FROM INTERORGANIZATIONAL MODELLING TO ESSENTIAL USE CASES

Luciana C. Ballejos and Jorge M. Montagna
CIDISI – INGAR Instituto de Desarrollo y Diseño (UTN-FRSF) - Avellaneda 3657 (3000) Santa Fe, Argentina

Keywords: Interorganizational modelling, i* framework, essential use cases, interorganizational information systems.

Abstract: Requirements Engineering (RE) must initially focus on information system (IS) domain modeling rather than on its functionalities. Existing techniques do not consider many attributes of contexts where two or more organizations interact (interorganizational –IO-). An extension is proposed for i* framework (Yu, 1995) for analysis of IO information systems (IOSs). Essential use cases are also derived from the proposed model. Diverse transformation rules are explained and an example is instantiated.

1 INTRODUCTION

Diverse techniques exist to represent organizational environments. However there is still a gap when a set of organizations which interact to attain certain goals, conforming *IO networks* (IONs) must to be considered. Thus, techniques capable to express the richness of social, organizational and IO relations are needed.

This work proposes an extension for IO environments of i* framework Yu (1995). A new model is proposed describing IOS dependencies. Essential use cases (Wiegers, 1997) are also derived and their importance is analyzed for carrying out the initial steps of RE for IOSs.

2 i* FRAMEWORK

i* describes intentionalities through two models: Strategic Dependencies and Strategic Rationale.

2.1 Strategic Dependencies Model (SD)

Each node in SD is an actor. A link indicates that an actor (*depender*) depends on another actor (*dependee)* for something (*dependum*) in order to attain an objective. Actors depend among them to achieve goals, perform tasks or share resources.

2.2 Strategic Rationale Model (SR)

SR models actors' intentional relationships. Nodes are goals, tasks, resources and soft-goals connected by *means-ends* or *task-decomposition* links. A goal may be associated through means-ends links with multiple paths to achieve it, generally tasks.

Task-decomposition links express what must be done to perform a task. The task can be hierarchically decomposed in *subtasks*, *subgoals*, *resourceFor* and *softgoalFor*. Fig. 1 shows a SR for a health insurance claim process (Yu, 1995).

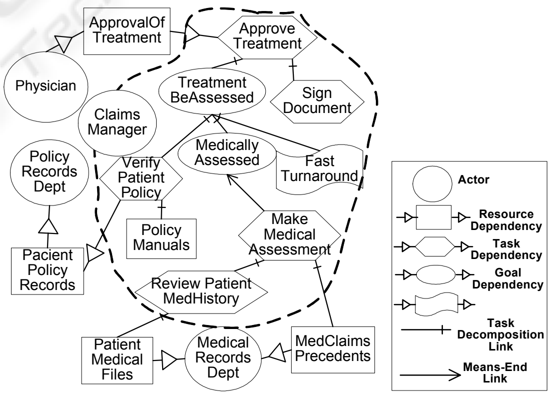

Figure 1: SR model example in the health area.

3 i* FRAMEWORK EXTENSION

3.1 SD Extension

Some elements are incorporated in SD model in order to show predominating ION properties.

Resource dependum specialization determines the *type of resource*, considering exchanges between

ION members: information, material, economic and knowledge resources (Fig. 2).

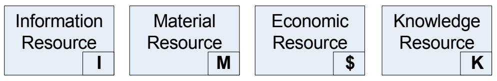

Figure 2: Resource types specification in SD model.

IONs analysis implies the incorporation of IO dimension, where actors also exist. Thus, dimensions differentiation in SD model is proposed through the inclusion of layers.

IONs have different structures (Table 1). Hierarchical, solar, and swingle ones have a leader member. In swingle ones leadership is temporal. Centreless IONs do not have a leader (Fig. 3).

Table 1: IONs types regardless their structure.

Hierarchical: Big firms subcontract processes to smaller firms. The central company coordinates activities.
Solar: A group of independent firms works around a strategic one that directs relations and controls satellite firms.
Centreless: Members operate in the same business area but cooperate and share resources.
Swingle: Organizations have key roles in different time periods. Network control rotate over time.

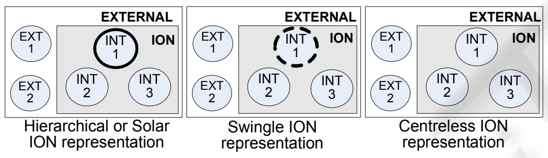
EXTERNAL
EXT 1
EXT 2
ION
INT 1
INT 2
INT 3
Hierarchical or Solar ION representation
Swingle ION representation
Centreless ION representation

Figure 3: IONs structures' specification in SD.

3.2 System Dependencies Model (SysD)

SR model is used for requirements analysis by assessing contributions in means-ends links for softgoals defined in SD, in order to know if they are satisfied. This analysis is based on the "goal oriented" approach for RE, to manage non-functional requirements (Mylopoulos et al., 1999).

Similarly to SR, SysD (from ***Sys***tem ***D***ependencies) considers SD explosion through means-ends and task-decomposition. A new detail level is added through the *stakeholder* concept. SysD cannot replace SR model: it works only with IOS supported dependencies, while SR has all dependencies underlying the environment.

SysD model is oriented to provide support in the capture of early functional requirements. Thus, it can be developed jointly with SR, in order to analyze particular types of requirements (SR → non-functional ones, SysD→ functional ones).

SysD considers each stakeholder interest (relation between his needs and project goals) and influence (relative power within a project) (Ballejos & Montagna, 2006). Then, 4 submodels are created from the combination of high (H) and low (L) interest and influence degrees: 1) H inf-H int, 2) H int-L inf, 3) L int-H inf, and 4) L int-L inf. Project manager decides which categories are analyzed. Each submodel uses information from SD model following these steps for each category:

1) A stakeholder category is determined.
2) *Depender* (having output dependencies) actors with stakeholders of the selected category must be analyzed. For each dependency, stakeholders must be associated with involved actors, in order to specify which stakeholder has dependencies from the system. A new detail level is included, since SD actors will be represented by stakeholders.
3) Dependencies without associated stakeholders or not supported by the system must be eliminated.
4) Means-ends and task-decomposition links are generated in order to describe system dependencies in SysD. All new tasks and resources supported by the system are illustrated with dotted lines.

SysD not only examines actors' dependencies through means-ends and task-decomposition links, but also differentiates means, dependencies, and subtasks supported by the IS. This is the previous stage to generate essential use cases specifications reducing the gap between IO modelling and early requirements analysis for IOSs development.

4 USE CASES FROM SysD MODEL

Use cases (UCs) are directly related with functional requirements of a software system, considering interactions between actors and the system under development. Essential UCs are stripped of implementation details, constraints, and alternatives. By focusing on them the analyst can derive software requirements (Wiegers, 1997).

Essential UCs have abstract, implementation-independent descriptions. Thus, heuristics are presented to derivate their descriptions from SysD.

While an actor in i* represents any person or organization with interactions with others, in UCs is a stakeholder who interacts with the system.

4.1 Resource Dependums

When the dependum is an information resource, UC name is "CUD Resource X" (CUD: create, update, delete) (Fig. 4). Actor 2 will use the system for the creation of the resource. STK will access to it through the system.

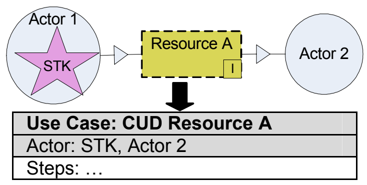

Figure 4: UC for information resource dependum.

With material or economic dependums the UC name is "Manage Resource X". Depender and dependee stakeholders are UC actors (Fig. 5).

When the dependum is an atomic task (without subtasks) which will be supported by the IOS, it directly transforms in a UC. The stakeholder from the depender actor is the UC actor (Fig. 6).

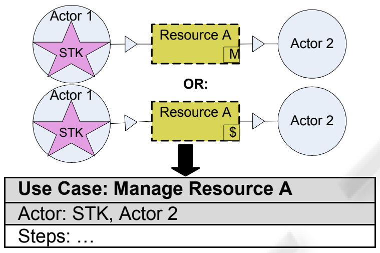

Figure 5: UC for material and economic dependums.

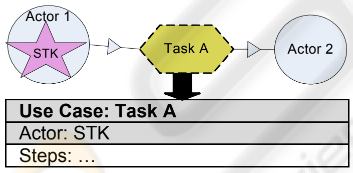

Figure 6: UC for task dependum.

4.2 Links Decomposition

When the dependum is a task decomposed in subtasks, the task is transformed in a UC where subtasks are steps. The stakeholder from the depender actor is the UC actor (Fig. 7).

An "*includes*" relation between UCs has place when one UC always includes the flow of another UC. Here it is created when some subtask will be also IOS supported, invocating the child UC in each execution of the principal one (Fig. 8).

In means-ends links the *end* can be a goal, a task, a resource, or a softgoal, while *means* is usually a

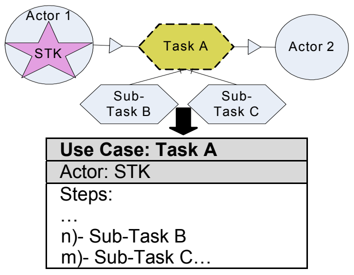

Figure 7: UC for task-decomposition links.

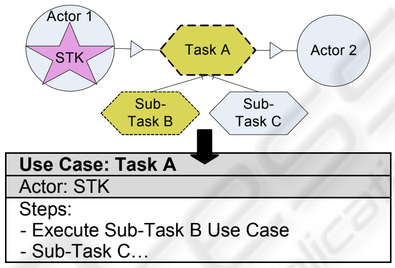

Figure 8: Includes relation derivation.

task. Different means indicates diverse ways of achieving the end. UC derivation focuses on links where the *end* is a task or a resource.

In means-ends links, the different means to execute a task or obtain a resource are alternatives. Thus, in both cases, *extends* relations (when an UC includes the flow of another UC under certain conditions) can be established between the "end" task UC and the "means" task UC (Fig. 9) or the UC related to the resource being the "end" and the task UC being "means" (Fig. 10).

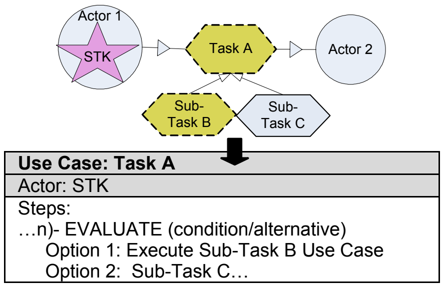

Figure 9: Extends relation for task ends.

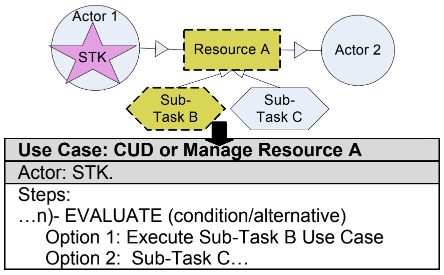
Actor 1
STK
Resource A
Actor 2
Sub-Task B
Sub-Task C
Use Case: CUD or Manage Resource A
Actor: STK.
Steps:
…n)- EVALUATE (condition/alternative)
Option 1: Execute Sub-Task B Use Case
Option 2: Sub-Task C…

Figure 10: Extends relation for resource ends.

5 EXAMPLE

An ION was created in an Argentinean province to manage medicines distribution to hospitals (Fig. 13), including the *Medicines Producer Laboratory* (MPL: produces generic medicines for the Central Pharmacy), the *Central Pharmacy* (CP: coordinates and controls medicines supplies to hospitals) and *Health Regions* (each one responsible for medicines distribution in *hospitals* and *health centers* depending on them). Also an IOS was proposed to transform the organizational systems into a unique model.

Fig. 11 shows extended SD model. Fig. 12 presents a partial SysD model, only considering dependencies between CP and Health Regions. Stakeholders with high interest and any value of influence were jointly selected. Their list is shown in the left bottom corner of Fig. 12. First, output dependencies for CP to be supported by the IOS are analyzed and then associated to stakeholders.

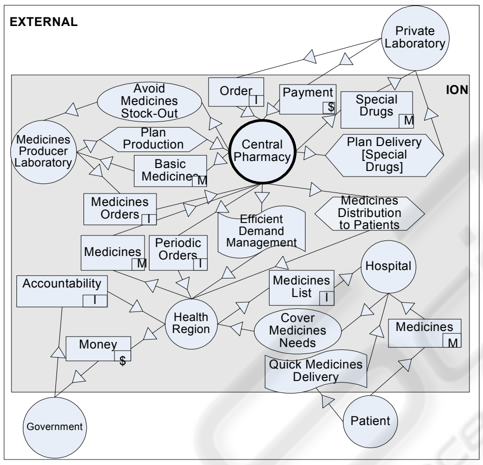

Figure 11: SD Model for the example.

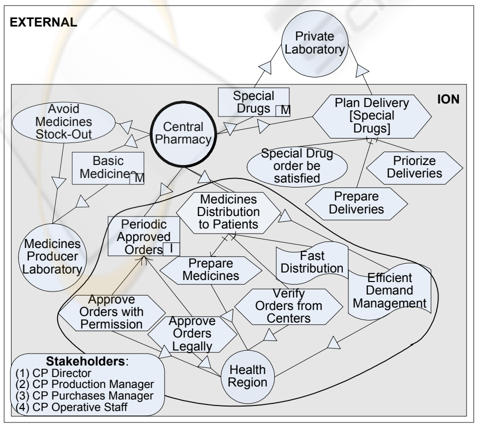
EXTERNAL
ION
Private Laboratory
Special Drugs M
Plan Delivery [Special Drugs]
Avoid Medicines Stock-Out
Central Pharmacy
Basic Medicine M
Special Drug order be satisfied
Priorize Deliveries
Medicines Distribution to Patients
Prepare Deliveries
Medicines Producer Laboratory
Periodic Approved Orders I
Fast Distribution
Prepare Medicines
Efficient Demand Management
Approve Orders with Permission
Verify Orders from Centers
Approve Orders Legally
Stakeholders:
(1) CP Director
(2) CP Production Manager
(3) CP Purchases Manager
(4) CP Operative Staff
Health Region

Figure 12: SysD Model for the example.

All non-system supported dependencies are deleted from the model. All system supported elements must be marked. Thus, the model in Fig. 13 is the result. Stars indicate stakeholders and are associated to IOS supported dependencies.
Fig. 14 shows UC documentations for Fig. 13.

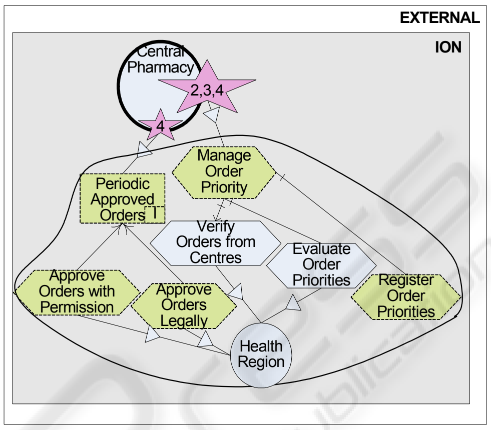
EXTERNAL
ION
Central Pharmacy
2,3,4
4
Manage Order Priority
Periodic Approved Orders I
Verify Orders from Centres
Evaluate Order Priorities
Approve Orders with Permission
Approve Orders Legally
Register Order Priorities
Health Region

Figure 13: SysD model with selected stakeholders.

6 CONCLUSIONS

This work extends i* framework for dealing with ION contexts in order to generate a systematic procedure to derive essential use cases. Thus, a preliminary use case diagram can be obtained for an IOS. This proposal works with knowledge modelled in the extended i* framework to obtain general descriptions for IOSs initial UCs indicating functionalities they should implement and reducing the gap between problem and solution domains.

Use Case: CUD Periodic Approved Orders
Actor: STK 4
Steps:…n)- EVALUATE (condition/alternative) **Option 1**: Execute UC "Approve Orders with Permission". **Option 2**: Execute UC "Approve Orders Legally".
Use Case: Approve Orders with Permission
Actor: STK 4
Steps:…
Use Case: Approve Orders Legally
Actor: STK 4
Steps:…
Use Case: Manage Order Priority
Actor: STKs 2, 3 and 4
Steps: 1)- Verify orders from centres. 2)- Evaluate order priorities. 3)- Execute UC "Register Order Priorities".
Use Case: Register Order Priorities
Actor: STKs 2, 3 and 4
Steps:…

Figure 14: UCs descriptions for the example.

ACKNOWLEDGEMENTS

The authors want to thank the financial support from CONICET, ANPCyT, UTN and Universia-Banco Río.

REFERENCES

Ballejos, L.C., & Montagna, J.M. (2006). Stakeholders Selection for Interorganizational Systems: A Systematic Approach. *IFIP 19th WCC.*

Wiegers, C. (1997). Listening to the customer's voice. *Software Development*, 5(3), pp. 49 – 55.

Yu, E. (1995). Modelling Strategic Relationships for Process Reengineering. *PhD Thesis,* Univ. of Toronto.