ThreadNet:
Tracing and visualizing associations between actions

Brian T. Pentland, Michigan State University
email: pentlan2msu.edu
IFIP Working Group 8.2, Dublin, December 2016

Abstract. Patterns of action are central to processes, practices, routines, and organizations, yet they are difficult to see and interpret. This paper reports research in progress on a software tool for converting narrative fragments (referred to here as *threads*) into networks of action by tracing associations between actions. This approach makes it possible to visualize action patterns. The goal of the research is to facilitate the interpretation of action patterns.

Keywords. Networks of action; Narrative networks; Exploratory data analysis.

1 Introduction

Processes, routines, and other patterns of action are hard to see. Even if you are looking right at the action, time is passing. Our narrow window of perception is framed in time and space, so we see one action at a time. The rest of the pattern, if there is one, can only be reconstructed. Even then, we reconstruct a single thread of action, from a single point of view. What happened the *last* time? What will happen the *next* time? What did *other people* see? If we knew, it would surely inform our interpretation of what seems to be happening *this* time.

This paper reports on ThreadNet, a software tool that is intended to help us see and interpret patterns of action. I will describe and show what ThreadNet does, but this research-in-progress is directed at the general problem of observing and interpreting *patterns of action.* Over the years, scholars in various disciplines have become quite good at interpreting words, stories, and all manner of symbols and signs. But patterns of action pose an interesting challenge: how to interpret something that we cannot readily see?

2 Motivation for ThreadNet

The basic idea for ThreadNet started as a way to address the problem of how to observe an organizational routine (Pentland and Feldman, 2007). In our fieldwork, Martha Feldman and I had the same frustration: our field notes contained fragments of routines, not whole routines. No matter where one sits, stands, or walks around, the best one can get is a partial picture. In some situations, archival records can solve the problem of going start-to-finish, but they miss a lot of important action along the way.

We recognized this as a general problem: how could we stitch together narrative fragments to create an overall picture of the pattern?

Martha and I were not the first to join a narrative perspective with a network metaphor. Barbara Czarniawska's (1997) book, *Narrating the Organization*, suggests that action nets might be a better unit of analysis that organizations. I continue to find this idea inspirational, but unobservable entities are hard to analyze.

3 ThreadNet: From narrative threads to narrative networks

ThreadNet attempts to address this problem by taking narrative fragments and summarizing them to create a picture of the overall pattern. I have started to refer to these narrative fragments as *threads*.

3.1 Threads have coherence

Actions do not happen in isolation -- they occur as part of streams of activity. Musicians rarely just play one note; more often, they play tunes. To a greater or lesser extent, these streams of activity are coherent. By *coherent*, I mean that there is a common element that connects one action or event to the next. Abbott (1992) refers to this as colligation. Pentland and Yoo (2016) argue that coherence arises from material, logical, cultural, and institutional sources. They explain that coherence is a key part of narrative reasoning (Bruner, 1986), sense-making (Weick, 1995) and accounting for our actions (Lyman and Scott, 1968; Orbuch, 1997). They go on to claim: "Without assuming coherence, people could not function in the world, and neither could our algorithms. Human and material agencies depend on and exploit coherence in their on-going efforts to get things done."

3.2 Action networks are woven from threads

By tracing the actions in a collection of threads, we can create a network. This is what ThreadNet does. In this process, each vertex of the network is observed in the data; the vertices represent stuff that happened in the threads. Each edge in the network is observed in the data; the edges represent what happens next, as we trace from one action to the next along the thread. The vertices and edges of the network are the most basic ingredients of narrative: the sequential relation of events in time.

While each vertex and edge is observed, the network as whole can never be observed, so it is manufactured by the ThreadNet algorithm. In the same way that nobody has ever seen a whole routine, nobody has ever seen a whole network. We have seen pictures and maps, but those are manufactured.

3.3 Example: What does a call center look like?

To illustrate what ThreadNet does, I will use it to show you some data from a call center in Texas (originally analyzed by Pentland, 2003). Within this call center, there were people responsible for investigating problems that could not be resolved in the initial phone contact. They investigated many different kinds of problems, but here, we focus on one type of problem.

If you collected and coded data about what people in the Investigations Unit are doing, one action at a time, it might look like Table 1. Table 1 shows two of the threads I collected. Each thread is defined by a caseID (in the first column). Within each thread, the actions are arranged sequentially, and each action is described by the actor, action and artifact involved. One could add additional attributes to describe each action, if so desired.

Table 1: Two threads

	caseID	seqNo	Actor	Action	Artifact
One thread	92870330	1	159152	1	Venice
	92870330	2	50184434	29	FILM
	92870330	3	50184434	29	FILM
	92870330	4	50184434	24	Venice
	92870330	5	50184434	21	Venice
	92870330	6	50184434	21	Venice
	92870330	7	50184434	10	Venice
	92870330	8	259152	7	CitiSmart
Another thread	92870373	1	159152	1	Venice
	92870373	2	7500531	24	Venice
	92870373	3	7500531	24	CitiSmart
	92870373	4	7500531	10	Word
	92870373	5	7500531	22	Venice
	92870373	6	7500531	6	Venice
	92870373	7	259152	7	CitiSmart

By mapping each attribute (or combination of attributes) as a color, we can visualize the threads as shown in Figure 1. This illustration is based on a problem that occurred 11 times in the data, so there are eleven threads in this sample. They all start the same way, but they vary in length and the order of what is done. In a sense, Figure 1 shows 11 color-coded stories, where each row is one story.

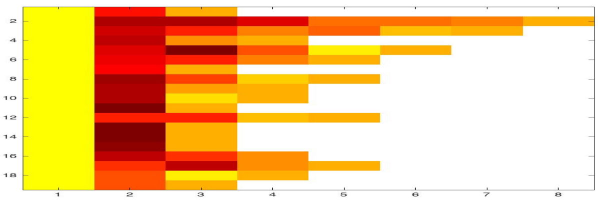
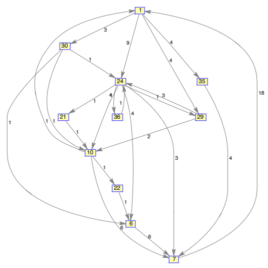
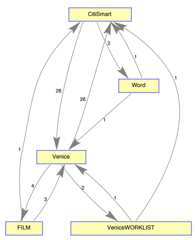


Fig. 1. Eleven color-coded threads

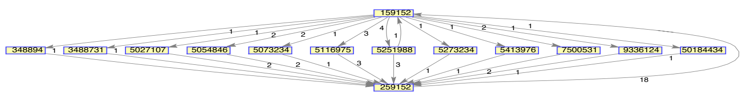
Figure 2 shows networks that are derived by tracing along these threads. For our purposes here, what matters here is that Figure 2 shows four different pictures of the *same threads,* woven in different ways by ThreadNet. The difference is simply the choice of which attributes are used to define the vertices of the network. In the first network, vertices are defined by the action attribute. Then, they are defined by the systems involved in the action, and the actors involved in the actions. Finally, they are defined by all three together. In each of these graphs, the vertices represent actions, but they are defined by specific attributes of the action and labeled accordingly. Note that this re-definition changes the structure of the graph. Presumably, it should affect the interpretation of the graph, as well.



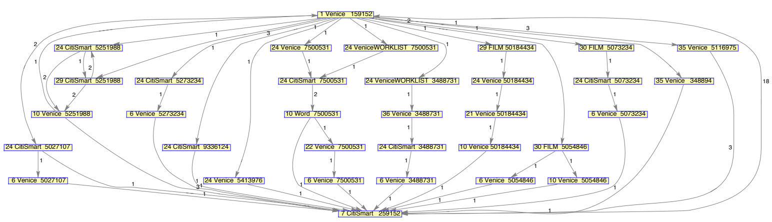
Actions only



Systems only



People only



People + Actions + Systems

Fig. 2. Four ways to visualize the same pattern of actions

We have written elsewhere about the mechanics of making these images (Pentland, Recker and Wyner, 2015). Now, we face the challenge of interpreting images that depict relations between actions, rather than relations between people, things, signs or symbols. Stated differently, we face the challenge of operationalizing Czarniawska's (1997) idea of making action nets a viable unit of analysis.

4 Relational ontology

Networks provide a vehicle for relational thinking. In his manifesto for a relational sociology, Emirbayer (1997) argued that relational thinking stands in contrast to "substantialist" thinking. The contrast is simple but profound. In the substantialist view, we interpret and explain things based on their properties. In the relational view, we interpret and explain based on relationships.

One very prominent example of relational sociology is the analysis of social networks. Another influential example is actor-network theory, which is based on tracing associations between actants (Latour, 2005). It seems clear that relational ontology has transformed social science. We see and understand the social world as networks of people and networks of actants. The impact of relational thinking is not just theoretical: just Google it. Google will sort the results of your query, in part, using the PageRank algorithm, which is based on the network of relationships among web pages.

Each of these examples is built on the idea of tracing the relationships between various kinds of entities. A natural next step would be to apply relational thinking to actions. Relationality is becoming a central concept in theorizing about the networks of action that make up organizational routines (Feldman, 2016; Feldman et al, 2016). There would appear to be an opportunity to extend this line of thinking more broadly.

Action nets pose a fascinating interpretive challenge because actions cannot be as easily taken for granted as "people." What counts as an "action" depends on context, circumstance, and its relation to other actions. Plus, because ThreadNet defines vertices based on combinations of attributes, the vertices themselves embody relationships. For example, in the bottom row of Figure 2, each vertex represents an actualized affordance (Strong et al, 2014). When viewed as affordances, things are not things and features are not features; they only become that way when they are appropriated (translated, actualized) in particular actions (Chemero, 2003).

5 Software for tracing actions

The underlying methodology of tracing actions along threads has been in the works for many years. The first examples were constructed entirely by pencil and paper (Pentland, 1999). They involved tracing the problem solving moves in a software support hot line and counting transitions with those little groups of 5 hash marks. As the volume of data increased, we developed specialized programs in Visual Basic and Microsoft Access to do the counting (Pentland, Haerem and Hillison, 2010, 2011). It worked well, but every variation required programming. In these early versions, network vertices were limited to actions. This made sense, because the goal was to put actions in the foreground and everything else in the background.

Then, in 2015, I enlisted the help of Lucy Han, an MS Analytics student with a computer science background, to create a new version of the basic algorithm for transforming threads into networks using MatLab, a programing language that is well suited to manipulating matrices. ThreadNet does what I was doing with paper and pencil twenty years ago at the dining room table: it counts transitions between categories in sequential (narrative) data. In this latest incarnation, however, there is a crucial little twist: vertices can be defined by any combination of the attributes that describe the actions.

Because they are constructed from multiple attributes, the vertices consist of related entities (e.g., people, things, actions, places, times, the name of the song that was playing, or anything else that might help describe something that happened in a story). Because you can use *any* combination of columns on the spreadsheet to make the vertices, you can bring in as much context as you please. Anything you can describe -- any contrast you can label -- can be incorporated.

Some astute readers may wonder: if you add attributes, aren't you going to get a combinatoric explosion in the number of vertices? Let's say there are 100 employees, using 10 tools, to perform 30 tasks, at 60 locations, on 2 shifts, for 300 different customers. These numbers seem modest enough, but together they generate 100x10x30x60x2x300= 1,080,000,000 possible vertices.

Fortunately, ThreadNet limits the size of the network to combinations of attributes that are actually observed. Without this methodological innovation, this approach would be limited to small toy examples. Because the world is usually somewhat structured, the network remains tractable. In data we have collected so far, a small percentage of possible nodes actually occur, and the network that connects those nodes is typically sparse (density less than 0.01) (Pentland, 2015).

6 Sociomateriality Lite

The ability to define heterogeneous vertices offers an important conceptual benefit: ThreadNet puts actants of all persuasions on an equal footing. It makes no distinction between people, things, places, times, or combinations of these categories. As a result, ThreadNet starts from a perspective that scholars of sociomateriality (e.g., Orlikowski and Scott, 2008; Pickering, 2010; Leonardi, 2013) have struggled for years to articulate, and goes from there.

6.1 Mangles and tangles of natural kinds

Vertices can be constructed with unusual combinations of entities. Words like "interpenetration" and "imbrication" seem insufficient and also superfluous. ThreadNet just concatenates the labels and tosses them into a hash table.[1] It is completely indifferent to their ontological status.

Conceptually, the resulting vertices are like little actor-networks (Pentland and Feldman, 2007). Within each vertex, the actants are associated by their co-occurrence in a particular action. In the case of an affordance network, each vertex should include a person using a tool to do a task, along with any other attributes that seem relevant. Then, in the network, each of those mangles is tangled with the others.

6.2 Heterotemporal narrative splines

Narrative is intrinsically temporal, but the time scale can vary a lot. Some processes take a few seconds, others take months or years. A little helper program (code named ThreadFactory), provides ThreadNet the information it needs to trace associations on six conventional time scales (year, month, day, hour, minute, second), in addition to sequential "event" time.

With due respect to ANT, tracing associations between *actants* is only one side of the coin. Flip it over and you can trace associations between *actions*, as well. So, by following threads (stories), ThreadNet traces the temporal associations between the vertices.

7 Is ThreadNet "beyond interpretivism"?

To even ask such a question might be a little shocking to some hard core interpretivists. After all, ThreadNet is written in a language called *MatLab*, which is a propriety product of a company called *MathWorks*. Obviously, ThreadNet is a quantitative Trojan Horse, attacking the Citadel of Semiotics with algorithms that are really

[1] The hash table is tied to integer codes that index the vertices of the network. Working by hand, you would just make a list -- a dictionary -- with one entry for each vertex in the network, and the count the entries with little hash marks. Trust me: the hash table is faster than the hash marks.

just *counting codes*. Exactly so, but the counting is not the interesting part. The interesting part is the ability to trace relations, so that the patterns of relations are available for interpretation. This seems interesting for a variety of reasons.

First, ThreadNet "de-centers" just about everything. It is a tool for the ontologically unwashed. Vertices can be nothing more than moments in time, or they can be combinations of actions, places and the color of the tablecloth. In the current version, edges represent temporal sequence, but they could represent other kinds of traces and relations. That seems like a plus.

Second, ThreadNet is relational most of the way down. At the foundation, there is a spreadsheet; this is suspicious, because the spreadsheet has columns. The labels on the columns represent categories (e.g., people, actions), and those categories stink of substantialist residue. But what goes in the columns? In the columns of the spreadsheet, we put words to describe actions, people, places, times, artifacts, systems, and whatever else seems relevant to the story and the context in which the story occurs. Inescapably, the meaning of each of those words exists in relation to the other words. That doesn't seem so bad.

Third, ThreadNet creates pictures, but it is not like a camera or a fancy medical imaging technology, such as MRI. That analogy fails because it implies that there is something really, objectively there that we can only see with a particular kind of camera (e.g., a tumor). We can be grateful that medical physicists are good at what they do, but we are doing something different.

In particular, the phenomena visualized by ThreadNet (e.g., performances of organizational routines) are intrinsically ephemeral, heterogeneous, and contextual. They are collections of moments in time that we describe in various levels of detail, at various time scales, from various points of view. To the extent that those moments can be recorded as parts of coherent narratives, we can trace the associations between them. Like those fancy medical cameras, ThreadNet allows us to see pictures we can't see any other way. Unlike those cameras, it makes pictures that seem less likely to have clear-cut physical or biological interpretations.

8 Limitations

Collecting sequential data raises issues about granularity and observability that need to be resolved in any study that involves sequential data. Actions can always be subdivided into micro-actions, as we slide down the slippery slope of reductionism, in the vain hope of finding some kind of "micro-foundation." And when we attempt to trace threads, the trail often goes cold. People stop, think, multi-task, delay, forget, get interrupted... In some cases, our digital companions can help by creating event logs, but they also do a lot of things we cannot readily observe (e.g., transmitting encrypted data). ThreadNet does not solve any of these problems.

9 What next?

Relational thinking has been a transformative influence in social science. We are a fair distance down the road when it comes to getting beyond simple substantialism, but it takes time. For example, after more than a decade of restating the obvious, we are starting to get beyond routines as things (Feldman et al, 2016). Relational thinking is helping us to get beyond things as things, as well. We are appreciating and interpreting relations between actants (humans, machines, concepts, etc.) Perhaps now we can flip the coin and start to appreciate and interpret relations between actions, as well.

References

1. Bruner, J.: *Actual Minds, Possible Worlds*. Harvard University Press, Cambridge, MA (1986)
2. Chemero, A.: "An Outline of a Theory of Affordances", *Ecological Psychology* (15:2), pp. 181-195 (2003)
3. Czarniawska, B.: *Narrating the Organization: Dramas of Institutional Identity*, Chicago & London: The University of Chicago Press (1997)
4. Czarniawska B.: On time, space, and action nets. *Organization*, 11(6), 773-791 (2004)
5. Emirbayer, M.: Manifesto for a relational sociology. *American journal of sociology*, 103(2): 281-317 (1997)
6. Emirbayer, M., & Mische, A.: What is agency? *American journal of sociology*, 103(4), 962-1023 (1998)
7. Feldman, M.S.: Routines as process: Past, present and future. In Claus Rerup and Jennifer Howard-Grenville (eds.). *Organizational routines and process organization studies.* Perspectives on Process Organization Studies Series, Oxford University Press, Oxford (2016)
8. Feldman, M. S., Pentland, B. T. D'Adderio, L., Lazaric, N.: Beyond routines as things: Introduction to the special issue on routine dynamics, *Organization Science* (2016)
9. Latour, B.: *Reassembling the Social: An Introduction to Actor-Network Theory*. Oxford University Press, Oxford (2005)
10. Leonardi, P. M.: Theoretical foundations for the study of sociomateriality. *Information and Organization*, 23(2), 59-76 (2013)
11. Lyman, S., and M. Scott: "Accounts." *American Sociological Review*, 33: 46-62 (1968)
12. Orbuch, T. L.: People's accounts count: The sociology of accounts. *Annual Review of Sociology*, 23: 455-478 (1997)
13. Orlikowski, W. J., & Scott, S. V.: Sociomateriality: Challenging the Separation of Technology, Work and Organization. *The academy of management annals*, *2*(1), 433-474 (2008)

14. Pentland, B. T.: Organizations as Networks of Action, in J. Baum and B. McKelvey, (Eds.), *Variations in Organization Science: In Honor of Donald T. Campbell,* Thousand Oaks, CA: Sage., pp. 237-253 (1999)
15. Pentland, B. T.: Sequential variety in work processes, *Organization Science*, 14(5): 528-540 (2003)
16. Pentland, B. T. and Feldman, M. S.: Narrative Networks: Patterns of technology and organization, *Organization Science,* 18(5), 781-795 (2007)
17. Pentland, B. T., T. Hærem and D. Hillison: The (n)ever changing world: Stability and change in organizational routines, *Organization Science,* 22(6), 1369-1383 (2011)
18. Pentland, B. T., T. Hærem and D. Hillison: Comparing Organizational Routines as Recurrent Patterns of Action, *Organization Studies,* 31(7), 917-940 (2010)
19. Pentland, B.T.: Innovation in routines: An Affordance Network Perspective, *European Group on Organization Studies*, Athens, Greece, July 2015 (2015)
20. Pentland, B.T., Recker, J. and Wyner, G.: A thermometer for interdependence: Exploring patterns of interdependence using networks of affordances, Thirty Sixth International Conference on Information Systems, Ft. Worth, TX (2015)
21. Pentland, B.T., Yoo, Y.: Social fabric from digital threads: Generativity and innovation in organizational routines Presented at the VU Workshop on Organizing for Digital Innovation, March 11-12, 2016, Amsterdam (2016)
22. Pickering, A.: *The mangle of practice: Time, agency, and science*. University of Chicago Press (2010)
23. Strong, D.M., Volkoff, O., Johnson, S.A., Pelletier, L.R., Bar-On, I., Tulu, B., Kashya, N., Trudel, J., and Garber, L.: A Theory of Clinic-EHR Affordance Actualization, *Journal of the Association for Information Systems* (15:2), pp. 53-85 (2014)
24. Weick, K. E.: *Sensemaking in organizations*. Sage, Thousand Oaks, CA. (1995)