



Analyzing Second-Order Dependencies in i^*

8th International i^* Workshop

Mohammad Hossein Danesh
Eric Yu

University of Toronto



Agenda

1. Introduction & related work
2. Analyzing socio-technical inflexibilities
3. Uncovering potential inflexibilities using
second order dependencies
4. Illustrative example
5. Discussion & conclusion



1. Introduction

- ▶ Dealing with change is crucial for both IT & Business
 - ▶ The intertwined nature of IT & Business calls for
 - ▶ Constant alignment and realignment
 - ▶ Related Work in dealing with Change in IT
 - ▶ Enterprise & Requirement modeling
 - ▶ Focus Context & evolutionary requirement description
 - ▶ Enable automated/semi-automated adjustment of software & services
 - ▶ Software & Enterprise architecture
 - ▶ Focus on effort needed and process of implementing changes
 - ▶ Apply scenario oriented or structural analysis to estimate time & effort
- (Souza et al, 2012)
(Zdravkovic et al, 2013)
- (Bengtsson et al, 2004)
(Bohner, 2002)
(De Boer, et al 2005)



2. Need to Address Socio-technical Inflexibilities

- ▶ Dealing with change has two dimensions
 - ▶ Ability to identify changing context and adjustment of software & Services
 - ▶ Addressed by the discussed related work
 - ▶ Flexibility of enterprise capabilities and organization setting to accommodate change, create new services & software and support their deployment

- ▶ Inflexibilities can arise as a result of

- ▶ Social and technical dependencies that exists within an enterprise

(Dreyfus & Iyer, 2008)

(Furukawa & Minami, 2013)

- ▶ Traveled path and commitments of enterprise capabilities

(Leonard-Barton, 1992)

(Teece et al, 1997)



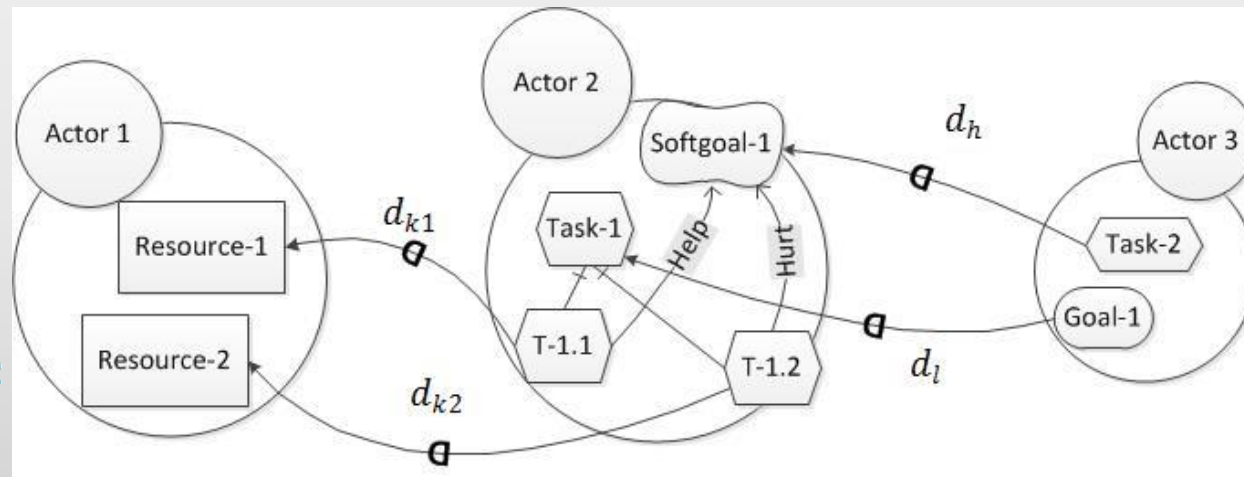
3. Second Order Dependencies

- ▶ Second Order Dependency is defined as
 - ▶ 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 dependencies among (first-order) dependencies
- ▶ Extracted from SR model

d_h is dependent on d_k

d_h is dependent on d_{k-1} and d_{k-2}

d_l is dependent on d_{k-1} and d_{k-2}





3. Second Order Dependencies Extraction Rules

▶ Extracted from SR model using

▶ Rule 1:

- ▶ If the dependee-side element of an i^* dependency (A) is dependent on
 - some other actor, i.e., dependency (B)
- ▶ Then a second order dependency exists from
 - A to B, i.e., A is dependent on B to be satisfied

▶ Rule 2:

- ▶ If dependee-side element is comprised of sub-elements
 - Sub-elements identified through
 - Contribution
 - Decomposition
 - Mean-end links
- ▶ Then a second-order dependency exists from
 - The dependency to each of the dependencies of sub-elements

4. Current State

- ▶ What are the Capabilities
- ▶ What are their
 - ▶ Intents
 - ▶ Routines
 - ▶ Resources
- ▶ How & Why do Capabilities depend on one another

Three Main IS

- ▶ Modeled as i* Agents
- ▶ Managed by Partners
- ▶ Hosted in-house

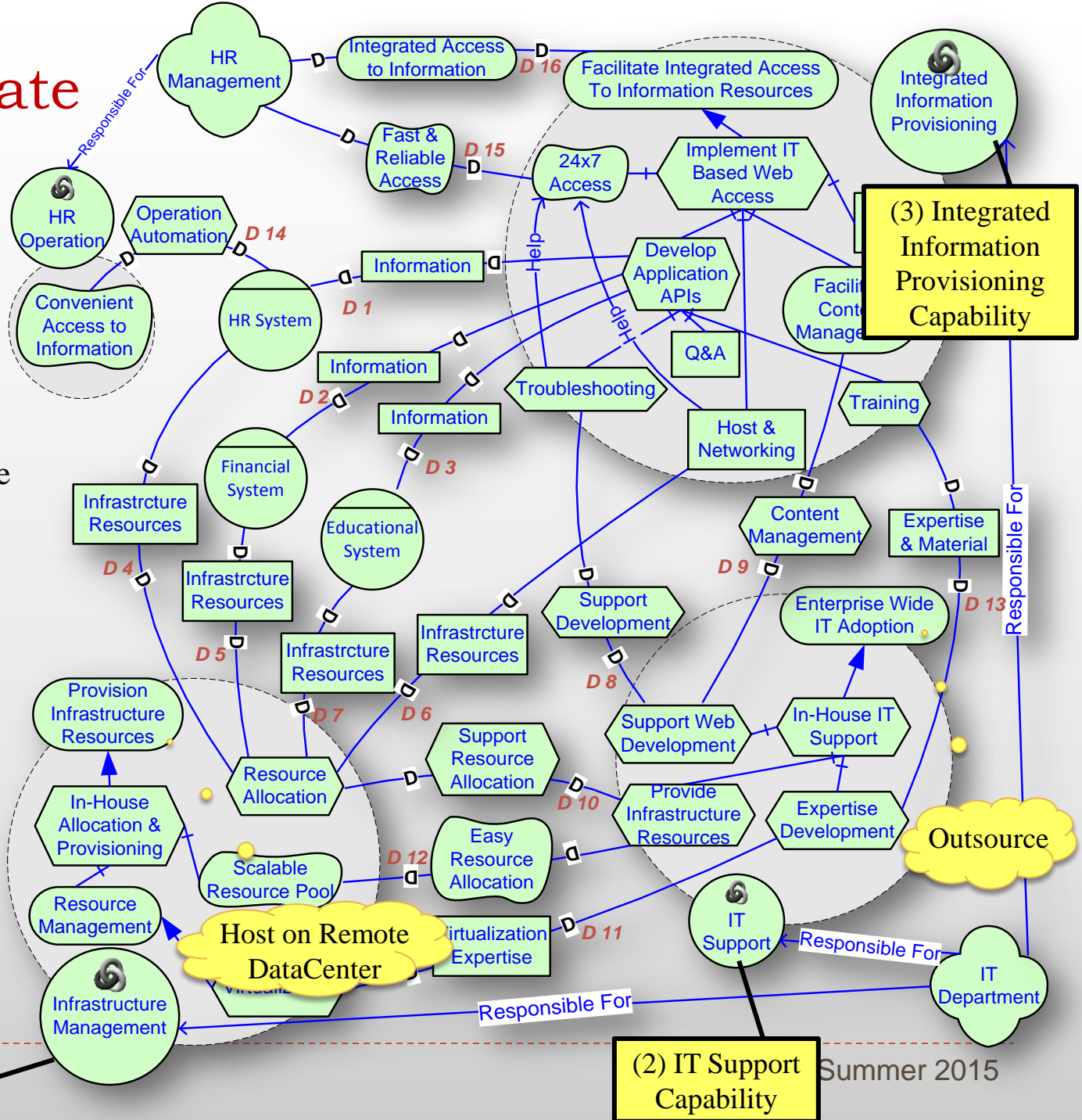
HR Dependencies

- ▶ Others have similar relations

Hosting in House (1)

Support & Maintain (2)

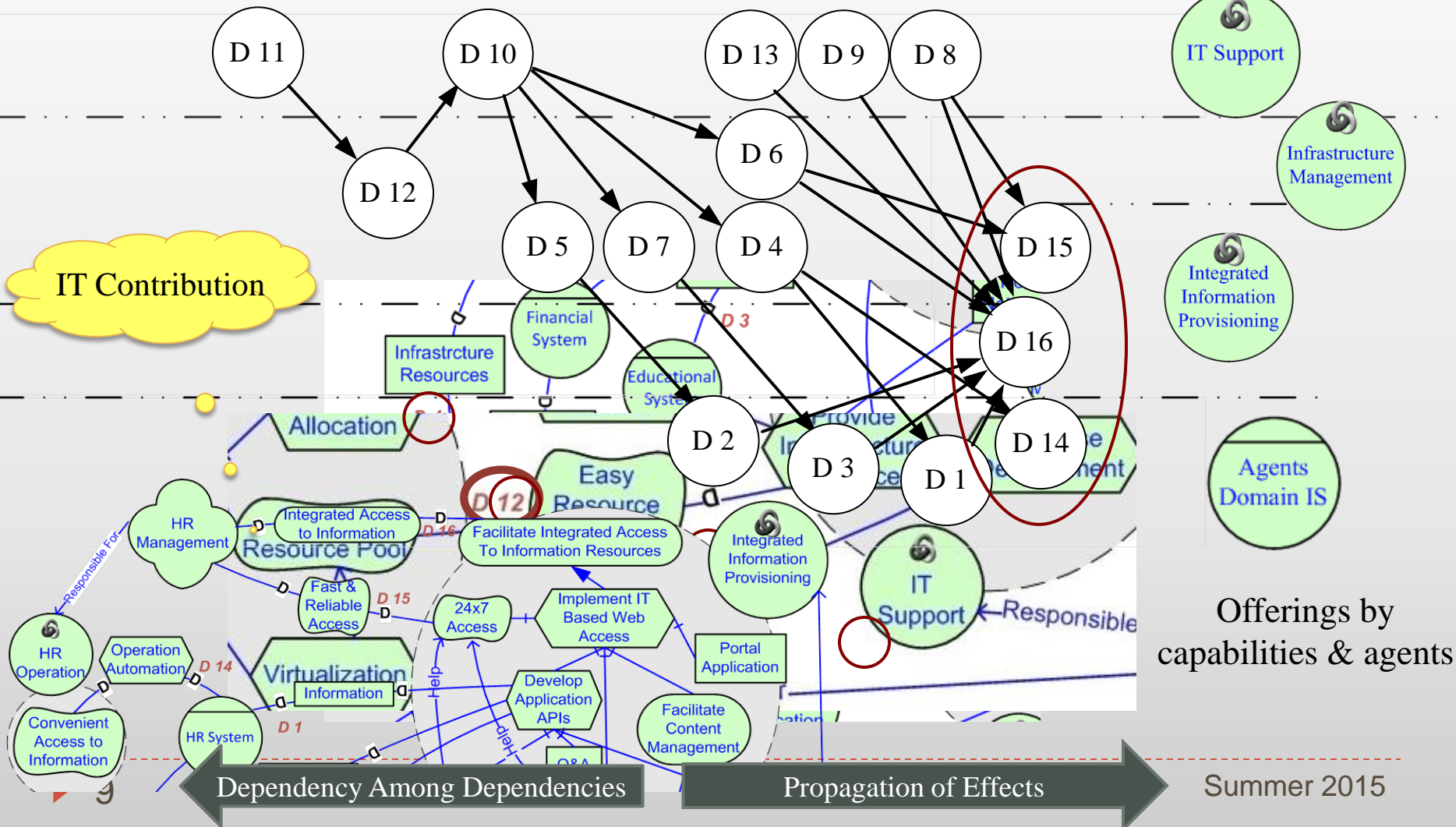
SSO & Access (3)





4. Use a Dependency Propagation Graph to Depict Capability Offerings & Contribution

Virtualization Expertise



5. Summary

- ▶ Lack methods to analyze inflexibilities in
 - ▶ Socio-technical context
- ▶ Use second order dependencies to
 - ▶ Identify potential inflexibilities
- ▶ The analysis can be used
 - ▶ at design-time to
 - ▶ Plan and Mitigate the risk imposed by inflexibilities
 - ▶ at run-time to
 - ▶ Monitor & Measure activities that can produce inflexibilities



5. Future Work

- ▶ Tool Support is essential for the analysis
- ▶ Quantification methodology & guidance is needed
- ▶ The current analysis cannot depict
 - ▶ Influences of different Flexibility requirements on one another
 - ▶ Cause & Effect of different Quality Attributes



THANK YOU

