SocioPath: In Whom You Trust?

Nagham Alhadad, Philippe Lamarre, Patricia Serrano-Alvarado, Yann Busnel, Marco Biazzini

SocioPath: In Whom You Trust?

Nagham Alhadad Philippe Lamarre Patricia Serrano-Alvarado Yann Busnel
Marco Biazzini

{Name.Lastname}@univ-nantes.fr
LINA / Université de Nantes
2, rue de la Houssiniére
BP 92208 – 44322 Nantes, France

Abstract

Distributed systems are getting more and more numerous, complex and used in a wide variety of applications. New solutions and new architectures arise (*e.g.*, clouds) that support new functionalities (*e.g.*, social networks) and pile up several software layers. This evolution implies new non negligible *dependences* among actors in the system (*e.g.*, providers and users). Some undesirable dependences could be hidden by this layer stacking, implying a reduced transparency for users. Thus, any software is directly dependent of the underlying layers. If one of these layers misbehave, the given software may be unable to provide promised services. We argue that users should be aware of the potential risks resulting from their dependences. To be able to deduce those dependences, one should know the way the system works (architecture, involved resources, providers, participants, *etc.*). This will help to deduce the potential trust a user should have toward the system. We consider this of utmost importance as, professional efficiency and personal privacy could be compromised if untrusted actors control the access to resources. This work proposes SocioPath, a generic meta-model that allows to expose hidden or implied relationships between actors of the system. SocioPath helps to make evident dependences among participants in the digital world which also introduces dependences at a sociological level. The notions presented in this approach are underneath many fields as security, privacy, trust, sociology, economy and so forth. SocioPath can be used in the evaluation process of a system as well as in its upstream design.

1 Problem definition

A large number of distributed systems arise nowadays that are more and more complex and used for a tremendous variety of applications. Actually, solutions proposed to users evolve toward new functionalities (*e.g.*, social networks), new architectures supporting the latter (*e.g.*, clouds) and piling up several software layers. This evolution implies new non negligible *dependences* among providers and actors in the system .

When users need to choose a system, they are overwhelmed by the plethora of free and lucrative available options. To make a choice, they *evaluate* systems depending on their needs, their objectives and the time they have to make a choice. Traditionally, evaluation covers functional, technical and economical aspects. From the functional point of view, users analyze the quality of provided services and the easy-to-use. Technical aspects orient the evaluation to performance criteria (like response time, reliability, availability, safety, security, *etc.*) but also to installation and maintenance requirements. Moreover, economical aspects must be considered, like the necessary investments to start and to assure a durable using.

From one user to another, those evaluation criteria have different weights and consequences and for the same application needs, different systems may be chosen. For instance, consider two users that need a software to edit and share documents. One user, working in a non-profit organization, may choose a freeware and disregard

performance characteristics, while another, which considers himself unskilled, may choose easy-to-use software without installation and maintenance efforts. Thus, the first one may decide to use OpenOffice[1] and the second one GoogleDocs[2]. Such decision implies several things, for instance, besides having a legal and running version of an operating system, the unskilled user is constrained to have an Internet connectivity continuously. Thus, functionalities and performance of any software is directly dependent of the underlying layers. If one of these layers misbehave, the given software may be unable to provide promised services.

In general, as users, we assume software developers and device manufacturers have the best intentions. When using a system, relationships of *trust* between users and providers are drawn. We notice that users are not always aware of those implicit relationships. A kind of 'trust among participants' can be constructed, based on the quality of their exchanges [1, 2, 3, 4, 5]. A general trust toward the system can be based on this, but also on the trust toward resources (data, programs, communications, etc.) and providers [6, 7, 8, 9]. In this work we are interested in another point of view of trust. We argue that users should be aware of the potential risks resulting from their dependences on systems and/or participants. And to be able to deduce those dependences they should know the way the system works (used architecture, involved resources, providers, participants, *etc.*). This will help to deduce the potential trust a user should have toward the system. We consider this very important, because by using systems, the professional efficiency and the personal privacy can be compromised if untrusted actors *control* the access to your resources.

SocioPath, the meta-model proposed in this work, helps to make evident the dependence relationships among participants of the sociological and digital world composing the system. The idea is that deduced relationships underline the potential repercussions of the trust users have toward the system in terms of security, privacy, sociology, economy, *etc.* Thus, when analyzing a system, we should take into account functional, technical, economical but also dependence aspects.

In the next, Section 2 introduces SocioPath and Section 3 talks about our ongoing work and gives some conclusions.

2 The SocioPath meta-model

SocioPath, the meta-model proposed in this work, aims at providing a formalism to help the user to answer questions related to her relationships in the sociological (dependency on persons) and in the digital world (dependency on resources). Here we are interested on what the persons can and cannot do (what they are able to do), rather than on what they have the right to do. Some of those questions are:

1. On whom the user depends to access her data?
   If a user stores her data instances on a server, she depends on the provider of the server and on the person who owns the server.

2. On which applications the user depends to access her data?
   When data instances are stored outside the user's computer, she may access her data through FTP clients, web browsers, *etc.*

3. Who can access user's data?
   When a user stores her data instances on a server, the administrator of the server and the service provider can access her data.

4. Through which resources somebody else can access the user's data?
   Those persons on whom a user depends to access her data can access her data if they collude.

5. What are the necessary coalition between persons to access to a particular data instance?

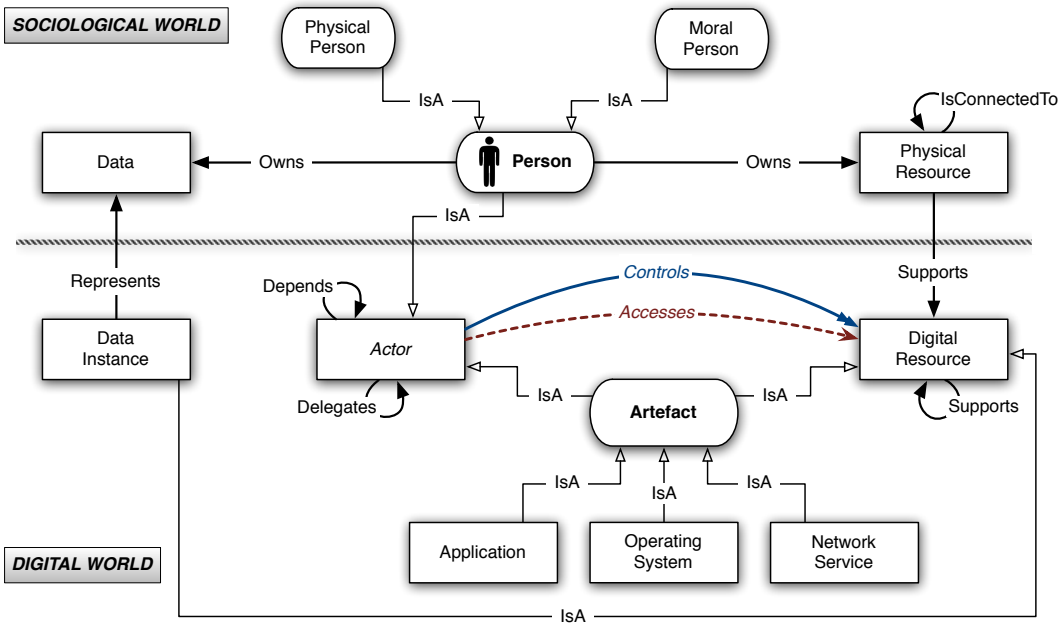

Figure 1: The Meta Model

Figure 1 shows the SocioPath meta-model which can be seen as a tool that helps to reflect the reality of two worlds that come together and interact between each other. It has several relations between nodes:

- *IsA* means that the source node is a specialization of the target node, *e.g.*, a physical person is a specialization of person.
- *Owns* means property. This relation exists only in the sociological (or physical) world.
- *Supports* means that the target node could never exist without the source node. It is a kind of 'allowing existence', *e.g.*, a running operating system exists only if it is hosted in a given hardware. When a digital resource supports another digital resource we can consider that the supporter can 'access' the supported one.
- *Represents* is a relation that exists between a datum in the sociological world and its existence on the digital one.
- *IsConnectedTo* means that two nodes are physically connected. This relation exists only in the sociological world.

The sociological world describes physical and moral persons (enterprises, companies, *etc.*), resources, data and the relations among them.

- A *Person* owns data. *Data* is an abstract notion that does not imply necessarily a physical instance (*e.g.*, an address, a software, *etc.*).
- A *Physical Resource* is a hardware, *e.g.*, PC, USB devices.
- A person *owns* physical resources.
- Physical resource (may be) *Is connected to* another physical resource.

[1] http://www.openoffice.org/
[2] https://docs.google.com/

The digital world has nodes characterizing actors, digital resources, artifacts, data instances, operating systems, networks services and applications.

- A *Data Instance* is a digital representation of a datum. It may be semantically equivalent to a datum existing in the sociological world. For instance, a person has an address (Data) in the sociological world. Whenever she writes it in a file, she creates a semantically equivalent digital instance of her address in the digital world (Data Instance). To express this relation, we say that a data instance *represents* a data.
- An *Artifact* can be a Network Service, an Operating System or an Application. We mean all of them to be "running software", thus considering them only in that they are being executed. By *Application* we mean a whole running entity. It may be a single process or a group of processes that may even be distributed in different locations, yet defining a single logically coherent entity.
- A *Digital Resource* can be an artifact or a data instance.
- A digital resource can *support* another digital resource, *e.g.*, an application is supported by the operating systems that hosts it.
- A digital resource is always supported by a physical resource, *e.g.*, no file would exist without its physical support.
- An *Actor* can be a person or an artifact.
- An actor can access or control a digital resource. *Access* relations we consider are read, write and execute. *Control* relations may be of different types. For instance, a moral person, who provides a resource to other persons, controls the functionality of this resource. The persons who uses this resource has some control on it as well. Each of these actors controls the resource in a different way. In this work we do not focus on the different types of control; we rather consider the control notion in a general meaning. Part of our future work will be devoted to specify the types of control.
- An actor can *delegate* to another actor some access types or control on a resource.
- An actor can *depend* on another actor to perform an activity, *e.g.*, a person depends on Google when she accesses her data instances by using the GoogleDocs application.

By applying the SocioPath meta-model it is possible to make 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 may access processes running on different hosts.

Every person owns data in the sociological world. These data have a concrete existence in digital world if they are represented by data instances, supported by physical resources (*e.g.*, CD, DVD, PC, *etc.*). As an actor in the digital world, a person can access and control data instances representing her (and others') data. This may possibly be done through chains of delegations, or accessing different resources, which implies some dependence on other persons.

In this work we are particularly interested in formalizing the relations in the digital world to derive the dependences between persons in sociological world.

GoogleDocs example

To illustrate the meta-model, Figure 2 presents a model drawn by applying SocioPath. This model represents a system on which a user uses GoogleDocs.

In the sociological world there is the person John that owns some data and a PC. There are also Microsoft (the provider of Windows and Internet Explorer) and Google (the provider of GoogleDocs and Google Cloud).

In the digital world, the Windows operating system is running on John's PC and it supports Internet Explorer. John's data are represented in the digital world with the document toto.doc which is supported by the physical resources owned by Google. We consider Google Cloud as the storage system used by the application GoogleDocs.

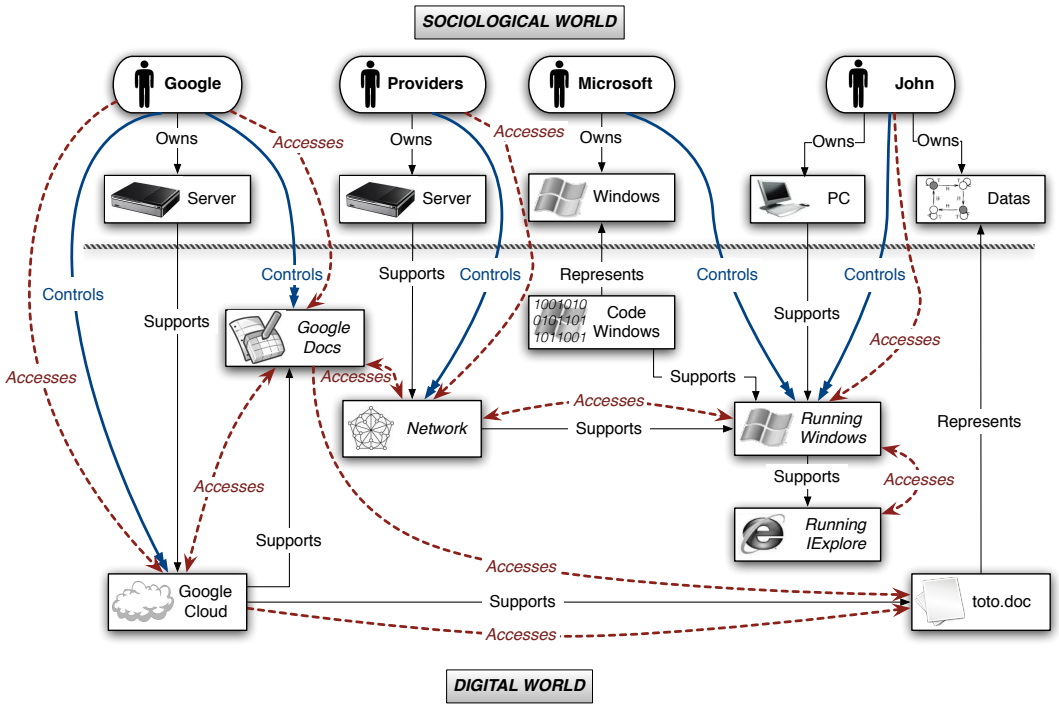

Figure 2: GoogleDocs

According to SocioPath and to some rules[3], John accesses Windows, which is supported by his PC. As Windows supports Internet Explorer, this first one can access Internet Explorer.

The physical resource that supports the network is connected to John's PC and to the physical resources owned by Google, so the network services, GoogleDocs and Windows may access John's PC. The Google Cloud supports GoogleDocs that can access the file toto.doc.

Thus, if John wants to access his document, he passes by Windows, then by Internet Explorer, then by Windows again, and through the network he arrives to GoogleDocs, then to Google Cloud, and finally to his document.

According to the meta-model and its rules we can answer the questions introduced at the beginning of this section.

1. On whom John depends to access his data? On Microsoft, the network providers and Google.
2. On which applications John depends to access his data? On Windows, Internet Explorer, the network, GoogleDocs and Google Cloud.
3. Who can access John's data? Microsoft, the network providers and Google.
4. Through which resources they can access the John's data? For Microsoft, through Windows, the network, Google Docs and Google Cloud. For the network provider, through the network, Google Docs and Google Cloud. For Google only through Google Cloud.
5. What are the necessary coalition between persons to access toto.doc? The coalition should be done among the ones that answer each point of the question 4. That makes evident that Google needs to collude only with himself.

[3]Due to space limitations those rules are not included in this paper.

3 Ongoing work and Conclusion

This paper introduced SocioPath, a meta-model to build system models with the goal of making evident the dependence relations among participants. The graphical view of SocioPath (see Figure 1) is very useful to understand the global idea. To complete and formally express SocioPath we have defined some rules based on first order logic (*e.g.*, any artifact is supported by an operating system, a person who owns a physical resource that supports an operating system accesses this operating system) and basic algebraic definitions (*e.g.*, path, order over paths, minimal path, dependency, degree of dependence). Those rules and definitions are being programmed in ProLog. The goal is to have a tool, based on SocioPath, to automatically infer dependences.

Such a tool may be very valuable in all the situations that require a person to evaluate the degree of interdependence of the various components of a given architecture. *E.g.*, it may help a manager in understanding all the implications entailed by decisions such as: switching from a corporate licensed software to an open source alternative; choosing a network service provider over a competitor one, validating a risk assessment plan for a given logistic architecture, *etc.*

Moreover, by applying SocioPath to build a model of her own system, a user may easily evaluate the 'cost' of changing something, in terms of side-effects and dependence shifts. One may also be able to evaluate the system's exposition to risks of misbehavior or failures of components the system itself depends on.

SocioPath can be used to point out access, control and relations within an architecture. This is particularly useful to check whether the system respects the trust and privacy expected by its users. Being able to test an architecture compliance with respect to users' privacy policies and trust models is one of our future goals.

We believe these are just few possible ways our meta-model can be useful to people in the real world, for them to better analyze and develop a useful understanding of consequences of the digital world everyone is more and more relying upon.

References

[1] Cornelli, F., Damiani, E., di Vimercati, S.D.C., Paraboschi, S., Samarati, P.: Choosing Reputable Servents in a P2P Network. In: Int. Conference on World Wide Web (WWW). (2002) 376–386

[2] Damiani, E., di Vimercati, D.C., Paraboschi, S., Samarati, P., Violante, F.: A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In: Int. Conference on Computer and Communications Security (CCS). (2002) 207–216

[3] Fahrenholtz, D., Lamersdorf, W.: Transactional Security for a Distributed Reputation Management System. In: Int. Conference on E-Commerce and Web Technologies (EC-WEB). (2002) 214–223

[4] Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust Algorithm for Reputation Management in P2P Networks. In: Int. Conference World Wide Web Conference (WWW). (2003) 640–651

[5] Gupta, M., Judge, P., Ammar, M.: A Reputation System for Peer-to-Peer Networks. In: Int. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV). (2003) 144–152

[6] Carchiolo, V., Longheu, A., Malgeri, M.: Reliable Peers and Useful Resources: Searching for the Best Personalised Learning Path in a Trust and Recommendation-Aware Environment. Information Sciences **180** (2010) 1893–1907

[7] Wang, L., Hill, R.: Trust Model for Open Resource Control Architecture. In: Int. Conference on Computer and Information Technology (CIT). (2010) 817–823

[8] Yoon, J.P., Chen, Z.: Service Trustiness and Resource Legitimacy in Cloud Computing. In: Int. Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC). (2010) 250–257

[9] Varalakshmi, P., Nandini, M., Krithika, K., Aarthi, R.: An Optimal Trust Based Resource Allocation Mechanism for Cross Domain Grid. In: Int. Conference on Recent Trends in Business Administration and Information Processing (BAIP). (2010) 342–348