

SOCIOPATH: Bridging the Gap Between Digital and Social Worlds

Nagham Alhadad¹ Philippe Lamarre² Yann Busnel¹
Patricia Serrano-Alvarado¹ Marco Biazzi¹ Christophe Sibertin-Blanc³

¹ {Name.LastName}@univ-nantes.fr

² Philippe.Lamarre@liris.cnrs.fr

³ Christophe.Sibertin-Blanc@univ-tlse1.fr

¹ LINA/Université de Nantes ² Liris/Université de Lyon ³ IRIT/Université de Toulouse 1
2, rue de la Houssinière 37, Rue du Repos 2, rue du Doyen-Gabriel-Marty
44322 Nantes, France 69007 Lyon, France 31042 Toulouse, France

Abstract. Everyday, people use more and more digital resources (data, application systems, Internet, *etc.*) for all aspects of their life, like financial management, private exchanges, collaborative work, *etc.* This leads to non-negligible dependences on the digital distributed resources that reveal strong reliance at the social level. Users are often not aware of their real autonomy regarding the management of their digital resources. People underestimate social dependences generated by the system they use and the resulting potential risks. We argue that it is necessary to be aware of some key aspects of system's architectures to be able to know dependences. This work proposes SOCIOPATH, a generic meta-model to derive dependences generated by system's architectures. It focuses on relations, like access, control, ownership among different entities of the system (digital resources, hardware, persons, *etc.*). Enriched with deduction rules and definitions, SOCIOPATH reveals the dependence of a person on each entity in the system. This meta-model can be useful to evaluate a system, as a modeling tool that bridges the gap between the digital and the social worlds.

1 Introduction

Nowadays, the most widespread architectures belong to the domain of distributed systems [1]. Most of participants' activities on these systems concern their data (sharing and editing documents, publishing photos, purchasing online, *etc.*) [2]. Using these systems implies some relationships with a lot of people. These people may be partly unknown, but one depends on them in several ways:

- Which person(s)/resource(s) a user depends on to perform an activity?
- Whom can prevent a user from performing an activity?
- Which persons a user is able to avoid to perform an activity?
- Who is possibly able to access user's data?

Some of these questions raise several issues as someone may be able to grab information about who the user is, what the user does, and so forth. That directly leads to privacy [3], trust [4] and security issues [5].

The analysis of such systems is usually limited to technical aspects as latency, QoS, functional performance, failure management [6], *etc.* The aforementioned questions

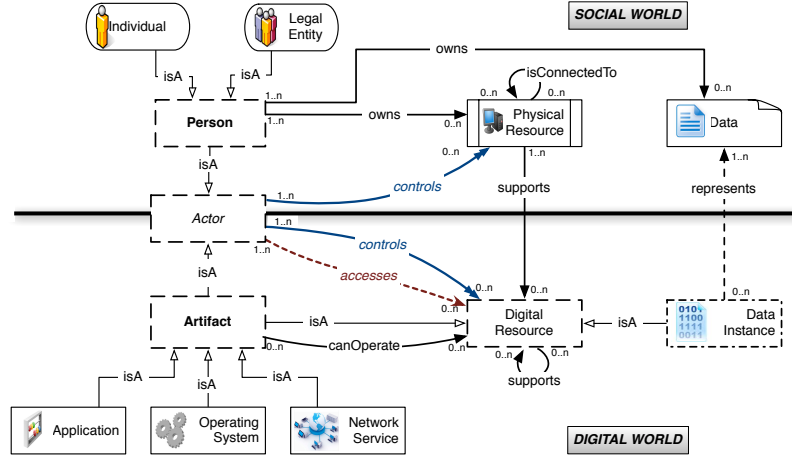


Fig. 1: Graphical view of SOCIOPATH as a UML class diagram.

give some orthogonal but complementary criteria to the classical approach. Currently, people underestimate social dependences [7,8,9] generated by the systems they use and the resulting potential risks. We argue that to be able to know dependences, one should be aware of some key aspects of systems.

This paper proposes the SOCIOPATH meta-model. This approach is based on notions coming from many fields, ranging from computer science to sociology. SOCIOPATH is a generic meta-model that considers two worlds: the social world and the digital world. SOCIOPATH allows us to draw a representation of a system that identifies persons, hardware, software and the ways they are related. Enriched with deduction rules, SOCIOPATH analyzes the relations in the digital world, to deduce the relations of dependences in the social world, given an activity concerning some data. As a modeling tool that bridges the gap between the digital and the social world, SOCIOPATH can be useful in the evaluation process of a system with respect to security, privacy, risk and trust requirements.

The paper is organized as follows. Section 2 introduces the SOCIOPATH meta-model and gives a simple example of its use. Section 3 defines the way to compute the user's digital and social dependences. Section 4 presents a brief overview of related works. Finally, Section 5 concludes and points out our ongoing work.

2 SOCIOPATH meta-model

The SOCIOPATH meta-model describes the architecture of a system in terms of the components people use to access digital resources, so that chains of dependences can be identified. It distinguishes the *social world*, where humans or organizations own physical resources and data, and the *digital world*, where instances of data are stored and processes run. Figure 1 shows the graphical representation of SOCIOPATH, which we analyze in the following sections.

2.1 The social world

The social world includes persons (users, enterprises, companies, *etc.*), physical resources, data and relations among them.

- *Data* represents an abstract notion that exists in the real life, and does not necessarily imply a physical instance (*e.g.*, address, age, software design, *etc.*);
- *Physical Resource* represents any hardware device (*e.g.*, PC, USB device, *etc.*);
- *Person* represents a generic notion that defines an Individual like Alice or a Legal Entity like Microsoft.

2.2 The digital world

The digital world has nodes that are defined as follows:

- *Data Instance* represents a digital representation of *Data*;
- *Artifact* represents an abstract notion that describes a “running software”. This can be an *Application*, an *Operating System* or a *Network Service*;
- *Digital Resource* represents an *Artifact* or a *Data Instance*;
- *Actor* represents a *Person* in the social world or an *Artifact* in the digital world.

2.3 The relations in SOCIOPATH

Several relations are drawn in SOCIOPATH. In this section, we briefly describe them.

- *owns*: this means ownership. This relation only exists in the social world;
- *isConnectedTo*: this means that two nodes are physically connected, through a network for instance. This symmetric relation exists only in the social world;
- *canOperate*: this means that an artifact is able to process, communicate or interact with a target digital resource;
- *accesses*: this means that an *Actor* can access a *Digital Resource*;
- *controls*: this means that an *Actor* controls a *Digital* or a *Physical Resource*. There exists different kinds of control relations. A legal entity, who provides a resource, controls its functionalities. The persons who use this resource have some kind of control on it;
- *supports*: this means that the target node could never exist without the source node;
- *represents*: this is a relation between data and their instances in the digital world.

Persons own some data in the social world. *Data* have a concrete existence in the digital world if they are represented by some *Data Instance* and supported by some *Physical Resource*. As an *Actor* in the digital world, a *Person* can access and control *Data Instances* representing her (and others’) *Data*. This may be done through different resources, thus implying some dependences on other persons. Moreover, we consider that a person *provides* an artifact (*cf.* the rightmost part of Figure 2) if this person owns data represented by a data instance which supports the artifact. Applying SOCIOPATH makes possible non-trivial deductions about relations among nodes. For instance, an actor may be able to access digital resources supported by different physical resources connected to each other (*e.g.*, a user can access processes running in different hosts).

Figure 2 shows a basic SOCIOPATH model¹ of a use-case on a unique PC. In the social world, a user John owns some *Data* and a PC. There are also legal entities as: Microsoft, provider of Windows, Word (MSWord) and Excel (MSExcel);

¹ We consider that a model conforms to a meta-model.

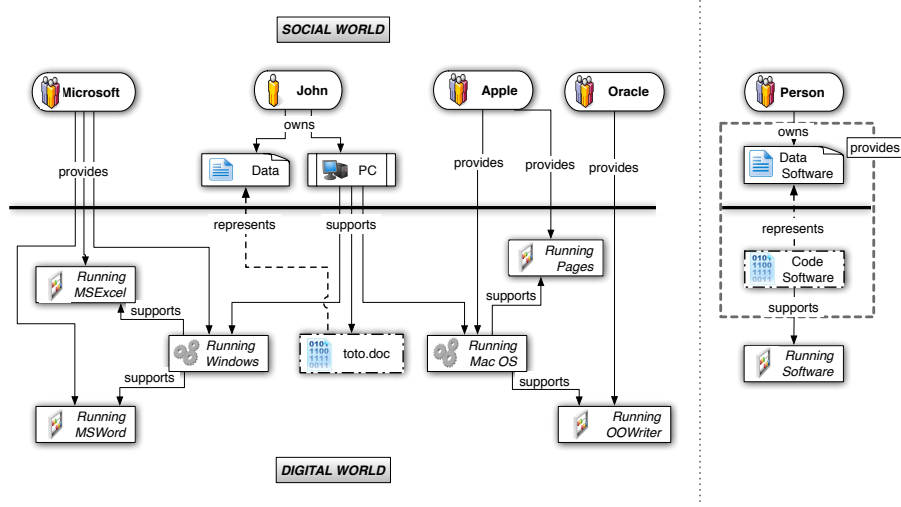


Fig. 2: Use case example: a document accessed by 2 different operating systems.

Basic type of instance	The set of all instances		A subset of instances		One instance	
	Notation	Remark	Notation	Remark	Notation	Remark
Person	\mathbb{P}	$\{P : person(P)\}$	\mathcal{P}	$\mathcal{P} \subset \mathbb{P}$	P	$P \in \mathbb{P}$
Actors	\mathbb{A}	$\{A : actor(A)\}$	\mathcal{A}	$\mathcal{A} \subset \mathbb{A}$	A	$A \in \mathbb{A}$
Artifact	\mathbb{F}	$\{F : artifact(F)\}$	\mathcal{F}	$\mathcal{F} \subset \mathbb{F}$	F	$F \in \mathbb{F}$
Digital resource	\mathbb{DR}	$\{DR : resource(DR)\}$	\mathcal{DR}	$\mathcal{DR} \subset \mathbb{DR}$	DR	$DR \in \mathbb{DR}$
Physical resource	\mathbb{PR}	$\{PR : phyresource(PR)\}$	\mathcal{PR}	$\mathcal{PR} \subset \mathbb{PR}$	PR	$PR \in \mathbb{PR}$
Data	\mathbb{D}	$\{D : data(D)\}$	\mathcal{D}	$\mathcal{D} \subset \mathbb{D}$	D	$D \in \mathbb{D}$
Data instance	\mathbb{DI}	$\{DI : dataInstance(DI)\}$	\mathcal{DI}	$\mathcal{DI} \subset \mathbb{DI}$	DI	$DI \in \mathbb{DI}$
Operating System	\mathbb{O}	$\{OS : operatingSystem(OS)\}$	\mathcal{O}	$\mathcal{O} \subset \mathbb{O}$	OS	$OS \in \mathbb{O}$
Path	\mathbb{I}	$\{\sigma : path(\sigma)\}$	\mathcal{I}	$\mathcal{I} \subset \mathbb{I}$	σ	$\sigma \in \mathbb{I}$
Activity	\mathbb{W}	—	—	—	ω	$\omega \in \mathbb{W}$
Set of activity restrictions	\mathbb{S}	$\{S = \mathcal{P}(\mathbb{R}^N)\}$	—	—	S	$S \in \mathbb{S}$

Table 1: Glossary of notations

Apple, provider of MacOS and Pages; and Oracle, provider of Open Office Writer (OOWriter). In the digital world, two operating systems exist on John's PC: Windows and MacOS. On Windows, two applications are available: MSWord and MSExcel. On MacOS are installed OOWriter and Pages. John's Data are represented in the digital world by the document *toto.doc*. We use this example to illustrate some deductions in the following. Table 1 summarizes the notations we use.

3 Using SOCIOPATH to derive dependences for an activity

SOCIOPATH features a set of rules we can apply to underline and discover chains of *accesses* and *controls* relations. Table 2 shows the formal definitions of some rules that we can deduce from the meta-model basic definitions. More details are given in [10]. In this work, we want to use these relations to better understand the “social and digital dependences” among entities in the model. Thus, informally, the sets of *digital* dependences of a person are composed by the artifacts a user passes through to reach a particular element. The sets of *social* dependences are composed by the persons who control these

Rule	Formal definition
Rule 1	$\forall F \in \mathbb{F}, \forall DR \in \mathbb{DR}, \forall PR1, PR2 \in \mathbb{PR} : \bigwedge \left\{ \begin{array}{l} canOperate(F, DR) \\ supports(PR1, F) \\ supports(PR1, DR) \\ supports(PR2, DR) \\ isConnectedTo(PR1, PR2) \end{array} \right\} \Rightarrow accesses(F, DR)$
Rule 2	$\forall P \in \mathbb{P}, \forall PR \in \mathbb{PR}, \forall OS \in \mathbb{O} : \bigwedge \left\{ \begin{array}{l} owns(P, PR) \\ supports(PR, OS) \end{array} \right\} \Rightarrow \bigwedge \left\{ \begin{array}{l} accesses(P, OS) \\ controls(P, OS) \end{array} \right\}$
Rule 3	$\forall F \in \mathbb{F}, \forall OS \in \mathbb{O} : \bigwedge \left\{ \begin{array}{l} supports(OS, F) \\ canOperate(OS, F) \end{array} \right\} \Rightarrow controls(OS, F)$
Rule 4	$\exists P \in \mathbb{P}, \exists D \in \mathbb{D}, \exists DI \in \mathbb{DI}, \exists F \in \mathbb{F} : \bigwedge \left\{ \begin{array}{l} owns(P, D) \\ represents(DI, D) \\ supports(DI, F) \end{array} \right\} \Rightarrow controls(P, F)$
Rule 5	$\forall A \in \mathbb{A}, \forall F \in \mathbb{F}, \forall DR \in \mathbb{DR} : \bigwedge \left\{ \begin{array}{l} accesses(A, F) \\ accesses(F, DR) \end{array} \right\} \Rightarrow accesses(A, DR)$
Rule 6	$\forall A \in \mathbb{A}, \forall F1, F2 \in \mathbb{F} : \bigwedge \left\{ \begin{array}{l} controls(A, F1) \\ controls(F1, F2) \end{array} \right\} \Rightarrow controls(A, F2)$
Rule 7	$\exists PR1, PR2 \in \mathbb{PR}, \exists OS1, OS2 \in \mathbb{O} : \bigwedge \left\{ \begin{array}{l} isConnectedTo(PR1, PR2) \\ supports(PR1, OS1) \\ supports(PR2, OS2) \end{array} \right\} \Rightarrow accesses(OS1, OS2)$

Table 2: Deduced access and control relations

artifacts. In the following, all those concepts are formally defined. The examples refer to Figure 2.

3.1 Activities and Paths

A user follows a path to perform an activity in a system. Some restrictions may be given to the ways the person might do their activity. For instance, if a person wants to read a .doc document, she must use an artifact that can “understand” this type of document, so activity is defined as follows.

Definition 1 (Activity).

We define an activity ω as a triple (P, D, S) , where P is a person, D is data and S is a set of ordered multisets of \mathbb{F} in a model, so an activity ω is an element of $\mathbb{P} \times \mathbb{D} \times \mathbb{S}$. The sets in the \mathbb{S} component of an activity are alternative sets of artifacts that are necessary for the person to perform her activity. For instance, the activity “John edits toto.doc” is defined as $\omega = (John, Data, \{\{MSWord\}, \{Pages\}, \{OOWriter\}\})$.

We call *paths* the lists of actors and digital resources that describe the ways an actor may access a resource. A person may perform an activity in different ways and using different intermediate digital resources. Each possibility can be described by a path.

Definition 2 (Activity path, or ω -path).

A path σ for an activity $\omega = (P, D, S) \in \mathbb{P} \times \mathbb{D} \times \mathbb{S}$ is a list of actors and digital resources such that:

- $\sigma[1] = P$;
- $\sigma[|\sigma|] = D$;
- $represents(\sigma[|\sigma| - 1], \sigma[|\sigma|])$;
- $\forall i \in [2 : |\sigma| - 1], artifact(\sigma[i]) \wedge accesses(\sigma[i - 1], \sigma[i])$;
- $\exists s \in S, s \subseteq \sigma$;

where $\sigma[i]$, denotes the i^{th} element of σ , and $|\sigma|$ the length of σ .

Notation: Assuming that there is no ambiguity on the model under consideration, the set of ω -paths where $\omega = (P, D, S)$ is noted Υ^ω and the set of all the paths in the model is noted Υ .

For example, concerning the ω -paths for the activity $\omega = \text{“John edits toto.doc”}$ in Figure 2, we have:

$$\left\{ \begin{array}{l} \{\text{John, Windows, MSWord, Windows, MSeExcel, Windows, toto.doc}\} \\ \{\text{John, MacOS, OOWriter, MacOS, toto.doc}\} \end{array} \right\}.$$

In the first ω -path presented above, MSeExcel is not mandatory to edit `toto.doc`. It appears in the ω -path because of the relation *accesses* between it and the artifact Windows. In order to exclude all the unnecessary elements from the ω -path, so we define the minimal paths as follows.

Definition 3 (Minimal path).

Let Υ^ω be a set of paths for an activity ω .

A path $\sigma \in \Upsilon^\omega$ is said to be minimal in Υ^ω iff there exists no path σ' such that:

- $\sigma[1] = \sigma'[1]$ and $\sigma[|\sigma|] = \sigma'[|\sigma'|]$;
- $\forall i \in [2 : |\sigma'|], \exists j \in [2 : |\sigma|], \sigma'[i] = \sigma[j]$.

Notation: The set of minimal paths enabling an activity $\omega = (P, D, S)$ is noted $\widehat{\Upsilon}^\omega$. For sake of simplicity, we name this set the ω -minimal paths.

For instance, for the activity $\omega = \text{“John edits toto.doc”}$, the set of the ω -minimal paths are:

$$\widehat{\Upsilon}^\omega = \left\{ \begin{array}{l} \{\text{John, Windows, MSWord, Windows, toto.doc}\} \\ \{\text{John, MacOS, OOWriter, MacOS, toto.doc}\} \\ \{\text{John, MacOS, Pages, MacOS, toto.doc}\} \end{array} \right\}.$$

Notation: Let say $F \in \sigma$ iff $\exists i$ such that $\sigma[i] = F$, and $s \subseteq \sigma$ iff $\forall F \in s, F \in \sigma$.

3.2 Dependence

We can now introduce the definitions of digital dependences (Definition 4 and 5) and social dependences (Definition 6 and 7). We say that a person depends on a set of artifacts for an activity ω if each element of this set belongs to one or more paths in the set of the ω -minimal paths.

Definition 4 (Person’s dependence on a set of artifacts for an activity).

Let $\omega = (P, D, S)$ be an activity, \mathcal{F} be a set of artifacts and $\widehat{\Upsilon}^\omega$ be the set of ω -minimal paths. P depends on \mathcal{F} for an activity ω iff $\forall F \in \mathcal{F}, \exists \sigma \in \widehat{\Upsilon}^\omega : F \in \sigma$.

For instance, one of the sets on which John depends for the activity “John edits `toto.doc`” is $\{\text{MacOS, MSWord}\}$.

A person does not depend in the same way on all the sets of artifacts. Some sets may be avoidable *i.e.*, the activity can be executed without them. Some sets are unavoidable *i.e.*, the activity cannot be performed without them. To distinguish the way a person depends on artifacts, we define the degree of a person’s dependence on a set of artifacts for an activity as the ratio of the ω -minimal paths that contain these artifacts.

Definition 5 (Degree of person dependence on a set of artifacts for an activity).

Let $\omega = (P, D, S)$ be an activity, \mathcal{F} be a set of artifacts and $\widehat{\Upsilon}^\omega$ be the set of ω -minimal paths and $|\widehat{\Upsilon}^\omega|$ is the number of the minimal ω -paths. The degree of dependence of P on \mathcal{F} , denoted $d_{\mathcal{F}}^\omega$, is:

$$d_{\mathcal{F}}^\omega = \frac{|\{\sigma : \sigma \in \widehat{\Upsilon}^\omega \wedge \exists F \in \mathcal{F}, F \in \sigma\}|}{|\widehat{\Upsilon}^\omega|}$$

For instance, the degree of dependence of John on the set $\{\text{MacOS}, \text{MSWord}\}$ for the activity “John edits `toto.doc`” is equal to one, while the degree of dependence of John on the set $\{\text{Pages}, \text{OOWriter}\}$ is equal to $2/3$.

From digital dependences we can deduce social dependences. A person depends on a set of persons for an activity if the persons of this set control some of the artifacts the person depends on.

Definition 6 (Person’s dependence on a set of persons for an activity).

Let $\omega = (P, D, S)$ be an activity, and \mathcal{P} be a set of persons.

$$P \text{ depends on } \mathcal{P} \text{ for } \omega \text{ iff } \bigwedge \left\{ \begin{array}{l} \exists \mathcal{F} \subset \mathbb{F} : P \text{ depends on } \mathcal{F} \text{ for } \omega \\ \forall F \in \mathcal{F}, \exists P' \in \mathcal{P} : \text{controls}(P', F) \end{array} \right.$$

For instance, one of the sets John depends on for the activity “John edits `toto.doc`” is $\{\text{Oracle}, \text{Apple}\}$.

The degree of a person’s dependence on a set of persons for an activity is given by the ratio of the ω -minimal paths that contain artifacts controlled by this set of persons.

Definition 7 (Degree of person’s dependence on a set of persons for an activity).

Let $\omega = (P, D, S)$ be an activity, \mathcal{P} be a set of persons and $\tilde{\gamma}^\omega$ be the ω -minimal paths. The degree of dependence of P on \mathcal{P} , noted $d_{\mathcal{P}}^\omega$ is:

$$d_{\mathcal{P}}^\omega = \frac{|\{\sigma : \sigma \in \tilde{\gamma}^\omega \wedge \exists P' \in \mathcal{P}, \exists F \in \sigma, \text{controls}(P', F)\}|}{|\tilde{\gamma}^\omega|}$$

For instance, the degree of dependence of John on the set $\{\text{Oracle}, \text{Apple}\}$ for the activity “John edits the `toto.doc`” is equal to $2/3$. We recall that `Oracle controls OOWriter` and `Apple controls MacOS`.

4 Related work

Frameworks and tools modeling IT (Information Technology) systems are widely used in the context of Enterprise Architecture Management (EAM) [11]. EAM aims at giving a structured description of large IT systems in terms of their business, application, information and technical layers, with the additional goal of understanding how existing architectures and/or applications should be changed to improve business or strategic goals. SOCIOPATH rather focuses on dependence relationships, although converging with these frameworks in some aspects. For instance, RM-ODP (Reference Model of Open Distributed Processing) [12], is a generic set of standards and tools to create and manage aspect-oriented models of systems. RM-ODP analyzes and decomposes the systems in detail, mainly focusing on standard compliance. Aiming at different goals, SOCIOPATH gives a simpler overview of a system that is meant to inform the users about the relations that are implied by the system architecture, without exposing technical details. TOGAF (The Open Group Architecture Framework) [13], an approach for designing, planning, implementation, and governance of an enterprise information architecture, converges with SOCIOPATH in some of the concepts used to model the technical architecture of the enterprise. Unlike SOCIOPATH, TOGAF focuses on several aspects of the software engineering process, while describing the hardware, software and network infrastructures.

None of these works considers dependence relations among users in the social world. SOCIOPATH aims at improving awareness and exposing the relations of dependence generated on the social world from the digital world. More details about the related works are available in [10]

5 Ongoing work and Conclusion

SOCIOPATH allows to deduce the degree of dependence of a person on the entities of a system architecture for an activity. With SOCIOPATH it is possible to know, for instance, who can prevent a person to perform an activity or whom a person can avoid while performing an activity.

Our ongoing work concerns several aspects. Currently SOCIOPATH does not distinguish the different kinds of access and control of an actor to a digital resource. In order to consider intentions and expectations of users regarding digital resources, SOCIOPATH can be enriched with access and control typologies, to define different kinds of dependences. Moreover, no difference is made between what persons can do and what they are allowed to do according to the law, the moral rules *etc.* We aim at distinguishing between dependences related to the system's architecture, and dependences related to social commitments. Finally, the results of SOCIOPATH can be related with the notion of trust of a person toward the persons she depends on. Indeed, if a user does not trust some person, she will be concerned about architecture-induced dependences on this person, whereas if she trusts a person, she will only be concerned about commitments-related dependences.

References

1. Coulouris, G., Dollimore, J., Kindberg: Distributed Systems: Concepts and Design. Second edition edn. Addison Wesley Publishing Company, Harlow, England (1994)
2. Weaver, A.C., Morrison, B.B.: Social Networking. *Computer* **41** (2008) 97–100
3. Westin, A.F.: Privacy and Freedom. Sixth edition edn. Atheneum, New York (1970)
4. Marti, S., Garcia-Molina, H.: Taxonomy of Trust: Categorizing P2P Reputation Systems. *Computer Network Journal* **50** (2006) 472–484
5. Lampson, B., Abadi, M., Burrows, M., Wobber, E.: Authentication in Distributed Systems: Theory and Practice. *ACM Transactions on Computer Systems* **10** (November 1992) 265–310
6. Aurrecochea, C., Campbell, A.T., Hauw, L.: A Survey of QoS Architectures. *Multimedia Systems* **6** (1996) 138–151
7. Emerson, R.M.: Power-Dependence Relations. *Amer. Sociological Review* **27**(1) (1962) 31–41
8. Molm, L.: Structure, Action, and Outcomes: The Dynamics of Power in Social Exchange. American Sociological Association Edition (1990)
9. Blau, P.: Exchange and Power in Social Life. John Wiley and Sons (1964)
10. Alhadad, N., Lamarre, P., Busnel, Y., Serrano-Alvarado, P., Biazzi, M., Sibertin-Blanc, C.: SOCIOPATH: In Whom You Trust? Technical report, LINA – CNRS : UMR6241 (September 2011)
11. Goethals, F., Lemahieu, W., Snoeckand, M., Vandenbulcke, J.: An Overview of Enterprise Architecture Framework Deliverables. In: Banda RKJ (ed) Enterprise Architecture-An Introduction, ICFAI University Press (2006)
12. Farooqui, K., Logrippo, L., de Meer, J.: The ISO Reference Model for Open Distributed Processing: An Introduction. *Computer Networks and ISDN Systems* **27**(8) (1995) 1215–1229
13. TOGAF: Welcome to TOGAF - The Open Group Architectural Framework. <http://pubs.opengroup.org/architecture/togaf8-doc/arch> (2002)