A Framework for the Definition of Metrics for Actor-Dependency Models

Xavier Franch, Gemma Grau, Carme Quer
Universitat Politècnica de Catalunya (UPC)
C/Jordi Girona 1-3, UPC-Campus Nord (C6), Barcelona (Spain)
{franch, ggrau, cquer}@lsi.upc.es
http://www.lsi.upc.es/~gessi/

1. Introduction

We define *actor-dependency models* as a restricted class of goal-oriented models in which we focus on the actors and dependencies that exist in a system. An example of actor-dependency models are *i** Strategic Dependency (SD) models [1]. We are interested in the structural analysis of actor-dependency models with respect some properties considered of interest for the modelled system (such as security, accuracy or efficiency), using some adequate metrics defined in terms of the elements of the model.

2. The Framework

An *actor-dependency model* comprises two types of elements, its *actors* and the *dependencies* among them. Dependencies connect source and target actors, called *depender* and *dependee* respectively. Fig. 1 presents an *i** SD model for a meeting scheduler system.
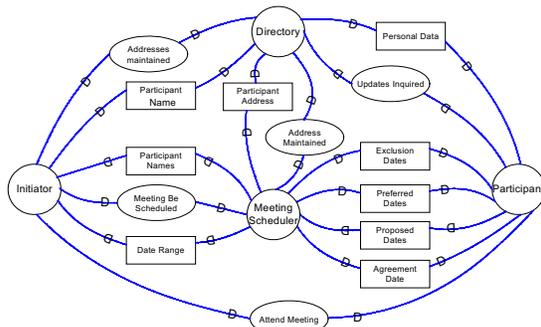

Figure 1. A i* SD model for a meeting scheduler system.

For our purposes, it is helpful to consider that actors and dependencies belong to one *sort* that group elements of the same kind; therefore we can talk about human actors, goal dependencies, and so on.

Definition 1. Actor-dependency model.

An actor-dependency model is a pair $M = (A, D)$, being A a set of actors and D the dependencies among them, such that:

1) The set A has a mapping $sort_A: A \rightarrow TA$, being TA the permissible sorts of actors.

2) D is defined as a set of ordered pairs of actors with the name of the dependency, $D \subseteq A \times A \times string$.

3) The set D has a mapping $sort_D: D \rightarrow TD$, being TD the permissible sorts of dependencies.

For a given model property object of measure, it may be the case that all its elements (actors and dependencies) influence the metric. However, it is more likely that just elements of some particular sorts affect this property. Furthermore, some individual elements may be identified as especially relevant for the property; in the most general case, all the elements may have a different weight in the metric. We need to take into account all these situations in order to have a widely applicable metrics formulation framework.

We distinguish 3 types of structural metrics. We focus in this paper on *global structural metrics* (see [2] for the 2 others), which take the model as a whole and produce a single measure for the property of interest. All types of metrics rely on two fundamental concepts, *actor* and *dependency evaluation*, depending on the kind of model element that is considered to influence most the metric. We choose one of them, dependency evaluation, to present the ideas of the framework; the actor evaluation case is defined similarly, see also [2].

Definition 2. Dependency evaluation.

Given a model property P, an actor-dependency model $M = (A, D)$ and a dependency inside the model, $d=(a, b, x) \in D$, the dependency evaluation of d for P over M is of the form $P_{M,D}(d) = f_{M,P}(d) \times g_{M,P}(d)$ being $f_{M,P}: D \rightarrow [0, 1]$ a mapping that assigns a weight for P to every dependency and $g_{M,P}: D \rightarrow [0, 1]$ a mapping that adjusts the weight for P of a dependen-cy considering its depender and dependee, respectively.

Dependency-based global structural metrics just sum the evaluations of its elements, and normalize the value considering the number of dependencies that satisfy a particular condition, using a function $limit_P$.

Definition 3. Dependency-based global structural metrics. Given a model property P, an actor-dependency model $M = (A, D)$ and a function $limit_P: D \to [1, \|D\|]$, a dependency-based global structural metric for P over M is of the form $P_M = \Sigma d \in D: P_{M,D}(d) / limit_P(D)$.

3. The Meeting Scheduler Case Study

In this section, we use our framework to assess the software meeting scheduler depicted in fig. 1 with respect to some non-functional goals, expressed as model properties: data privacy (DP), data accuracy (DA) and process agility (PA). Since the flow of data is a crucial factor in these properties, we use dependency-based global structural metrics.

Table 1 shows a feasible dependency evaluation. The values appearing therein belong to the interval [0, 1], being 1 the best possible case. We define $g_{M,P}(d)$ in terms of a function h on the actor's sort, $g_{M,P}((a, b, x)) = h_A(a) \times h_A(b)$, thus we show at the table h_A's value instead of $g_{M,P}$. For instance, we consider that software actors are better for data accuracy (not misunderstandings about meeting dates or assistants), while they hamper data privacy (computers are more vulnerable than personal agendas). We consider the four sorts of dependencies given by $i*$ (but include just one in the table, for brevity); two sorts of actors, human (H) and software (S); three derived types of dependencies, human-human (H-H), software-software (S-S) and human-software (S-H). Dependency evaluation depends only on this information. For instance, we consider that human to human communication is more private than human to software. We may use weighting techniques such as the AHP.

f_D	Dependency		Property		
	Sort	Type	DP	DA	PA
		H-H	1	0,6	0,5
	Resource	H-S	0,9	0,8	0,8
		S-S	0,8	1	1

h_A	Sort	DP	DA	PA
	H	1	0,7	0,8
	S	0,7	1	0,9

Table 1. Dependency evaluation for the metrics.

The definition of the metrics on top of these dependency evaluations consists on giving values to the function $limit_P$. In this case, we define $limit_P(D) = 1$ for data privacy and process agility, and $limit_P(D) = \| \{d \in D: sort_D(d) = resource\} \|$ for data accuracy (see [2] for a justification). Table 2 provides the evaluation of two alternatives for the system using these definitions and taking into account the complete definition of the metrics (table 1 is just an excerpt). We transport the results into the interval [0, 1], preserving the distances found in the measurement. The observation of the table shows that a human meeting scheduler supports data privacy but is worse with respect to data accuracy and process agility.

	Proposed system with human meeting scheduler	Proposed system with software meeting scheduler
DP	0,64	0,55
DA	0,63	0,91
PA	0,66	0,74

Table 2. Evaluation of the metrics for 2 cases.

4. Conclusions

The main purpose of this ongoing work is to provide a framework to analyse actor-dependency models in a systematic way with respect to a given set of model properties. The framework is expressive (see [2] for the complete definition), sound (based on formal definitions), cost-sensitive (metrics can be further refined accordingly to the scheduled effort), reusable (metrics can be used in different models of the same kind of domain) and general (not tightened to any particular goal-oriented language or formalism).

As related approaches, we mention concepts such as opportunity and vulnerability proposed by Yu [1] that could be modelled within our framework. Also, the AGORA method [3] provides techniques for estimating the quality of requirements specifications in a goal-oriented setting, but it does not provide any kind of general form of the metrics. The idea of using dependencies to analyse the behaviour of the system, as well as the importance to distinguish among human and software actors, is addressed in [4].

5. Acknowledgments

6. References

[1] E. Yu, *Modelling Strategic Relationships for Process Reengineering*, PhD. thesis, University of Toronto, 1995.

[2] X. Franch, G. Grau, C. Quer. "A Framework for the Definition of Metrics for Actor-Dependency Models" (extended version). Technical Report LSI-04-10-R, http://www.lsi.upc.es/dept/techreps/contenido.php, 2004.

[3] H. Kaiya, H. Horai, M. Saeki. "AGORA: Attributed Goal-Oriented Requirements Analysis Method". *Proceedings of the IEEE 10th RE*, Essen, Sept. 2002.

[4] A. Sutcliffe, S. Minocha. "Linking Business Modelling to Socio-Technical System Design". In *Procs. of the 11th CAiSE*, LNCS, Heildelberg, June 1999.