A thermometer for interdependence: Exploring patterns of interdependence using networks of affordances

Research-in-Progress

Brian T. Pentland
Michigan State University
East Lansing, MI 48824 USA
pentland@broad.msu.edu

Jan Recker
Queensland University of Technology
Brisbane, QLD 4001 Australia
j.recker@qut.edu.au

George Wyner
Boston College
Chestnut Hill, MA 02467 USA
george.wyner@bc.edu

Abstract

Interdependence is a central concept in systems and organizations, yet our methods for measuring it are not well developed. Here, we report on a novel method for transforming digital trace data into networks of events that can be used to visualize and measure interdependence. The edges in the network represent sequential flow and the vertices represent actors, actions and artifacts. As with conventional approaches such as process mining, our method uses input from a stream of time-stamped occurrences, but the representation is simpler and more appropriate for exploration and theory building. As digital trace data becomes more widely available, this method may become more useful in information systems research and practice. Like a thermometer, it helps us measure a basic property of a system that would otherwise be difficult to see. We outline conceptual and methodological contributions, and discuss ways in which our work can be extended.

Keywords: interdependence, affordance network, organizational routines, process mining, narrative network, exploratory data analysis.

Introduction

Interdependence is a central issue in systems and organizations (Orlikowski 1992) and a defining feature of organizational routines (Feldman and Pentland 2003). Interdependence describes how actions, actors and technology in routines are intertwined in mutually dependent ensembles (Orlikowski and Scott 2008). Interdependence is like air; it surrounds everything in an organization, but it is difficult to see. Interdependence has been theorized to exist between organizational subunits (Thompson 1967), jobs (Kiggundu 1981), actions (Polyvyanyy et al. 2015), technologies (Leonardi 2011), but it is difficult to observe or measure (Wyner 2011).

We aim to develop a method for observation and measurement of interdependence. Just as a thermometer, together with other meteorological instruments, allows us to record observations of the air
and develop sophisticated models of its behavior, we believe that it is important to develop new techniques for measuring and visualizing interdependence.

In this paper, therefore, we report research-in-progress on a tool for visualizing and measuring interdependence in what we call a network of affordances. So far, we have devised an algorithm for taking a stream of sequential, time-stamped occurrences and converting them into a network of interdependent events, much in the same way that existing technologies such as process mining (van der Aalst 2011b) operate. We have implemented this algorithm in a prototype software artifact, and we have examined data from an illustrative case of organizational routines.

Our contribution is both conceptual and methodological. Conceptually, we make two key moves. First, we describe events in terms of three dimensions: actors, actions, and artifacts. Together, these three ingredients make up the minimum required to describe an affordance (Chemero 2003). Affordances describe action possibilities that exist between an actor and a technological object, and they allow evaluating which action possibilities were actualized in routines. Thus, affordances allow description of what someone did with something. This is an important extension to other event networks, which typically focus on actors or artifacts but not the relationships between them (Pentland and Feldman 2008).

Second, we identify the sequential relationships between affordances as they are used in practice and construct a network based on these sequential relations. We use the term “affordance network” to describe this representation. This representation is similar to other event- (e.g., van der Aalst et al. 2004) or activity-based network models (e.g., Pentland et al. 2012), but it extends them by explicitly including actors, actions, as well as artifacts, all of which contribute to interdependence. This move also extends conceptualizations of affordances from a focus on technological objects themselves and the identification of one or more affordances (Leonardi 2011; Seidel et al. 2013) to the analysis of the sequences of events in which they are actualized. Importantly, this extension allows us to examine the interaction of actors and technological artifacts in a sociomaterial ensemble (Orlikowski 2007; Leonardi and Barley 2008), contributing a more nuanced view of the role of technology in routines.

Methodologically, we offer a novel way to construct this kind of network from a stream of time-stamped occurrences. Traditionally, time-stamped occurrences are analyzed using process mining techniques, which construct state-transition graphs such as Petri nets (van der Aalst 1998). This analysis usually considers timestamps and activity events but largely ignores actors and artifacts let alone the relationships between them. Our algorithm explicitly includes these latter elements. We describe the algorithm, which is implemented in a prototype software artifact written in MatLab. We are starting to see event-based networks used in information systems research (e.g., Goh et al. 2011). The method described here provides a rigorous, consistent foundation for building on that work.

We proceed as follows: Next we will describe the motivation for our research and position our affordance network within related work on interdependencies and approaches to modeling or analyzing organizational routines, and networks. Then we will provide a formal definition of an affordance network and briefly describe our implementation. We then present an illustration using data from routines observed in a US call center (citation withheld). We conclude by reviewing expected contributions, limitations and future work.

**Related Work**

**Interdependence and its measurement**

In theory, interdependence occurs between many kinds of entities, such as tasks (Arthur Jr. et al. 2005), actors (including roles, jobs, and organizational units, Thompson 1967) and technologies (Bailey et al. 2010). In practice, we see manifestations of interdependence primarily when there are breakdowns (Wyner 2011). When a process is running smoothly, it can be difficult to tell how the activities are related. When interdependencies become manifest, however, workarounds and other escalations are often required (Alter 2014). Managing interdependence is considered a critical aspect of organizational design in general (Thompson 1967) and information systems design in particular (Malone et al. 1999).
Networks of affordances

In research, interdependence is traditionally measured using perceptual survey measures (e.g., Campion et al. 1993; Arthur Jr. et al. 2005). The approach we describe here advances on this by using objective data (time-stamped digital “trace data”) to infer sequential relationships between activities, actors and technological artifacts. We focus on sequential relationships for two reasons. First, sequential flow is a core issue in the design of information systems (Dumas et al. 2005). Second, Thompson (1967) described interdependence as a “Guttman scale”, where sequential flow was the foundation. Higher levels of interdependence (pooled and reciprocal) are built on underlying flows. Subsequent frameworks (e.g., Malone et al. 1999) use the same basic idea: flow is the basic phenomenon. Interdependence can also be defined in terms of the economic payoff (Puranam et al. 2012), but our approach does not address that form of interdependence.

Approaches to modeling and analyzing organizational routines and processes

Our approach sets out to visualize and measure patterns of interdependence in routines of actors, actions and artifacts. Similar ambitions exist also in classical approaches to organizational analysis (Knights and Willmott 2010), most notably in approaches to business process management (vom Brocke and Rosemann 2010).

Process modeling is an approach often used by analysts to document and analyze current organizational operations, because these models help business personnel understand the work domain and identify improvement opportunities related to the routines and the involved information systems (Dean et al. 2001). Process models are a type of conceptual model, that is, they provide a typically graphical representation of some features of a real-world domain (Burton-Jones and Weber 2014) – the organizational routine the analyst is interested in. Process models are usually employed as part of an effort to analyze current operations (‘as is’ modeling), or as part of an effort to design improved blueprints for future operations (‘to be’ modeling). In either case, process models typically include graphical depictions of at least the process steps, agents, actors, roles and (sometimes) artifacts that are involved in a business process (Curtis et al. 1992). Importantly, process models are abstractions in that they describe largely the common way of how a routine should be enacted – to the point that escalations, workarounds or variations (that is, interdependencies) are deliberately excluded (Sharp and McDermott 2009). Also, in practice, any actor may choose to follow the modeled procedure but may also interpret and enact the procedure differently (Lee et al. 2008).

Process mining describes a technology-based methodology for process analysis that constructs Petri nets using data from event logs (van der Aalst et al. 2004). Because they are able to model concurrency and system states, Petri nets provide a state-based framework for process representation (van der Aalst 1998). Current process mining research emphasizes the discovery of accurate models to allow for compliance checking and deviation analysis (van der Aalst 2011b). Given appropriate data, it is possible to recover an accurate, detailed model for a digitized process at any given point in time. A common limitation of this approach, however, is that the mined models provide excruciatingly accurate, detailed information (so-called spaghetti models, see van der Aalst 2011a) such that pattern recognition or other analysis is often not possible without further transformation.

To explore and visualize patterns of interdependence, we argue that we need a model that does not abstract away variations as is typically the case with process models, and which allows views that do not always foreground the sequence of actions as is the case with both process models and process mining. Where actions are temporal in nature and thus evanescent, actors and artifacts are easier to observe and follow over time. By visualizing an organizational process as a network of affordances we hope to make visible aspects of interdependence that are more dependent on the relationships among actors and artifacts as well as actions.

Network models

We will define a network model, of which several kinds already exist. The most common are social networks (Borgatti et al. 2013), where the vertices represent individuals and the edges represent relations between the people, such as communications (Kossinets and Watts 2006), distance (Backstrom et al. 2010) or friendship ties (Ellison et al. 2007). Recently, scholars have extended the social network to include non-human entities (Kane and Alavi 2008), but the basic framework remains the same.
Of course, network models are widely applied to technical artifacts as well (Barabási and Albert 1999). Like social networks, the vertices represent discrete entities, such as web pages. So-called “actor-networks” (Latour 1987) can also be conceptualized as networks of artifacts, although research in this tradition does not typically employ mathematical network methods or models.

The most closely related work involves networks where the vertices represent actions or activities, rather than people or things. For example, Eppinger’s (1991) design structure matrix is a network of activities or processes. Pentland and Feldman (2007) use the phrase “narrative network” to describe the same basic idea: a network where the vertices represent activities or events. Such event-based networks are becoming increasingly common in empirical research in information systems. For example, Goh et al. (2011) use narrative networks to identify where and how health information technology influences patterns of work. Yeow and Faraj (2011) use narrative networks to compare patterns of action generated by invoice processing systems. In research on healthcare information systems, Hayes et al. (2011) use action networks to study the impact of new technologies and potential needs for additional training at a medical center. They have also been applied as a tool for detecting change in processes (Pentland et al. 2014).

In summary, prior methods tend to highlight actions, people or technology, but not all three. Recent interest in socio-materiality (Orlikowski and Scott 2000; Leonard 2013) has shown that these dimensions are tightly inter-related, but existing models generally pick out one or two dimensions, at most. Here, we introduce an approach that incorporates all three dimensions.

Definitions and Notation

We define an affordance network as a class of event network where the actor, action and artifacts are all available to identify the vertices of the network. The basic notation for event networks is summarized in Table 1 and explained below.

On the basis of the event network notation, an affordance network can be constructed whenever the stream of occurrences contains the three basic dimensions: actor, action, and artifact. Thus, affordance networks are a subclass of event networks. It should be clear to readers who are familiar with the nuanced models of affordances (Chemero 2003) that we are using a simplified definition here. For instance, we do not examine affordances that exist as action possibilities but only those that are in fact actualized (e.g., Bernhard et al. 2013; Strong et al. 2014), that is, enacted by actors in their routines. Also, rather than focusing on relations within a particular affordance, we are using digital trace data to examine sequential relations between affordances as they are actualized in a given setting. These relations provide an indication of the pattern of interdependence in the setting. Thus, we do not consider potential or hidden affordances, only actualized ones, and how these actualizations are sequentially related in the data.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occurrences ($\Omega$)</td>
<td>Each occurrence, $\omega_i$, is an instantaneous observation of the form $(t_i, y_i)$, where $t_i$ is the time of the observation and $y_i$ is the set of values observed for a set of attributes $A = { a_{i1}, ..., a_{ip} }$.</td>
<td>$\Omega = { \omega_1, ..., \omega_m }$</td>
</tr>
<tr>
<td>Time ($t$)</td>
<td>Timestamp on each occurrence</td>
<td>$t_1 \leq t_2 \leq ... \leq t_m$</td>
</tr>
<tr>
<td>Threads ($\Theta$)</td>
<td>Each thread, $\theta_i$, is a set of occurrences, $\theta_i = { \omega_1, ..., \omega_r }$ which all pertain to a single coherent flow of action.</td>
<td>$\Theta = { \theta_1, ..., \theta_k }$</td>
</tr>
<tr>
<td>Events ($V$)</td>
<td>Each event, $v_i$, is a set of occurrences that have the same observed values for a given set of attributes $B \subseteq A$. Events are vertices in a valued, directed graph, $G$.</td>
<td>$V = { v_1, ..., v_n }$</td>
</tr>
<tr>
<td>Edges ($E$)</td>
<td>Each edge, $e_j$ is indicated by the presence of one or more threads that extend between two events.</td>
<td>$E = { e_{i1}, ..., e_{is} }$</td>
</tr>
<tr>
<td>Event network ($G$)</td>
<td>A valued, directed graph, $G = (V, E)$, comprising $G = (V, E)$</td>
<td></td>
</tr>
</tbody>
</table>


### Networks of affordances

<table>
<thead>
<tr>
<th>a set $V$ of events together with a set $E$ of edges</th>
</tr>
</thead>
</table>

**Table 1: Essential concepts, definitions and notation**

**Occurrences.** We define an occurrence-based data set as consisting of a set of occurrences, $\Omega = \{ \omega_1, \ldots, \omega_m \}$. The index $i$ indicates the $i_{th}$ occurrence in the stream of data. The occurrences can be sequentially ordered by time such that $t_i \leq t_{i+1}$, so that $\omega_i$ precedes $\omega_{i+1}$.

Occurrences are treated as instantaneous observations. Each occurrence has a timestamp and may have other data that records what occurred at that moment. Thus, each occurrence has a set of attributes such as the time, location, and type of the occurrence. Attributes may also include actors, artifacts, text, or anything else. Each occurrence $\omega_i$ takes the form $(t_i, y_i)$, where $t_i$ is the time of the occurrence and $y_i$ is a function from $A$, the set of attributes being observed, to $O$ the set of possible values for those attributes.

Thus $y_i(a)$ is the value observed for attribute $a$ in occurrence $\omega_i$.

In practice, an occurrence-based data set can be stored in a simple spreadsheet, where each row is an occurrence. One column includes the timestamp and the other columns include the other attributes. For example, when actors are included in the attributes, it is possible to infer a social network (O’Madadhain et al 2005). When actors, actions, and artifacts are included, it is possible to construct a network of affordances.

**Threads.** Occurrences are often observed and recorded as coherent threads. A thread is a set of time-ordered occurrences that all pertain to a single coherent flow of action. Thus, any given thread, $\theta_i$, is a subset of the overall set of occurrences: $\theta_i \subseteq \Omega$. Within any thread, occurrences can be sequentially ordered. The thread is a central concept in this framework because it is used to define the edges of the event network. It is an indicator of what Abbott (1992) referred to as “colligation”: the logical relationship of events.

**Events.** Following Abbott (1992) and related literature, we distinguish between occurrences and events. Events are defined as the subset of occurrences $(v_i \subseteq \Omega)$ that have the same observed value for an attribute or set of attributes. For a set of attributes $B \subseteq A$, we can partition $\Omega$ into a set of events $v_i$ over $B$ such that every $\omega_i$ in $\Omega$ is in exactly one such $v_i$, and for any $\omega_i$ and $\omega_j$ which are both in $v_i$, $y_i[B] = y_j[B]$.

Occurrences are instantaneous, but events have duration, which can be computed as the difference in timestamp between the first and last occurrence in the event. While the distinction between occurrences and events may seem non-intuitive at first, it is an essential step in any rigorous analysis of sequential, historical data (Abbott 1992) or indeed processes (van der Aalst 2011b).

**Event network.** The event network is a valued, directed graph. In an event network, the set of events, $V$, form the vertices. The set of edges, $E$, is formed by following the threads. An edge $e_i$ exists between events $v_i$ and $v_j$ whenever there is a thread containing occurrences $\omega_i$ and $\omega_j$ that are sequential within the thread and where $\omega_i$ is in $v_i$ and $\omega_j$ is in $v_j$. Edges can be interpreted as indicators of sequential interdependence between two entities.

In an event network, the vertices represent events, and the edges represent the sequential interdependence between those events. Like a social network, an event network represents relationship between pairs of actions. It is important to emphasize that event networks are not multimodal (e.g., actors × event); they are unimodal (event × event). Because it only represents pairs of events, an event network cannot represent concurrency or higher-order sequences. Further, unlike a Petri net, it does not attempt to model the state of the underlying system.

**Software Artifact and Illustration**

Using MatLab, we have implemented an algorithm for converting digital trace data into an event network, as defined in Table 1. It computes the network graph and displays it using the MatLab Biograph viewer. It also exports an XML file in GEXF format, which contains lists of vertices, edges, and their attributes. This allows for statistical analysis and further processing. Details of the algorithm are omitted to conserve space but are available from the lead author upon request.

Please do not circulate or cite without permission. 5
**Illustration: Problem resolution at a financial services call center**

To illustrate the concept of an affordance network, we revisit data that was previously published (citation withheld), but not analyzed this way. The data are from a call center in the US. The work we analyze, in simple terms, is that of call center agents responsible for resolving inquiries and problems of customers that cannot be resolved by branch office personnel or the regular call-center staff. The resolution processes include different investigations, work assignments and interactions depending on the type of problem encountered.

For purposes of illustration, we compare two types of problems carried out by two different, specialized sub-units within the call center. The original data contained 20 instances of problem A (with 4.35 occurrences per problem instance) and 29 instances of problem B (with 6.06 occurrences per problem instance). Actors and actions were coded as integers, while the artifacts were described using abbreviated names (e.g., “Word”). Figure 1 contains four different views of the same data for each type of problem. In these figures, we are concerned with the overall structure, not the labels on particular vertices. Plus, if the vertices were legible, they could reveal the identity of the research site, so we have deliberately produced them in tiny-print.

The first row in Figure 1 shows the network that results from defining vertices by the types of actions. This view could be called an “action network” (Pentland et al. 2011), and is closest to a traditional process model, either created or mined (van der Aalst 2011a). Problem B involves more kinds of actions, so it would have higher “component complexity” (Wood 1986). Problem A is mostly sequential, while problem B shows a larger number of reciprocal relations between actions (indicated by ties that go both ways), but the two patterns do not obviously look different in kind. Both have lots of branches and a few loops. This view indicates hand-offs between process steps, which is one dimension of interdependence.

The second row in Figure 1 shows the network that results from defining vertices by the individual actors. This view can be interpreted as a social network. In this view, the two types of problems are strikingly different. Problem A looks like a classic example of a “pooled” interdependence (Thompson 1967). One actor initiates every problem, assigns it to someone to resolve, and one actor closes every problem. Other types of problems in this work unit exhibit a similar pattern when we view sequential relations between actors. Problem B, however, looks quite different. Work is passed back and forth, with reciprocal interdependence between some members of the unit.
When viewed as a network of artifacts (third row in Figure 1), we note that Problem A involves fewer technologies (5) than Problem B (9). In both types of problems, there is a great deal of reciprocal interdependence between the systems used to carry out the work. Work is frequently passed back and forth between the systems. This suggests that for problem B, more action possibilities afforded by the technological systems in place are actualized, but this view fails to disclose how and why this is the case. In other words, the interdependences that exist between actor, action and a potentially useful technology artifact remain hidden in this network. It does not convey the sociomaterial ensemble of the role of technology in the routine (Orlikowski 2007; Leonardi and Barley 2008); and neither do the previous two network models.

Our affordance network (last row in Figure 1) visualizes this interdependence. By increasing the granularity of the description to include the (actor, artifact, action) tuple, our “thermometer” reflects interdependence across all three dimensions. In doing so, we find that the pattern of interdependence can be reduced almost entirely (as theorized by Thompson 1967) to sequential flow. Reciprocal relations between actors or artifacts become more clearly articulated as sequences of actualized affordances. Reciprocal relations among affordances remain, but are relatively few in number, especially in Problem A. We are still learning how to interpret and compare these alternative perspectives, but visualizing the different kinds and formats of interdependence seems like an essential first step.

**Figure 1: Comparing the interdependence structure of two problems**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Problem A ( (n = 20) )</th>
<th>Problem B ( (n=29) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions</td>
<td>11 actions</td>
<td>18 actions</td>
</tr>
<tr>
<td>Actors</td>
<td>14 actors</td>
<td>12 actors</td>
</tr>
<tr>
<td>Artifacts</td>
<td>5 artifacts</td>
<td>9 artifacts</td>
</tr>
<tr>
<td>A-A-A</td>
<td>33 actualized affordances</td>
<td>62 actualized affordances</td>
</tr>
</tbody>
</table>

Please do not circulate or cite without permission.
Summary and On-going Work: Using Affordance Networks for Theory Building

So far, we have focused on constructing an affordance network viewer and exploring its application to data sets of occurrences of actions, actors and artifacts. We view this as the generative, theory-building phase of our research. As we continue our experiments with this “interdependence thermometer,” a key issue will be how to interpret the results. We envision at least two key advantages. First, since we use digital trace data, the method should be less prone to subjective bias than existing methods based on perceptual survey measures. We intend to explore new visualizations and metrics, so that the relative contribution of each dimension can be seen. Second, it allows us to measure interdependence among any set of entities, not just between pairs of entities. By counting edges that cross boundaries between regions of the graph, we can measure interdependence between heterogeneous entities. Thus, we hope to extend the rigor and applicability of this central concept.

The key expected contributions of our research-in-progress are methodological and conceptual. Research on socio-technical systems in general and information systems in particular has long been concerned with incorporating actors and artifacts on an equal footing. The affordance network accomplishes this in a simple, intuitive way. We are just beginning to learn how to interpret and analyze these networks, but our “interdependence thermometer” seems likely to facilitate a variety of methodological contributions:

1) Multiple views: examining data about actors and actions as well as artifacts in one model;
2) Nesting: Extending models of single or shared affordances to flows of nested actualized affordances, that is, affordances in sequential combination (Ye et al. 2009);
3) Forensic analysis: tracing evolutionary changes to business processes over time through changes to actors, actions and/or artifacts.

Conceptually, the technique reported here provides a number of promising theory building opportunities.

1) Computing an index of task complexity for tasks carried out by multiple actors (human and otherwise). Haerem et al. (forthcoming) have proposed a conceptual approach based on counting pathways in an affordance network. The technique described here makes it possible to construct such a network from digital trace data;
2) Computing an index of interdependence (e.g., based on density or centrality). This index could allow theorizing and evaluating complexity of routines and predicting routine changes through introduction of new actions or new artifacts;
3) Understanding dynamic relationships between the dimensions (social, technical and action). This will especially contribute to affordance theory, by being able to visualize and measure the relational nature of affordances and their actualization over the course of a routine.

Limitations of our research are as follows. Conceptually, our model of an affordance as the tuple (actor, artifact, action) is restricted to an actualized affordance. Thereby, it does not capture affordances that were (a) perceived but not actualized or (b) not perceived in the first place (Chemero 2003). Methodologically, the design of our network viewer is at present restricted to reading and visualizing data. Computations of applicable network statistics such as Markov variety, centrality, or entropy are not yet included but will be added. Empirically, we have so far considered data from one set of routines. The case we consider has fairly well-defined tasks within an organizational container (Winter et al. 2014). Different types of routines exist, from highly formalized processes such as invoice processing (Pentland et al. 2010) to highly generative and unstructured routines such as behaviors on online social networks (Kane et al. 2014). It will be interesting to compare interdependencies in affordance networks across cases of formalized versus generative routines or within versus across organizations.

Acknowledgements

We wish to thank Yu Lucy Han for writing the prototype software.
References


Leonardi, P.M. "Theoretical Foundations for the Study of Sociomateriality", Information and Organization (23:2) 2013, pp. 59-76.


