Modelling and Applying OSS Adoption Strategies

Lidia López[1], Dolors Costal[1], Claudia P. Ayala[1], Xavier Franch[1], Ruediger Glott[2], Kirsten Haaland[2]

[1] Universitat Politècnica de Catalunya, Barcelona, Spain
[2] UNU-MERIT, Maastricht, The Netherlands

Abstract. Increasing adoption of Open Source Software (OSS) in information system engineering has led to the emergence of different OSS business strategies that affect and shape organizations' business models. In this context, organizational modeling needs to reconcile efficiently OSS adoption strategies with business strategies and models. In this paper, we propose to embed all the knowledge about each OSS adoption strategy into an *i** model that can be used in the intentional modeling of the organization. These models describe the consequences of adopting one such strategy or another: which are the business goals that are supported, which are the resources that emerge, etc. To this aim, we first enumerate the main existing OSS adoption strategies, next we formulate an ontology that comprises the activities and resources that characterise these strategies, then based on the experience of 5 industrial partners of the RISCOSS EU-funded project, we explore how these elements are managed in each strategy and formulate the corresponding model using the *i** framework.

Keywords: OSS; Open Source Software; OSS adoption strategy; OSS ontology; *i** framework; i-star.

1   Introduction

The key purpose of any business is to create value and to achieve revenues. The business model of an organization holistically captures the ways and means how these goals can be achieved. Therefore, there is no organization without a business model, regardless of whether or not a company explicitly describes it [1][2].

A business strategy describes the approach of a business to successfully compete with other businesses in a given market. A business model can be seen as the translation of a company's business strategy into a blueprint of the company's logic of earning money [3]. Business strategies are dependent on many factors, and information technology (IT) approaches are one of them. At this respect, Open Source Software (OSS) has become a driver for business in various sectors, namely the primary and secondary IT sector. Estimates exist that in 2016, a 95% of all commercial software packages will include OSS components [4]. OSS adoption impacts in fact far beyond technology, because it requires a change in the organizational culture and reshaping IT decision-makers mindset. The way in which OSS affects and shapes business mod-

els is becoming object of increasing attention, and as a result, several OSS business strategies have been identified so far [5][6][7].

Leveraging business strategies with the organization business model is a challenging task per se, and it implies reconciling them from very different perspectives [3]. Organizational modelling can provide a way to define the organization's goals and to serve as the context in which processes operate and business is done. In this context, in order to support organizations that would like to adopt OSS (hereafter OSS adopter) and analyze the implications of such adoption; we describe seven different OSS adoption strategies in terms of models that relate business goals and resources. These models can be used as a reference for understanding and assessing the impact of the OSS adoption strategies on the OSS adopter organization; as well as complementing the OSS adopter organizational model.

The remainder of the paper is organized as follows. Section 2 introduces the basic concepts needed in the paper. Section 3 presents the research method followed. Sections 4 to 6 develop the main contributions of the paper: the OSS ontology used, the arrangement of its elements into models for the OSS adoption strategies and the application of such models. Last, Section 7 provides conclusions and future work.

2 Background: OSS Adoption Strategies

OSS can play a role at any place of the business model of an organization. The most usual roles are: it can be received as a supply from an OSS community or another organization; it can be produced in-house; it can be part of the organization's value proposition (e.g. by lowering costs or by improving compatibility); it can be used as infrastructure for the development of software or for the execution of business processes; it can be sold in order to allow for revenues (e.g. based on a dual license); it can determine customer segments (e.g. other organizations that produce OSS); or it can be used in order to lower costs (e.g. by using a license free operating system for IT systems that are sold to customers).

Depending on from where OSS is received or how it is produced, and where in the business model and for which purposes OSS is used we may distinguish several types of OSS adoption strategies, which are not necessarily mutually exclusive. In the context of the FP7 RISCOSS project (www.riscoss.eu), we have identified the strategies below, and are the ones considered in the rest of the paper:

- *OSS Integration* means the integration of an organization in an OSS community with the purpose to share and co-create OSS. In this case, being part of the community in order to benefit from the commonly created OSS components is the key goal of the OSS strategy; it is not necessary for the adopter organization to play a leading role within the community.
- *OSS Initiative* means to initiate an OSS project and to establish a community around it. Usually, the key goal of this strategy is to create community support, but in contrast to the OSS Integration strategy, the adopter establishes the community as a resource that directly serves the company's business strategy and model. As a consequence, exercising control over the OSS community is typical for this strategy.

- *OSS Takeover* means to take over an existing OSS project/community and to control it. The main difference from the OSS Initiative strategy is that the OSS community already exists.
- *OSS Fork* means to create an own independent version of the software that is available from an existing OSS project or community. This strategy is usually followed when an OSS community on which the adopter organization depends develops in directions that contradict or hamper the organization's business goals. Exercising control over the forked community is typical but not necessary, as the fork community should consist of developers that share the adopter organization's view on how the community and the software should evolve.
- *OSS Acquisition* means to use existing OSS code without contributing to the underlying OSS project/community.
- *OSS Release* implies that the organization releases own software as OSS but does not care whether an OSS community takes it up or forms around it. This strategy can, for instance, be observed in the public sector, when software owned by public bodies is released under an OSS license and made available to other public bodies via a repository.

3   Research Approach

This research is performed in the context of the European FP7 RISCOSS project, which aims to support the OSS adopter organizations to understand, manage and mitigate the risks associated to OSS adoption. The consortium includes 5 industrial partners from public and private sectors, with diverse OSS adoption contexts, which have served to formulate the results presented here (as described below). In line with this objective, this paper focuses on supporting organizations in analyzing the implications of adopting a particular OSS business strategy. With this aim, our research approach was based on 3 complementary stages corresponding to 3 research questions:

RQ1: Which activities and resources characterize OSS projects?

This is aimed to understand the relevant activities and resources taking place in the context of OSS projects, especially related to software development and community management. To do so, we conducted a Systematic Literature Review (SLR) using the well-known guidelines defined in [8] in order to identify existing ontologies on the field. We analysed them with respect to our objectives and complemented the results with knowledge coming from RISCOSS industrial partners. The ontology is described in Section 4.

RQ2: How do OSS activities and resources map to OSS business strategies?

In this question, we wanted to inquiry about how the activities and resources emerging from RQ1 map into the different OSS business strategies enumerated in Section 2.1. Our aim was mainly to emphasize and represent the different effects that each OSS business strategy has over the OSS adopter organization. As a result, the

detailed definition of the different models, one for strategy, was obtained. See Section 5 for details.

RQ3: How OSS strategies relate to organizational goals?

OSS business strategies' models resulting from RQ2 were mostly focused, as mentioned in the context of RQ1, on software development and community management activities and resources. However, in order to understand the impact of the activities and resources enclosed in each strategy model, we needed to understand their relationship to the OSS adopter's organizational goals. Therefore, we held some off-line workshops with the five RISCOSS industrial partners, and ended up with a set of related goals that were integrated into the models. Furthermore, we devised a matching process to integrate the strategies' models with the OSS adopter organization. Thus, the use of the OSS business strategies models provides an efficient way to complete the OSS adopter organizational model.

It is important to highlight that these three RQs and their corresponding stages have a formative character as they are aimed to conceive the OSS business strategies models. In all the stages, the industrial partners of the RISCOSS project have been involved to shape and endorse our approach in their respective contexts. In addition, a proof of concept software prototype has been built in order to operationalize the resulting approach and help as a tool for the summative evaluation of the approach in the context of other industrial organizations besides the RISCOSS partners, which would become the last stage of our method (see Figure 1).
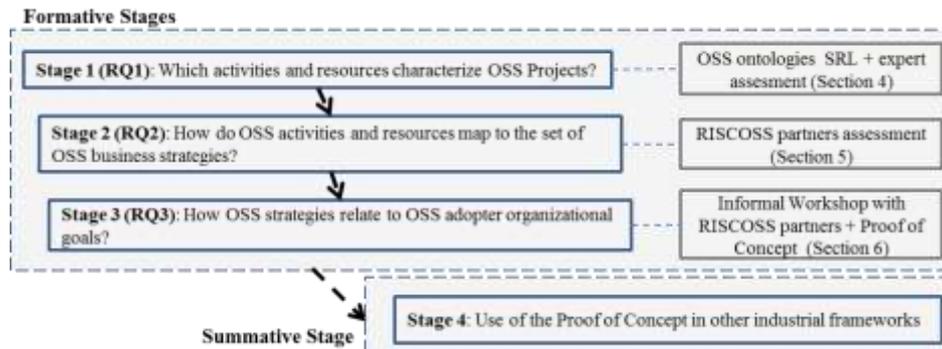
Figure 1: Research method followed in the paper.

4   An Ontology for OSS Adoption Strategies

In order to state the basis for defining OSS adoption strategies models, we have developed an ontology that embraces terms related to OSS projects and the business strategies involved in them. To do so, as a first step, we conducted a Systematic Literature Review (SLR) using the guidelines defined in [8] with the purpose of identifying existing OSS ontologies. The details of the SLR and the subsequent analysis can be found in [9].

Although the SLR was conducted in order to find ontologies related to OSS field, the search string included some terms besides the term ontology (metamodel, glossary, taxonomy and reference model) in order to avoid missing papers with a kind of implicit ontology. This also resulted in a large set of initial papers that was considerably pruned: from an initial set of 1214 primary studies, we selected 9 papers to be analysed as papers that contain an ontology (explicit or not), as a result of this analysis, we ended up with 4 of them that stated 3 relevant ontologies: Dhruv's ontology [10], OSDO [11][12], and OFLOSSC [13].

We finally chose OFLOSSC as departing point because: (1) it was the most complete amongst the ontologies reviewed (it actually includes parts of the other 2 OSS ontologies also covered in the SLR), and (2) it is an ontology for supporting OSS development communities and covers concepts related to community interactions for developing software. However, it lacks of adoption concepts related to OSS adopter organizations, therefore, we have extended the ontology with these missing (with respect to our purposes) concepts. To do so, we performed a thorough analysis of the activities and resources that an adopter organization should consider when participating in an OSS project by running off-line workshops (i.e., discussion that were centralized in a wiki tool) with partners of the RISCOSS project.

The results from such analysis in terms of the identified activities and resources are listed in Table 1. For each element, the table includes its identifier and a brief description. The elements identified have been classified into five groups: software development activities, community-oriented activities, communication activities, personnel activities and resources, as can be seen in Table 1.

The ontology concepts exhibit some relationships that are themselves part of the ontology. Some of these relationships can be expressed with plain logic, e.g. the responsible of developing an OSS component (Act-DEP) is also responsible of testing it (Act-TEST-Comp). Others may require introducing additional concepts, for instance the activity of reporting a bug (Act-RepBUG) *produces* a bug report resource (Res-BUG) and not other types of resources (such as patch, roadmap, etc.).

Table 1: Activities and resources for OSS business strategies

Identifier	Description
Software Development Activities	
Act-SEL	Selection of an OSS component for its deployment or integration in an organization
Act-DEP	Deployment of an OSS component for its actual use in the organization
Act-DEV	Development of an OSS component (specification, design, code)
Act-INT	Integration of an OSS component into another software artifact
Act-TEST-Comp	Testing of an OSS component
Act-TEST-Prod	Testing of a software artifact that integrates an OSS component
Act-MAINT-Comp	Maintenance of an OSS component
Act-MAINT-Prod	Maintenance of a software artifact that integrates an OSS component
Act-PATCH	Development of a patch to correct some bug or add some new feature for an OSS component
Community-Oriented Activities	
Act-NewCOMM	Creation of an OSS community
Act-DECIDE-	Decision of the roadmap of an OSS component. It includes planning of

Roadmap	releases and which features are included
Act-DECIDE-Acc	Acceptance of a contributor in an OSS community
Act-DECIDE-Wishlist	Decision of the desired features for the next releases of an OSS component (but without a concrete planning)
Act-RELEASE	Making available a software component under OSS license (either first time or an evolution)
Communication Activities	
Act-RepPATCH	Communication of a patch for an OSS component
Act-RepBUG	Report of a bug
Act-SUPP	Any kind of support given to the OSS community (except bug reports and patches; e.g. organising or endorsing sponsoring events)
Personnel Activities	
Act-ACQ-Tech	Acquisition of the necessary knowledge about an OSS component to be able to master its technology
Act-ACQ-Man	Acquisition of the necessary knowledge about managing an OSS community
Act-LEARN	Acquisition of the necessary knowledge about an OSS component to be able to operate it (as end user)
Resources	
Res-OSS-Comp	An OSS component as a software artifact
Res-Tech-DOCUM	Technical documentation of an OSS component
Res-User-DOCUM	User documentation (e.g., tutorials) of an OSS component
Res-PATCH	Patch provided for an OSS component
Res-BUG	Report of a bug or post, etc., referred to an OSS component
Res-NEWFEATURE	Report of desired feature(s) for an OSS component
Res-ROADMAP	Strategy for new features and releases of an OSS component

5 Building OSS Adoption Strategy Models

We present next a catalogue of OSS adoption strategy models for each of the strategies described in Section 2. These models are built on top of the ontology presented in Section 4 and each of them combines the ontology elements as required by its corresponding strategy. Models focus on the adopter organization and refer to the particular OSS component under adoption. We have used the *i** framework [14] as modeling approach. *i** is an intentional actor-oriented modelling and analysis framework, which supports representing and analyzing synergistic and conflicting stakeholder interests and decision-making within and across organizational settings. In *i**, we find: the Strategic Dependency (SD) model, declaring the actors and their dependencies; and the Strategic Rationale (SR) model, declaring the goals and intentions of the actors.

The main actors involved in the OSS adoption strategies are: (1) the organization that adopts the OSS component and (2) the OSS community that produces it. The activities performed and the resources produced by each of these actors vary significantly depending on the business strategy, and this is the basis of the model construction: first, for each adoption strategy, we have allocated the activities and resources, presented in Section 4, to the two actors, depending on which one is responsible. Table 2 provides the allocation of activities and resources to the adopter organization actor depending on the adoption strategy followed. The expertise obtained from the

RISCOSS partners has provided us the rationale to choose the most adequate allocation according to the main features of each identified OSS business strategy.

Table 2: Adopter Activities and resources (rows) per OSS strategies (columns)

	Integration	Initiative	Take-over	Fork	Acquisition	Release
Software Development Activities						
Act-SEL	X		X	X	X	
Act-DEP	X	X	X	X	X	X
Act-DEV		X				X
Act-INT	X	X	X	X	X	X
Act-TEST-Comp						
Act-TEST-Prod	X	X	X	X	X	
Act-MAINT-Comp						
Act-MAINT-Prod	X	X	X	X	X	X
Act-PATCH	X	X	X	X		
Community-oriented Activities						
Act-NewCOMM		X		X		
Act-DECIDE-Roadmap		X	X			
Act-DECIDE-Acc		X	X			
Act-DECIDE-Wishlist	X			X		
Act-RELEASE		X				X
Communication Activities						
Act-RepPATCH	X	X	X	X		
Act-RepBUG	X	X	X	X		
Act-SUPP	X	X	X	X		
Personnel Activities						
Act-ACQ-Tech	X	X	X	X	X	X
Act-ACQ-Man	X	X	X	X		
Act-LEARN	X		X	X	X	
Resources						
Res-OSS-Comp		X				X
Res-Tech-DOCUM		X				X
Res-User-DOCUM		X				X
Res-PATCH	X	X	X	X		
Res-BUG	X	X	X	X		
Res-NEWFEATURE	X			X		
Res-ROADMAP		X	X			

Some general observations on Table 2 are the following:

1. There are two strategies that do not require community involvement or contribution of the organization: acquisition and release. Therefore, the activities allocated to these strategies are mainly internal-oriented software development activities and not community-oriented or communication activities (except for *Act-RELEASE* in the release strategy case).

2. For the rest of strategies: the organization participates in communication activities (e.g. *Act-RepBUG*) and contributes with their corresponding resources (e.g. *Res-BUG*). Additionally, the organization develops different community-oriented activities depending, mainly, on whether it is exercising control over the community or not. Remarkably, in the initiative and takeover case, the organization decides the community roadmap (*Act-DECIDE-Roadmap*, *Res-ROADMAP*).
3. Two of the strategies, namely, initiative and fork, require that the organization sets up an OSS community (*Act-NewCOMM*).

The activities of maintaining and testing the OSS component (*Act-MAINT-Comp* and *Act-TEST-Comp*) are not allocated to the organization in any strategy meaning that they are basically developed by the OSS community. Taking as a basis the allocation described in Table 2, we have built an i* model for each strategy. These models have been complemented with two kind of goals: some goals used to structure the model (e.g. *Technical Quality* to embrace the tasks related to acquire skills for using the component Act-ACQ-Tech and Act-Learn) and some high-level goals and softgoals more related to the business strategy goals (e.g. *Take benefit from OSS community*) .
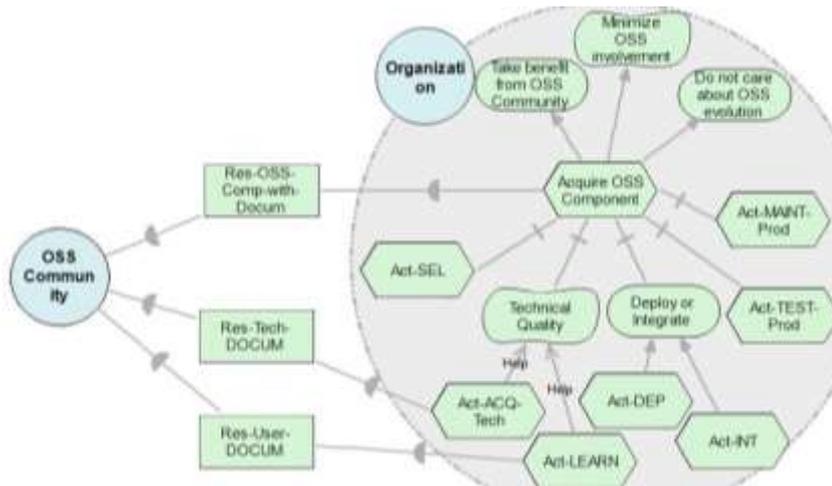
Figure 2: OSS Acquisition model.

Figure 2 and Figure 3 show the models corresponding to the Acquisition and Takeover strategies, respectively. They represent in detail the adoption strategy SR model that will be incorporated into the model of the adopter organization, and the dependencies that exist between the organization and the OSS community. The organization SR model permits to understand the relation of the allocated activities with the goals and softgoals they attain or they require. The dependencies permit to understand the vulnerabilities that the organization has with respect to the OSS community when adopting the strategy.

The model for the OSS acquisition strategy (see Figure 2) shows how the adopting organization only needs the component from the OSS community and does not give

any return to it, therefore only outgoing dependencies stem from the organization actor. Meanwhile the OSS takeover model (see Figure 3) shows dependencies in both directions, because the OSS community receives some payback from the adopter (bug reports, patches…). Goals and activities such as *Community Managed*, *Act-DECIDE-Roadmap* and *Act-DECIDE-Acc* appear inside the organization actor in the takeover model because the organization controls the OSS community in this case. Also, some goals and tasks appear in order to contribute the OSS community (*OSS Community Contributed* and the tasks and softgoals that decompose it).

Taking as a central element the task representing the type of OSS strategy (*Acquire OSS Component* and *Takeover OSS Component*), the models include a set of high-level *business-related goals* directly attained by the strategy (goals above the task, e.g *Minimize OSS involvement* in Figure 2) and the low-level goals or tasks which are *requirements* for an adequate application of the OSS business strategy (below the task, e.g. *Act-SEL* and *Manage OSS Community* in Figure 3).
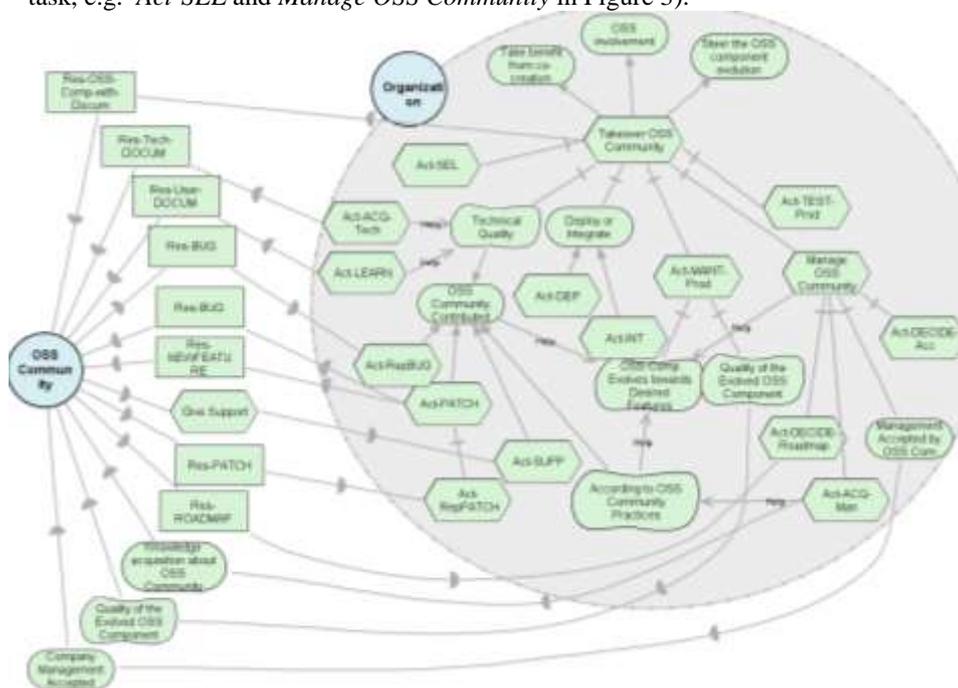
Figure 3: OSS Takeover model.

Models focus on the portion of the organization SR model that is influenced by its corresponding OSS adoption strategy and does not include the general set of (higher-level) business goals pursued by the organization. Nevertheless, the intentional elements shown in the models are certainly related to the more general business goals of the organization since they may contribute to their attainment. Section 6 provides the details about this issue.

6 Applying OSS Adoption Strategy Models in an Organization

OSS adoption strategy models have been developed as general models; therefore, a question to be answered is how to apply them in a specific situation (in our case, an OSS adopter organization). The question is twofold: first, when to apply an OSS adoption strategy model; and second, how to couple the organizational model with the strategy model. To answer the first question, we start by assuming the existence of an organizational model that declares the higher-level goals pursued by the organization and using this existing organizational model we select the most suitable OSS adoption strategy to the organization needs. To answer the second question, we describe the process of instantiation of the organizational model, i.e. extending the model with the elements from the selected OSS adoption strategy model and making the necessary retouches to these new elements.

6.1 Selecting the OSS adoption strategy

Figure 4 shows an excerpt of the organizational model for company *ACME*, that produces the Road Runner Locator (*RR Locator*) product for its customers. This is a company interested in reducing in-house development costs, therefore they decided to reduce the development effort integrating an OSS component as part of its software, but they are not interested in being involved with the OSS community behind the OSS component.
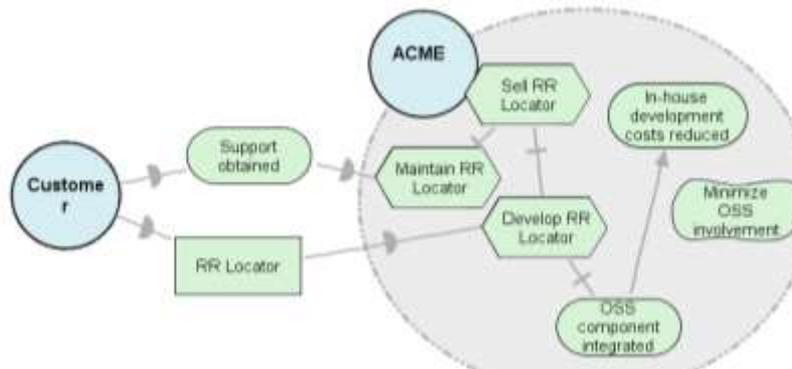
Figure 4. Organizational model

In order to facilitate the process to find the more suitable OSS adoption strategy, we adopt the model matching approach presented in [15]. In this paper, *i** models are used to describe market segments and software packages in order to evaluate the matching between both models with an *i** organizational model, in order to select the best software package for the organization needs. In this paper, we may use the same idea and then evaluate the matching between an organizational model and the set of OSS business strategies models in order to identify the strategies that better match the organizational goals and eliminate those that clearly do not apply to the specific organization. [15] uses a concept called coverage for classifying the matching results, that is adapted in this paper as follows:

- Coverage of the Organizational model: a matching is org-complete if every intentional element from the Organization actor in the organizational SR model is matched at least with one intentional element in the OSS business strategy SR model, otherwise it is org-incomplete. In other words, in an org-complete matching, an adoption strategy supports all the business goals of the organization.

- Coverage of the OSS adoption strategy model requirements: a matching is str-complete if every intentional element, representing *requirements*, from the Organization actor in the OSS adoption strategy SR model is matched at least with one intentional element in the Organization actor in the organizational SR model, otherwise it is str-incomplete. In other words, in a str-complete matching, an adoption strategy can effectively be adopted by an organization since this organization fulfills all the needs ofthe strategy.

The aim of the matching is finding the OSS adoption strategies that cover as much as possible the organization business goals, taking into account that the company has or is willing to have the resources required by the strategy. Therefore, any combination where the coverage of the OSS adoption strategy model is incomplete due to some OSS adoption strategy requirements are missing, excludes the strategy from being used by the organization. Before discarding the strategy, the organizational model can be extended in order to have a str-complete requirements coverage.

As an example, Table 3 shows the matching between the ACME organizational model (Figure 4) and the OSS Acquisition strategy model (Figure 2) for the organizational model coverage. The coverage of the ACME organizational model by the strategy is almost complete, since the only element not covered is the task *Sell RR locator* which, nevertheless, represents a kind of activity not addressed by the OSS adoption strategies. Therefore, the Acquisition OSS adoption strategy is a good choice to be applied to the organization.

Table 3. Matching for organizational model coverage

OSS Organizational model	Matching with OSS business strategy model
Sell RR Locator	Missing
Maintain RR Locator	*Act-Maintain-Prod*
Develop RR Locator	*Act-INT* and *Act-TEST-Prod* are subtasks of task *Develop RR Locator*
OSS component integrated	*Acquire OSS component* is the means to achieve this goal
In-house development cost reduced	*Take benefit from OSS Community* is the means to achieve this goal
Minimize OSS involvement	The same as *Minimize OSS involvement*

Table 4 shows the matching for the OSS adoption strategy coverage. In this case, the coverage is incomplete. The requirements related to achieve the technical skills in order to use the component (*Technical Quality*, *Act-ACQ-Tech* and *Act-LEARN*) are missing, but in this case if an organization wants to use an external component (OSS or not) has to be willing to acquire the necessary knowledge to use it. Therefore, the organizational model should be extended including them in order to have a complete OSS strategy requirements coverage.

Table 4. Matching for OSS acquisition strategy model coverage

OSS Acquisition strategy	Matching with Organizational Model
Requirements	
Acquire OSS Component	Means to achieve the goal *OSS component integrated*
Act-SEL	Considered part of the tasks defining the process related to the goal *OSS component integrated*
Technical Quality	Missing
Act-ACQ-Tech	Missing
Act-LEARN	Missing
Use or Deploy	This is an intermediate node, it does not need matching
Act-DEP	The component has to be "integrated", so this element has to deleted from the OSS business strategy model
Act-INT	This is the element kept in the OSS business model as a means to achieve the goal *OSS component integrated*
Act-TEST-Prod	Considered a subtask of *Develop RR Locator*
Act-MAINT-Prod	The same task as *Maintain RR Locator*

6.2 Instantiating the organizational model

Once the organization has decided the OSS adoption strategy is going to adopt, the instantiation process consists on the following steps:

Figure 5. Organizational model adhering acquisition OSS business strategy

1. Applying the matching, including the intentional elements from the selected OSS adoption strategy model into the organizational model. Only the matched elements are included (Table 3 and Table 4).
2. Making the needed retouches to the resulting model in order to adapt the general OSS adoption strategy model to the specific case.

In the case that we are using as example in this paper, the organization *ACME* is acquiring the component *OSS GIS* to be integrated in its product *RR Locator*, in order to know where the RR is as shown in Figure 5. The new elements that come from the OSS acquisition strategy model (Figure 2) are shown in italics. Task *Acquire OSS Component* and its decomposition is included as the means to achieve the goal *OSS component integrated*, except for the task *Act-DEP*, it does not appear in the model because the organization uses the OSS component integrating (*Act-INT*) it in its own software (*RR Locator*). The task *Maintain RR Locator* replaces the *Act-MAINT-Prod* and the task *Act-TEST-Prod* is renamed as *Test RR Locator*. *ACME* organization adheres to the business-related goal *Minimize the OSS involvement*, but does not agreewith *Do not care about OSS component evolution*.

7   Conclusions and future work

In this paper, we have proposed the use of generic models as a way to model OSS adoption strategies. The main contributions of our work are:

- *An ontology for OSS adoption*. We have defined an ontology for activities and resources implied in OSS adoption strategies. As result of the SLR we undertook, we observed that there are several ontologies for OSS but they focus on the perspective of the community, with special attention to: roles of developers (committers, contributors, …), licenses, etc. In the context of business strategies for organizations, this perspective is not the right one, since the needs are different. Therefore, our ontology goes beyond the state of the art and possibilitates its use in other works that may be interested in the adopting organization perspective.
- *Characterisation of OSS adoption strategies*. For each identified OSS adoption strategy, we have assigned these activities and resources to the actor that is in charge: the adopting organization or the OSS community. As a result, we have provided a characterisation of each strategy in terms of activities undertaken, resources provided, and dependencies of these two actors on each other. Again this is a result that goes beyond the goal of our paper and maybe of interest for researchers that want an overall comparison of OSS adoption strategies.
- *Set of OSS adoption strategy models*. As ultimate result of the paper, we have designed a set of models for the different strategies expressed in the *i** language. The proposal relies upon expert assessment, is operational (has tool support, not presented here) and from a methodological point of view, it is integrated with the organizational model that can be expressed also in *i**. The use of these models provides an efficient way to build organizational models in those organizations that adopt OSS solutions.

Future work goes along several directions. First, we need to work further in the link among business models and OSS adoption strategies, so that the process that has

been depicted in Section 6 becomes more prescriptive. Second, concerning the models for the patterns, we want to use *i** roles as a way to organize the ontology elements: roles like Contributor, Governance Body, OSS User, etc., may arrange the different activities and resources, and then the adoption strategies will simply put together the indicated roles in each case. Third, we need to be able to combine OSS adoption strategy models, either because more than one strategy applies at the same time for the same OSS component, or because more than one OSS component is being integrated in a project. Last, a validation plan based in case studies in the context of our RISCOSS project needs to be executed.

References

[1] Chesbrough, H.: Open Business Models: How to thrive in the new Innovation Landscape. Harvard Business School Press, 2006.

[2] Teece, D. J.: Business Models, Business Strategy and Innovation. In Long Range Planning Journal, Vol. 43(2-3), 2010.

[3] Osterwalder, A. (2004). The business model ontology- a proposition in a design science approach.Lausanne, Switzerland: University of Lausanne. PhD Dissertation.

[4] Driver, M.: Hype Cycle for Open-Source Software. Technical Report, Gartner, 2013.

[5] Kudorfer, F., Laisne, J.P., Lauriere, S., Lichtenthaler, J, Lopez, G., Pezuela, C.: State of the art concerning business models for systems comprising open source software. 2007.

[6] Daffara, C.: Business models in FLOSS-based companies. 2008.

[7] Lakka, S., Stamati, T., Michalakelis, C., Martakos, D.: The Ontology of the OSS Business Model: An Exploratory Study. In Open Source Software and Processes J., Vol. 3(1), 2011.

[8] Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in Software Engineering v2.3. EBSE Technical Report EBSE-2007-01.

[9] Ayala, C., López, L.: D1.1 Modeling Support (initial version). Technical Report, RISCOSS FP7 project.

[10] Ankolekar, A.; Sycara, K.; Herbsleb, J.; Kraut, R.; Welty, C.: Supporting online problem-solving communities with the semantic web. In *Proceedings of* WWW 2006, pp. 575-584.

[11] Simmons, G.; Dillon, T.: Towards an Ontology for Open Source Software Development. In Open Source Systems IFIP International Federation for Information Processing, Vol. 203, 2006, pp 65-75.

[12] Dillon,T.S.; Simmonsg,G.: Semantic web support for open-source software development. In Proceeding of SITIS 2008. pp. 606-613.

[13] Mirbel, I.: OFLOSSC, an ontology for supporting open source development communities. In Proceedings of ICEIS 2009, Volume SAIC, pp. 47-52.

[14] Yu, E.: Modelling Strategic Relationships for Process Reengineering. PhD. thesis," Toronto, 1995.

[15] Franch, X.: On the Lightweight Use of Goal-Oriented Models for Software Package Selection. In Proceeding of CAiSE 2005. LNCS 3520, pp. 551-566, 2005