineer, analyst, internal or external developer etc.). Thus, it is easier to apply the component-based development (CBD) using interfaces, since the architecture is layered. For example, Brechó platform is a web application developed over Java EE platform using MVC pattern and web (Struts) and persistent (Hibernate) frameworks, and use Tomcat container and MySQL database servers to run the system. SVN version control system presents 17 developers since 2005. Finally, the platform mostly deals with code artifacts, based on modules and components and has a manager (keystone player since 2007) and two developers in different geographical regions (niche players since January 2011). From the lack of market-style documentation and the niche players volatility (undergraduate students), a strategy adopted since the beginning of Brechó SECO was the use of *javadoc* and design patterns, as well as known and free technologies, though it is difficult to update and migrate the platform to new ones because sometimes there is no available human/financial resource.

Activity 2: *define information elements* – its objective is to make three platform architectural key elements explicit, the first one related to platform *translucence interfaces* and the last one to *visibility* of how modeling, designing and coding platform components evolve, based on [9]:

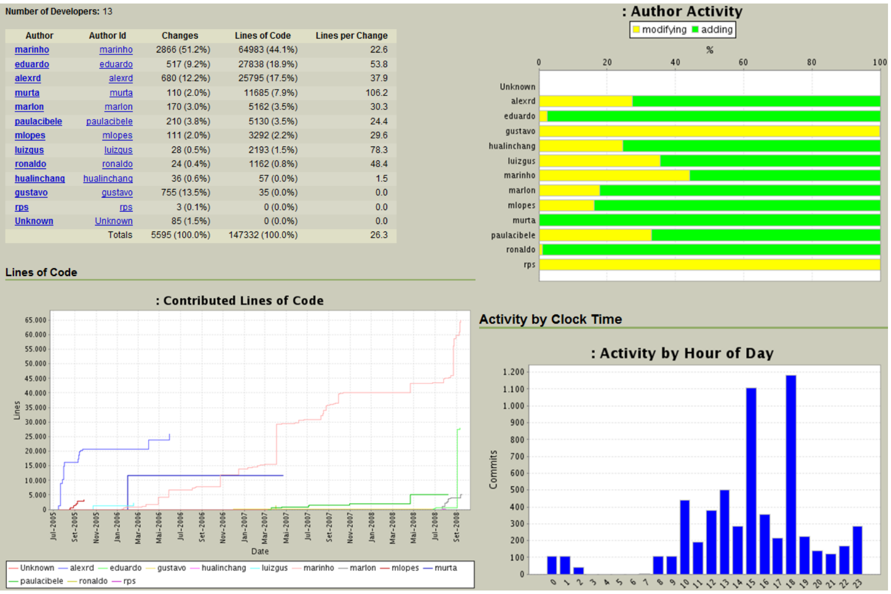
- *uncertainty*: is the probability of changing platform interfaces, requiring decisions in *time* (e.g., evolution, correction etc.) and *space* perspectives (e.g., collaborative development etc.). The goal of treating this information element is the *interfaces stability*, impacting the tracing among different abstraction levels and types of manipulated components. For example, Odyssey SECO has niche players using its platform kernel to work in different geographical regions, and they need to coordinate their activities through discussion lists and annual workshops in order to orchestrate decisions overtime.

- *complexity*: is the property of standard and understandable interfaces compose the platform development process, exploring the information hiding principle in order to benefit from the niche players activities in different abstraction levels of platform components (code, model and requirement). The goal of treating this information element is the *standardization* (use of code and design patterns), impacting product characteristics such as maintainability and reusability (how to calculate cost and effort to evolve the platform). For example, Brechó platform was developed over MVC architectural pattern, using known frameworks and following code patterns established by Sun's Java Code Conventions – this decision directly represents an advantage to propitiate the entrance of new niche players in Brechó SECO.
- *activity awareness*: is the capability of actors to clearly know process activities, dependencies and barriers in two perspectives, *artifacts* and *roles*, based on [24]. The goal of treating this information element is the *coordination and communication* (from CSCW), impacting the platform knowledge comprehensibility (e.g., how to calculate cost and effort to manage the platform). For example, Odyssey, Brechó and EduSE Portal SECOs platforms are submitted to a version control system (SVN) and have a modification control system (Bugzilla) that allows the old and new niche players to communicate and collaborate in developing and maintaining architectural components in each platform, including Yahoo and Google groups.

Activity 3: *calculate and analyze metrics related to information elements* – its objective is to extract platform architecture knowledge from the information elements discussed in the last activity:

- *Measuring uncertainty* requires data collected from niche players' experiences (e.g., architects and developers) when they face requirements comprehension and platform characteristics (e.g., knowledge map), as well as from models that quantify uncertainty points based on historical similar projects in the platform. For example, the time line and the effort (LoC) to develop a new component or extension to show information in bar/pizza graphics can be extracted from SVN data considering the components developed in marketing and evaluation mechanisms at Brechó platform [23].
- *Measuring complexity* requires data collected from components interfaces in different layers (e.g., number and parameters types), from OCL architectural descriptions, and from platform's nonfunctional properties or crosscutting concerns. For example, in Brechó platform, *javadoc* improves the code legibility and maintainability, and that characteristics can be verified through the use of product metrics on component interfaces overtime.
- *Measuring activity awareness* requires data collected from contracts that govern the links between artifacts and roles, and from reports of architectural design tools that capture, document and track the

platform evolution during decision-making processes (e.g., uses of new resource, blocks of code parts, establishment of pre and post conditions etc.). For example, StatSVN was used to collect and analyze Brechó platform data such as source code time line, packages and files per change, developers contribution (commits history), activities per hour, per day or per week etc. This can help the licenses definition. Some information is presented in Fig. 2.



Number of Developers: 13

Author	Author Id	Changes	Lines of Code	Lines per Change
marinho	marinho	2866 (51.2%)	64983 (44.1%)	22.6
eduardo	eduardo	517 (9.2%)	27838 (18.9%)	53.8
alexrd	alexrd	680 (12.2%)	25795 (17.5%)	37.9
murta	murta	110 (2.0%)	11685 (7.9%)	106.2
marlon	marlon	170 (3.0%)	5162 (3.5%)	30.3
paulacibele	paulacibele	210 (3.8%)	5130 (3.5%)	24.4
mlopes	mlopes	111 (2.0%)	3292 (2.2%)	29.6
luizgus	luizgus	28 (0.5%)	2193 (1.5%)	78.3
ronaldo	ronaldo	24 (0.4%)	1162 (0.8%)	48.4
hualinchang	hualinchang	36 (0.6%)	57 (0.0%)	1.5
gustavo	gustavo	755 (13.5%)	35 (0.0%)	0.0
rps	rps	3 (0.1%)	0 (0.0%)	0.0
Unknown	Unknown	85 (1.5%)	0 (0.0%)	0.0
	Totals	5595 (100.0%)	147332 (100.0%)	26.3

Fig. 2. LoC per change and contributions by developers (on the left side), and activity *type* (modification/correction) and *flow* (per hour per day) in Brechó SECO.

Activity 4: *apply translucence to artifacts interfaces in the platform* – its objective is to contribute to the SECO coordination and communication mechanisms, supporting collaboration and cooperation and avoiding information overload (i.e., each stakeholder profile can access an abstract level and a type of platform knowledge according to his/her role). In parallel, the property and safety should be treated in order to maintain the reliability of the manipulated knowledge (e.g., in models and source code). For example, the Brechó platform implementation is based on *javadoc* and is controlled by SVN. This fact allows mining repository data to visualize change impacts in platform components, and filter information (e.g., set of classes) to be shown to a developer based on his/her task requirements and stakeholder profile. Cataldo & Herbsleb [9] point out that a strategy to do this would make information elements explicit through *tags* in architectural descriptions and *javadoc* in source code, contributing to the interfaces translucence, i.e., improving the visibility of information elements or behaviors and hiding others via links among technical and socio-organizational roles. Thus, rules that relate knowledge in different abstract levels and types to stakeholders profiles at SECOs can benefit from information visualization. More details can be found in [9].

4 Conclusion and Future Work

Since the lack of theoretical and applied research in SECOs management through a SE point of view is a challenge to make this field well established in academic research, this paper presented an initial proposal for SECOs engineering to outline a set of steps that combines three different dimensions of SECOs and joins different existing perspectives in the SECOs research literature through a survey. The preliminary contribution was to understand how SECO community is treating ecosystems in a SE point of view, and integrate the works presented at the two first IWSECO editions. In this paper, the focus was on the first dimension, that is, architecture, but the others (business and social) are being studied and linked to the architectural one, since it is impossible to treat SECOs with a pure engineering approach. The discussed proposal will provide a framework to guide and allow deeper researches related to an approach to support SECOs management and development based on empirical studies (primary and secondary ones) which also will involve case studies with well-known SECOs such as Android, Force.com, Eclipse, Microsoft, Linux etc. This can provide a body of knowledge to make SECOs diagnosis, design, and validation and decision-making processes available based on the fact that ecosystems appear, are developed, mature and/or disappear as well as markets, technologies, platforms and organizations, processes, models, techniques etc.

The results of a preliminary analysis of the Software Reuse Lab's Brazilian SECOs at COPPE/UFRJ point out that several concepts presented at IWSECO 2009 and IWSECO 2010 can be connected in a broader SE approach, as discussed in Section 3. Thus, it can be realized that understanding SECOs requires joining a lot of instable IT elements in an entity (platform), adding SE elements which alter those elements during the ecosystems creation, development and maintenance – a SE challenge of treating the social and economic aspects [3]. Future work consists of expanding the proposal with other case studies and with expert-based surveys to calibrate the architectural dimension and integrate it to the other two dimensions in a unified approach. Additionally, extensions in the Brechó Library to support component-based architecture SECOs management and development will be done, since this tool has a lot of mechanisms that can help business and social dimensions.

Acknowledge. We thank to CNPq/FAPERJ for their financial support to the research.

References

1. Alspaugh, T.A., Asuncion, H.U., and Scacchi, W. 2010. The Role of Software Licenses in Open Architecture Ecosystems. In: *Proceedings of the 1st IWSECO*, *11th ICSR*, Falls Church, USA, 4-18, September.
2. Anvaari, M., and Jansen, S. 2010. Evaluating Architectural Openness in Mobile Software Platforms. In: *Proc. of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 85-92, Aug.
3. Boehm, B. 2006. A View of 20th and 21st Century Software Engineering. In: *Proceedings of the 28th ICSE*, Shanghai, China, 12-29, May.
4. Bosch, J. 2009. From Software Product Lines to Software Ecosystem. In: *Proceedings of 13th SPLC*, San Francisco, CA, USA, 1-10, August.
5. Bosch, J. 2010. Architecture Challenges for Software Ecosystems. In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 93-95, August.

6. Boucharas, V., Jansen, S., and Brinkkemper, S. 2009. Formalizing Software Ecosystem Modeling. In *Proc. of the 1st IWSECO*, 11th ICSR, Falls Church, USA, 34-48, Sep.
7. Campbell, P.R.J., and Ahmed, F. 2010. A Three-dimensional View of Software Ecosystems. In: *Proc. of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 81-84, Aug.
8. Capuruço, R.A.C., and Capretz, L.F. 2010. Integrating Recommender Information in Social Ecosystems Decisions. In: *Proc. of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 143-150, August.
9. Cataldo, M., and Herbsleb, J.D. 2010. Architecting in Software Ecosystems: Interface Translucence as an Enabler for Scalable Collaboration. In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 65-72, August.
10. Dhungana, D., Groher, I., Schludermann, E., and Biffl, S. 2010. Software Ecosystems vs. Natural Ecosystems: Learning from the Ingenious Mind of Nature. In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 96-102, August.
11. Fricker, S. 2009. Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems. In: *Proc. of the 1st IWSECO*, *11th ICSR*, Falls Church, USA, 19-33.
12. Hunink, I., van Erk, R., Jansen, S., and Brinkkemper, S. 2010. Industry Taxonomy Engineering: The Case of the European Software Ecosystem. In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 111-118, August.
13. Iansiti, M., and Levien, R. 2004. *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation and Sustainability*. Harvard Business Scholl Press.
14. Jansen, S., Finkelstein, A., and Brinkkemper, S. 2009. A Sense of Community: A Research Agenda for Software Ecosystems. In: *Proc. of the 31st ICSE*, *New and Emerging Research Track*, Vancouver, BC, Canada, 187-190, May.
15. Jansen, S., Brinkkemper, S., and Finkelstein, A. 2009. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. In: *Proceedings of the 1st IWSECO*, *11th ICSR*, Falls Church, USA, 34-48, September.
16. Kittlaus, H.B., and Clough, P.N. 2009. *Software Product Management and Pricing: Key Success Factors for Software Organizations*. Springer Publishing Company.
17. McGregor, J.D. 2010. A Method for Analyzing Software Product Line Ecosystems". In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 73-80, August.
18. Messerschmitt, D.G., and Szyperski, C. 2003. *Software Ecosystem: Understanding an Indispensable Technology and Industry*. The MIT Press.
19. Moore, J.F. 1996. *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*. Harper Business.
20. Montoni, M.A., Rocha, A.R., and Weber, K.C. 2009. MPS.BR: A Successful Program for Software Process Improvement in Brazil. *Sof. Process: Impr. and Practice* 14(5):289-300.
21. Pettersson, O., Svensson, M., Gil, D., Andersson, J., and Milrad, M. 2010. On the Role of Software Process Modeling in Software Ecosystem Design. In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 103-110, August.
22. Santos, R.P., and Werner, C.M.L. 2010. Revisiting the Concept of Components in Software Engineering from a Software Ecosystem Perspective. In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 135-142, August.
23. Santos, R., Werner, C., and Silva, M. 2010. Brechó-VCM: A Value-Based Approach for Component Markets. *Intern. Trans. on Systems Science and Applications* 6(2/3):179-199.
24. Seichter, D., Dhungana, D., Pleuss, A., and Hauptmann, B. 2010. Knowledge Management in Software Ecosystems: Software Artefacts as First-class Citizens. In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 119-126, August.
25. van den Berk, I., Jansen, S., and Luinenburg, L. 2010. Software Ecosystems: A Software Ecosystem Strategy Assessment Model. In: *Proceedings of the 4th ECSA*, *2nd IWSECO*, Copenhagen, Denmark, 127-134, August.
26. Werner, C., Murta, L., Marinho, A., Santos, R., and Silva, M. 2009. Towards a Component and Service Marketplace with Brechó Library. In *Proceedings of the IADIS International Conf. WWW/Internet*, Rome, Italy, 567-574, November.