Analyzing Second-Order Dependencies in i*

Mohammad Hossein Danesh[1,] Eric Yu [2, 1]

[1]Department of Computer Science, University of Toronto, Toronto, Canada
[2]Faculty of Information, University of Toronto, Toronto, Canada

{danesh,eric}@cs.toronto.edu

Abstract. Dependencies among intentional actors is a fundamental feature of i* Modelling. By depending on others, an actor can achieve much beyond what it can by itself. At the same time, the dependent actor becomes vulnerable to the failings of dependees. However, even when dependees are fully fulfilling expectations, over time, depending on other actors can result in structures that are hard to change. By analyzing second-order dependencies, i.e., dependencies among (first-order) dependencies, we determine the extent to which a dependency is depended on by other dependencies. The more a dependency is depended on by other dependencies, the more likely it is to become a barrier to change. Our approach is a model-based formulation of the concept of rigidity in the study of dynamic capabilities in strategic management. The i* models are used to analyze resistance to change in socio technical structures.

Keywords: i* Dependencies, Barriers to Change, Second-order Dependencies

1 Introduction

The importance of dealing with change and enabling adjustment to changing requirements has been studied in both management and Information System (IS) design [1, 2]. The importance of alignment and realignment of business and technical architectures with respect to changes is identified by the literature [3]. As a result, it is crucial to consider the intertwined nature of business and IS when modeling and representing enterprise requirements [4]. The challenge of dealing with change is twofold: (1) the ability to identify changing conditions and adjust to satisfy new requirements (either automated or with human intervention); and (2) the flexibility of enterprise capabilities and organizational settings to accommodate change, create new services or information systems and support their deployment [5].

While many researchers in IS and software engineering have attempted to overcome the first challenge, not many approaches exist that can analyze social and technical inflexibilities in an enterprises [6]. In this paper a model-based formulation of potential inflexibilities is presented using i* models that describe enterprise capabilities, their dependencies and alternatives [7]. The formulation investigates the structure of dependencies among capabilities (modeled as specialized actors) and other actors within the organization to analyze the commitments resulting from networks of dependencies. This formulation is motivated by research in strategic management about the positive and negative consequences of collaboration [8]. While collaboration can produce better qualitative and quantitative achievements, it also entails vulnerability as actors committed to such dependencies become confined in their future alternatives [8, 9]. The analysis of potential inflexibilities in this paper is demonstrated on a hypothetical educational institute presented in our earlier work [5].

2 Related Work in IS Design to Enable Change

Two classes of research are presented that deal with adaptation of information systems. The first category focuses on the context and changing requirements while the second category addresses architectural reconfiguration and modification.

Souza et al [2] deal with the adaptation challenge from a requirement perspective and propose capturing evolutionary requirements of information systems in order to enable automated or semi-automated adjustments. Zdravkovic et al [10] propose using enterprise models to capture the business context and its variation points to enable runtime adjustment of services in accordance to changes in the capability context.

Researchers in software architecture analysis address change with a particular focus on the effort and process required to enable implementation and modification of a software system to accommodate changes in stakeholder requirements. For example, Bengtsson et al [11] propose a scenario oriented analysis to enable evaluation of alternative software architectures with regards to specified change scenarios. Bohner [12] proposes structural analysis of the software architecture to study the rippling effect of a change, this enabling estimation of effort and time required to implement changes. Building on impact analysis approaches, De Boer et al [3] propose identification of rippling effects of a change in an Archimate model to allow realignment of the technical and business architectures.

Both of the discussed categories enable adjustment of information systems in accordance to changing context, hence focusing on overcoming the first adaptation challenge. However enterprises face emergent needs that arise as a result of interactions of social and technical entities within the organization and its ecosystem [13, 14]. Studies indicate that enterprise capabilities can resist to changing context and implementation of emerging requirements if changes contradict the capability evolution path (the evolution path is shaped by the history of decisions made over its lifetime) [8, 9]. Accommodating such changes requires architectural governance that can identify socio-technical inflexibilities that constitute the second adaptation challenge [13, 15].

3 Uncovering Potential Inflexibilities using Second-order Dependencies in i*

The i* modeling framework is known for its ability to capture intentions of different actors when modeling enterprise requirements. As part of i*, dependencies among actors is modeled to enable analysis regarding how actors rely on one another to satisfy goals and softgoals, acquire resources, and perform tasks. However such dependencies entail commitment among actors and can introduce barriers to change. In this section we propose a method to investigate the most influential elements in i* networks of dependencies to enable identification and analysis regarding highly influential dependencies that can cause inflexibilities.

To determine which dependencies have a higher potential of being barriers to change the degree of coupling among dependencies is (algorithmically) computed using second-order dependencies in i*. The approach enables identification of dependencies that have high impact on the overall network of dependencies. In addition it can enable qualitative and quantitative analysis regarding how a certain change will impact the network of dependencies and enterprise capabilities. An example of qualitative impact analysis on IT and organizational capability alignment is discussed in a related work [5].

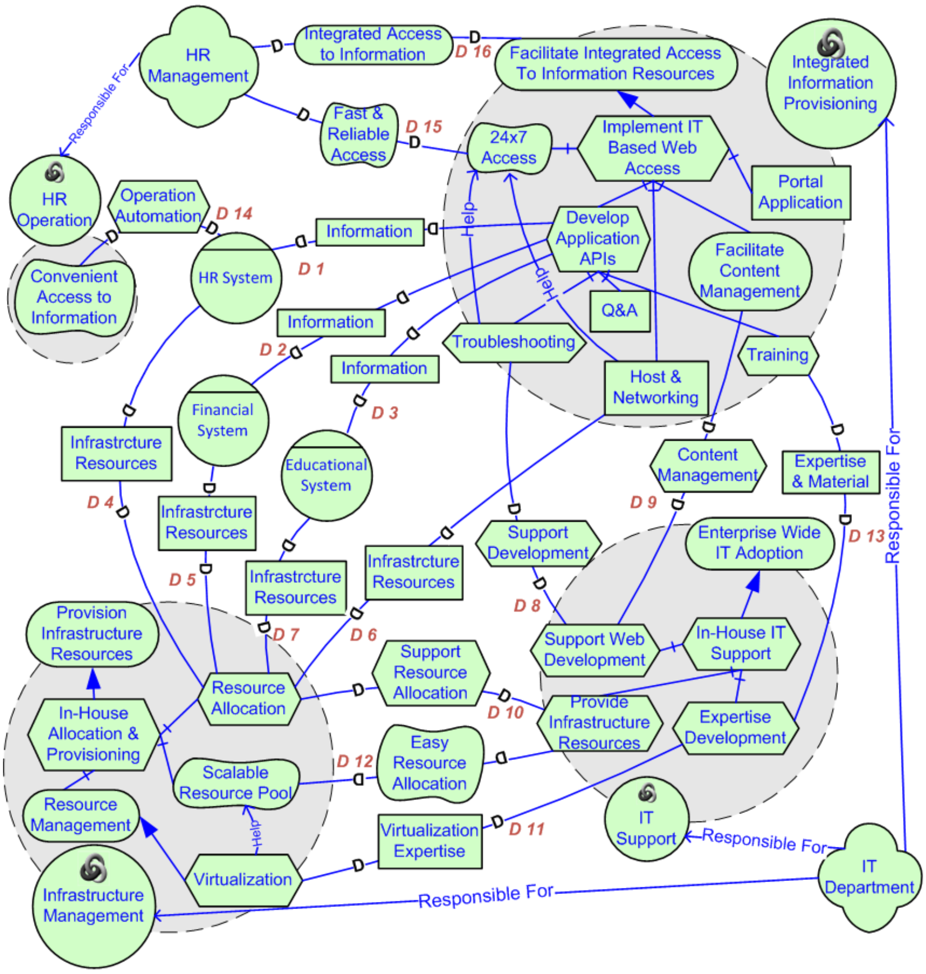

Fig. 1. An i* model of IT Capabilities

In an earlier work [7], an extended version of i* was introduced to enable the modeling of enterprise capabilities and their development, orchestration and deployment alternatives. Using that extension a method for analyzing second-order dependencies is introduced in the context of a hypothetical educational institute. Figure 1 depicts a snapshot of the enterprises IT capabilities and their relations to organizational actors and information systems. A capability is depicted as a specialized type of actor.

A second-order dependency is defined as the reliance of one dependency to another to the extent that it cannot perform with the required quality unless the former dependency is satisfied. In other words second-order dependencies refer to dependencies among (first-order) dependencies.

To extract second-order dependencies one can investigate the strategic rationale model of i*. If the dependee-side element of an i* dependency (element which resides in dependee) such as *D 7* in Fig. 1 (the dependee-side element in *D 7* is *Resource Allocation*), and that element itself is dependent on some other actor as is the case with *Resource Allocation* (which is dependent on the *IT support* capability), then *D 7* is dependent (second-order) on *D 10.*

If the source element of a dependency is comprised of sub-elements where sub-elements are identified through contribution, decomposition and mean-end links (for softgoals, tasks and goals respectively); then a second-order dependency exists from the dependency to each of the dependencies of sub-elements. For example *Virtualization* (a task of *Infrastructure Management* capability) contributes to the dependee-side element of *D 12* (*Scalable Resource Pool*) and depends on *Virtualization Expertise* (resource) which is provided by the *IT Support* (capability), therefore a second-order dependency exists from *D 10* to *D 12*. The second-order dependency exists as *Easy Resource Allocation* (D 12) relies on setup and engineering of the virtualization infrastructure which provides *Scalable Resource Pool*.

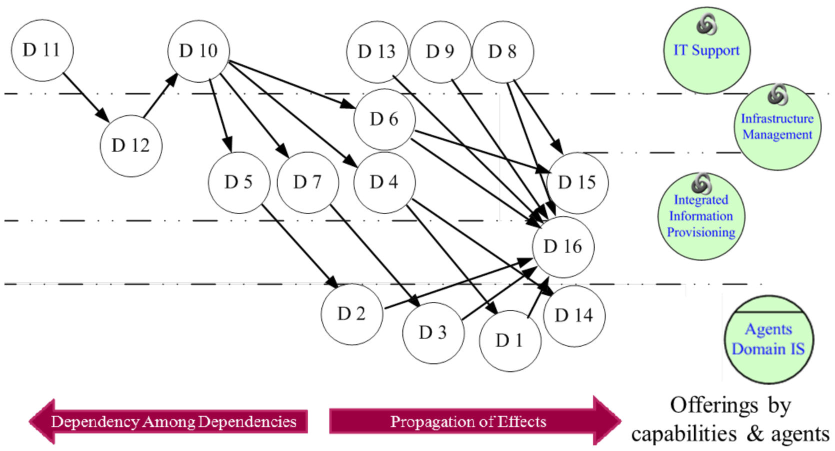

Fig. 2. Dependency Propagation Graph

The dependency propagation graph presented in Figure 2 enables analysis of the rippling effects of dependencies among actors in i*. Its construction can be automated in a tool using the rules described earlier in this paper. The directions of arrows depict the path in which rippling effects of a change can propagate, i.e., the opposite direction of the dependencies in the SR model. In this graph the dependencies are grouped into rows according to the dependee actors. The grouping facilitates visual analysis regarding how changing certain capabilities or systems will impact the overall network of dependencies. With additional information the graph can serve as a roadmap to quantify economical contribution of each dependency and its role in value creation.

According to graph presented in Figure 2, *D 11* which refers to *Virtualization Expertise* provided by the *IT Support* capability to the *Infrastructure Management* capability of Figure 1, is a sensitive element as deficits in resources to design and govern a virtual infrastructure can have extensive impacts on the functionality and use of information systems across the enterprise. Furthermore making changes to the process (i* task) by which this resource is provided, i.e., *Expertise Development* in *IT Support*, can impact many other applications and organizational dependencies. Hence when making decisions regarding its evolution, one should carefully consider consequences and alternatives.

4 Conclusion and Future Work

Building the flexibility required to enable enterprise transformation is a major concern in both management and IS research. While many have proposed approaches to deal with automated adjustment of IS, there is a lack of methods that allow analysis regarding inflexibilities that arise in a socio-technical context. An approach that enables analysis and identification of potential inflexibilities is introduced by investigating second-order dependencies in an i* model of enterprise capabilities.

As future work, in order to fully recognize causes of rigidity, one needs to investigate the degree of impact that a certain sensitive dependency has. This can be achieved through assignment of quantitative measures to the edges of the dependency propagation graph. The measures can be assigned as a weight to depict the importance of the second-order represented by the edges. If the dependency is resulted from a softgoal, the contribution links can serve as a roadmap for assigning values. Such quantitative measures assist human judgement regarding the sensitivity of an element and how it can cause barriers to change.

The results of the analysis can be used at design time to enable accurate planning and mitigation of the risks imposed by any potential inflexibility. In the case presented in this paper, careful planning and consideration in training human resources with the skillsets to manage a virtual infrastructure should be a major concern at the design time. Furthermore the dependency graph can be used at runtime to monitor and measure potential inflexibilities in order to alert the changes in the probability of some dependency causing inflexibility.

Analyzing and interpreting the significance of second-order dependencies without tool support that points to the source i* elements is difficult and reduces the practical usage of the method. Furthermore as the models scale and enterprises grow creation of the graph requires automated tool support that can take an i* model and produce second-order dependencies based on the proposed algorithm.

References

1. Combs, J.G., Ketchen Jr., D.J., Ireland, R.D., Webb, J.W.: The role of resource flexibility in leveraging strategic resources. J. Manage. Stud. ***48***, 1098–1125 (2011)
2. Silva Souza, V.E., Lapouchnian, A., Mylopoulos, J.: (Requirement) Evolution Requirements for Adaptive Systems. In: Proc. of the 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp. 155–164. IEEE (2012)
3. de Boer, F.S., Bosanque, M.M., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., Stafford, J.: Change impact analysis of enterprise architectures. In: Proceedings of the 2005 IEEE International conference on Information reuse and integration (IRI 2005), Las Vegas, Nevada, USA. IEEE Computer Society, Los Alamitos (2005)
4. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., Robinson, W.: The brave new world of design requirements. Information Systems 36, 992–1008 (2011)
5. Danesh, M., Yu, E.: Analyzing IT Flexibility to Enable Dynamic Capabilities. In: Persson, A. and Stirna, J. (eds.) Advanced Information Systems Engineering Workshops. pp. 53–65. Springer International Publishing (2015).
6. Buckl, S., Schweda, C.M.: Classifying Enterprise Architecture Analysis Approaches. In: Poler, R., van Sinderen, M., Sanchis, R. (eds.) IWEI 2009. LNBIP, vol. 38, pp. 66–79. Springer, Heidelberg (2009)
7. Danesh, M.H., Yu, E.: Modeling enterprise capabilities with i*: reasoning on alternatives. In: Iliadis, L., Papazoglou, M., Pohl, K. (eds.) CAiSE Workshops 2014. LNBIP, vol. 178, pp. 112–123. Springer, Heidelberg (2014)
8. Teece, D.J., Pisano, G., Shuen, A.: Dynamic capability and strategic management. Strateg. Manag. J. ***18***, 509–533 (1997)
9. Leonard-Barton, D.: Core capabilities and core rigidities: a paradox in managing new product development. Strateg. Manag. J. 13, 111–125 (1992)
10. Zdravkovic, J., Stirna, J., Henkel, M., Grabis, J.: Modeling business capabilities and context dependent delivery by cloud services. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 369–383. Springer, Heidelberg (2013)
11. Bengtsson, P., Lassing, N., Bosch, J., van Vliet, H.: Architecture-level modifiability analysis (ALMA). J. Syst. Softw. ***69***, 129–147 (2004)
12. Bohner, S.A.: Software change impacts-an evolving perspective. In: Proceeding of International conference on Software Maintenance, (ICSM 2002), pp. 263–272 (2002)
13. Dreyfus, D., Iyer, B.: Managing architectural emergence: a conceptual model and simulation. Decis. Support Syst. 46, 115–127 (2008)
14. Furukawa, M., Minami, A.: A Study on the "Flexibility" of Information Systems (Part 1): Why Do They Need to Be Flexible? Int. J. Bus. Manag. 8, p48 (2013)
15. Sirmon, D.G., Gove, S., Hitt, M.A.: Resource Management in Dyadic Competitive Rivalry: The Effects of Resource Bundling and Deployment. Acad. Manage. J. 51, 919–935 (2008)