

# Identifying Uptake, Sessions, and Key Actors in a Socio-Technical Network

Dan Suthers

Dept. of ICS, University of Hawaii  
1680 East West Road, POST 317  
Honolulu, HI 96822  
1.808.956.3890  
suthers@hawaii.edu

Nathan Dwyer

Ibis Networks  
828 Fort Street Mall, Suite 600  
Honolulu, HI 96813  
1.808.485.9044  
ndwyer@gmail.com

## Abstract

*This paper outlines an implementation of an analytic framework that addresses the need to understand socio-technical networks as intertwined processes at local and network levels of agency distributed across various media. We integrate data sources in a single abstract transcript of events, and identify empirical relationships between these events to build a graph of uptake relations—how one act builds on another. We then use cohesive subgraph detection to identify “sessions” in the space-time dimensions represented by these graphs. Analysis can then focus on individual sessions and identify key actors within these sessions using sociometrics, or find relationships between sessions that might be vectors for the transmission of ideas or practices. An example is provided using data from the Tapped In educator professional network.*

## 1. Introduction

Socio-technical networks such as online communities and social media can involve intertwined processes at individual, small group, and network levels of agency distributed across various media. Phenomena of interest may involve multiple processes that have been theorized and investigated separately [11, 14]. For example, theories of learning in social settings vary in the agent of learning (e.g., individual, small group, or community), and in the process or ‘mechanism’ of learning (e.g., information transfer, intersubjective meaning-making, shifts in participation and identity, and accretion of cultural capital) [17]. Learning and other phenomena of interest take place simultaneously at all of these levels of agency and with all of these processes. Thus, understanding socio-technical systems requires revealing the relationships between individual and collective levels of agency and potentially coordinating analysis at multiple time scales [9].

The multifaceted and multi-granular nature of participation and production of individual and collective value in socio-technical networks presents an analytic challenge that motivates the work reported here: how to meaningfully connect analytic approaches

across levels of analysis. For example, interpretative micro-analysis of interaction (e.g., of chat sessions) shows actors' concerns, orientations, and accomplishments in the moment, but the relationships latent in these interactions must be identified and aggregated to identify clusters of activity and communities of actors. Yet, aggregated graphs of ties fail to capture the means by which these ties were accomplished, so such graphs are insufficient for understanding the network without also examining the interactions from which they were derived. Communities of actors empirically emerge from the moment-to-moment interactions between actors, but also influence the potential for such interactions. These observations motivate our use of layered data and analytic representations that bridge between session and network granularities of analysis.

We developed our methods using historical data from Tapped In®, an environment that offered asynchronous threaded discussions, quasi-synchronous chat, file sharing, and other media for interaction [6, 16]. Participants included a mixture of members of organizations and others who came to the socio-technical network of their own accord. All participants were free to wander between specific settings and events. Consequently, the data generated by a given participant resulted in different kinds of traces in the log files associated with these media, at different times and different virtual spaces. The trace of a participant’s unitary experience is fragmented across these logs, and needs to be reassembled to reveal this activity. For example, Tapped In chat events were recorded in distinct files for each chat room, while other events such as posting discussion messages or sharing files were recorded in a database as web interface events. Thus, if a participant chatted in a reception room and participated in chat-based events held in two group-meeting rooms, and also read some threaded discussion messages in one of the rooms, that participant’s activity would be recorded in four different persistent stores. This presents a second analytic challenge: how to reassemble fragmented traces of activity into a single analytic artifact.

To preview our approach, logs of events are abstracted and merged into a single abstract transcript of events, and this is used to derive a series of representations that support levels of analysis of interaction and of ties. Three kinds of graphs model interaction: *Contingency graphs* record how events such as chatting or posting a message are observably related to prior events by temporal and spatial proximity and by content. *Uptake graphs* aggregate the multiple contingencies between each pair of events to model how each given act may be “taking up” prior acts. *Session graphs* are abstractions of uptake graphs: they cluster events into spatio-temporal sessions and capture relationships between sessions. Relationships between actors and artifacts are abstracted from interaction graphs to obtain “*associograms*” [20, 21], which can be folded into traditional *sociograms*.

Other publications have detailed some of the theory [8, 17] and analytic representations [19, 21] behind this work. In this paper we report on computational methods we have developed for transforming log files into interaction and session graphs and sociograms, and methods for drawing conclusions based on these representations. First, we provide an overview of the series of computational transformations taken. Then, details of the methods will be provided in the context of an extended example using the Tapped In data.

## 2. Overview of the Framework

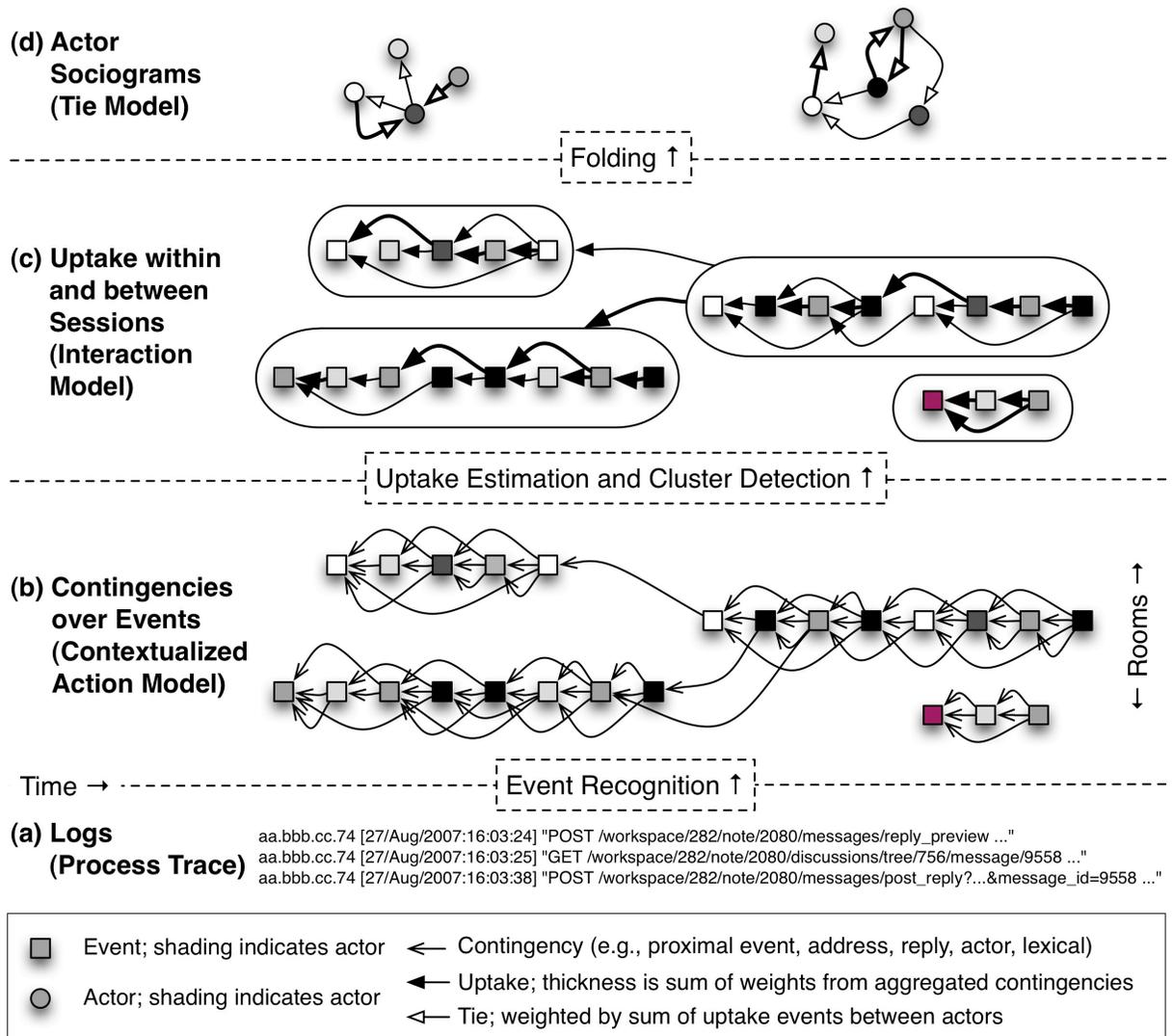
The representations used at various levels of analysis are shown in Figure 1 (next page). At the bottom, we exemplify various traces of activity (such as log files of events) that provide the source data. (HTTP logs are shown just for illustration: our actual data includes database logs and textual transcripts of chats.) These are parsed using methods that are necessarily system-specific to abstract log entries into an event stream, shown in the second level (boxes in Figure 1b). This event stream is represented using an abstract transcript format that records actor and media participants in events. Meaningful events at this level of analysis may not have a 1-1 correspondence to log file events. For example, the act of posting a message with previewing may show up as multiple GET and POST entries in an HTTP log file.

At this level of abstraction (Figure 1b), we compute *contingencies* between events, to produce a model of how acts are mutually contextualized (shown as arrows). Human action is contingent upon its setting in diverse ways: our computational methods capture some of these contingencies that are amenable to automated detection. A contingency called *proximal event* reflects the likelihood that events occurring close

together in time and space are related. For example, in analyzing quasi-synchronous chat, contingencies are installed to prior contributions in the same room that occur within an adjustable time window but not too recently. A Keystroke Level Model [13] of how long a typist would have taken to type the contribution is used to ensure that the prior contribution was already visible when typing began by closing the window this amount of time before the contribution. *Address* and *reply contingencies* are installed from an utterance mentioning a user by name to the last contribution (address) and next contribution (reply) by that participant within a time window, using a parser/matcher of user IDs to first names. *Same actor contingencies* are installed to prior acts of a participant over a larger time window to reflect the continuity of an agent’s purpose. Overlap in content as represented by sets of lexical stems is used to produce a *lexical contingency* weighted by the number of terms overlapping. Semantic analysis can also be done. Our development strategy is to see how far we can get with these simpler contingencies before adding computational complexity. Examples of specific contingencies are given later in the paper.

The resulting *contingency graph* (e.g., Figure 1b) is represented as the first layer of abstraction in what we call an Entity-Event-Contingency graph or EEC [19]. In the EEC, graph vertices are Events. These can be of various types (e.g., enter chat, exit chat, chat contribution), and are annotated with time stamps and optionally associated Entities, which include human actors, content (e.g., message content), and locations (e.g., rooms) involved in the event. Contingencies are typed edges between vertices (the types were described in the previous paragraph), and there may be multiple edges between any two vertices (e.g., two proximal events by the same actor with lexical overlap).

It is necessary to collapse the multiple edges between vertices into single edges for two reasons. First, most graph algorithms assume that there is at most only one edge between two vertices. Second, and more importantly, we are interested in *uptake*, the relationship between events in which a human action takes up some aspects of prior events as being significant in some manner [19]. For example, replying to prior contributions in a chat or discussion is an example of uptake. The concept of uptake is not specific to a medium (it can cross media) or limited to “transactivity” [3] (uptake need not be explicitly directed at a dyadic partner). Contingencies are of interest only as collective evidence for uptake. So, we abstract the contingency graph to an uptake graph, using a weighted (and presently linear) combination of the various types of contingencies between contributions (vertices) to derive a single uptake



**Figure 1. Levels of Analysis and their Representations**

relation represented as a graph edge weighted to reflect strength of evidence in the contingencies. Different weights can be used for different purposes (e.g., finding sessions; analyzing the interactional structure of sessions).

As shown in Figure 1c, uptake graphs are similar to contingency graphs in that they also relate events, but they collect together bundles of contingencies between a given pair of vertices into uptake relations, optionally filtering out low-weighted bundles. At this point, we can do several interesting things with these uptake graphs. A graph clustering algorithm is applied to the uptake graph to find clusters of related contributions that we call “sessions” (indicated by rounded containers in Figure 1c). A session can cross

settings such as chat rooms. Inter-session and intra-session analysis proceeds from here.

For *inter-session analysis*, we collapse each session into a single vertex representing the session, but retain the inter-session uptake links. (For example, there are four sessions in Figure 1c and two inter-session uptakes.) These inter-session links indicate potential influences across time and space from one session to another. An example will be given shortly in conjunction with Figure 4.

For *intra-session analysis*, the uptake graph for a session is isolated. Two paths are possible from here. The sequential structure of the interaction can be micro-analyzed to understand the development of group accomplishments: this part is not automated,

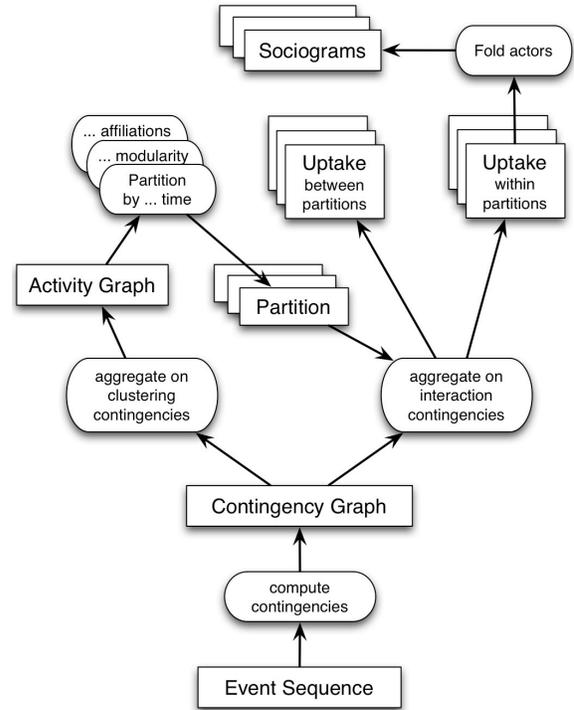
although future work includes analysis of uptake patterns in the form of graph configurations to identify sessions with features of interest. Or we can fold the uptake network into an actor-actor sociogram (directed weighted graph), where the tie strength between actors is the sum of the strength of uptake between their contributions (Figure 1d). This sociogram can be analyzed using conventional social network analysis methods such as degree or eigenvector centrality to identify key actors, etc. [12, 23].

In summary, as an analytic integration tool our framework provides multiple pathways for analysis, some of which are diagrammed in Figure 2. Contingencies are applied to events in the EEC abstract transcript to produce a contingency graph. Contingencies are then aggregated into uptake between the same events. A single aggregation can be used, or optionally (as shown in Figure 2) different weightings can be used for identifying sessions by graph partitioning versus for doing detailed interaction analysis. (We are currently studying whether it is productive to use different weightings for these purposes.) In either case, uptake that crosses partitions can be used to identify influences across space and time, and uptake within partitions can be analyzed to study the interactional structure of a session. Uptake graphs may be folded into networks where nodes are actors rather than events, to which sociometrics are applied.

Another line of analysis not discussed in this paper is to fold events into actor-artifact networks, or bipartite weighted directed graphs that are called “associograms” for short, because they capture how actors are associated with each other via mutual read and write of media objects. In that line of work, we have undertaken community analysis of associograms to detect not only human participants in communities, but also the artifacts that reflect their mediated nature (e.g., synchronous or asynchronous) [20].

### 3. Implementation

Our primary implementation is in Java. The core implementation includes classes defining events and entities that participate in events, along with base classes for contingencies that relate events, and a class that collects them into EEC graphs. Composite contingencies and events aggregate these objects to represent uptake and sessions, respectively. Domain model classes define specific kinds of entities such as actors, chats, discussions, etc., and datasource classes define how to import log files into these representations. We use the Hibernate object/relational model and persistence engine ([hibernate.org/orm](http://hibernate.org/orm)) to



**Figure 2. Process Model**

enable processing of large graphs and storage of results.

Layered on top of the EEC, analyzer classes define workspace objects for specifying and storing analyses, classes that run specific analysis steps (e.g., computing lexical stems and contingencies, combining contingencies into uptake, finding sessions, exporting graphs), and another class defining scripts that control these steps. As exemplified in Tables 1 and 2, XML scripts control processing, including selection of type and range of source data from the EEC (not shown); sequencing of analytic steps (partially shown in Table 1); and weighting of contingencies (Table 2). We call out to the JVM-based implementation of Python ([www.jython.org](http://www.jython.org)) to use the NLTK library ([nltk.org](http://nltk.org)) for lexical processing, and spawn external processes to utilize the iGraph package ([igraph.sourceforge.net](http://igraph.sourceforge.net)) for graph operations such as graph partitioning. Presently we run analyses in Eclipse ([www.eclipse.org/](http://www.eclipse.org/)) and export results for viewing in tools such as Gephi ([gephi.org](http://gephi.org)).

### 4. Example

Here we illustrate the approach with an analysis we conducted of data from the Tapped In network.

**Table 1. Portion of an XML analysis script**

Analysis is controlled by a script such as the following. Time window sizes over which contingencies scope have been chosen by experimentation with our data, and can easily be adjusted for other data.

```
<step bundlename="apps.analyzer" classname="apps.analyzer.script.PythonScriptStep" >
  <stepconfig scriptref="nlk/lancaster_stemmer.py" />
</step>
<step bundlename="apps.analyzer" classname="apps.analyzer.script.ProximalEventRule">
  <stepconfig windowsize="120" tag="PE120" />
</step>
<step bundlename="apps.analyzer" classname="apps.analyzer.script.LexicalRule">
  <stepconfig windowsize="300" tag="LR300" />
</step>
<step bundlename="apps.analyzer" classname="apps.analyzer.script.AddressRule">
  <stepconfig windowsize="300" tag="AR300" />
</step>
<step bundlename="apps.analyzer" classname="apps.analyzer.script.ReplyRule">
  <stepconfig windowsize="300" tag="RR300" />
</step>
<step bundlename="apps.analyzer" classname="apps.analyzer.script.SameActorRule">
  <stepconfig windowsize="300" tag="SA300" />
</step>
... steps to find sessions, compute uptake and sociograms, and export graphs follow ...
```

**Table 2. A weighting scheme for combining contingencies into estimations of uptake**

LexicalContingency is additionally weighted proportional to the number of overlapping lexical stems. Weights reflect how strongly a contingency evidences uptake (e.g., mentioning someone by name is stronger evidence than merely chatting soon after that person), and are being adjusted through experimentation with our data. We do not claim that these weights are the best for other applications.

```
<weighter bundlename="apps.analyzer" classname="apps.output.weighting.StandardWeighter">
  <weighting defaultweight="1" threshold="2">
    <entry typesuffix="AddressContingency" weight="3" />
    <entry typesuffix="ReplyContingency" weight="3" />
    <entry typesuffix="SameActor" weight="2" />
    <entry typesuffix="LexicalContingency" weight="1" />
    <entry typesuffix="ProximalEvent" weight="1" />
  </weighting>
</weighter>
```

## 4.1 Tapped In

This study drew on data from SRI International's Tapped In® (tappedin.org), an international online network of educators involved in diverse forms of informal and formal professional development and peer support [6, 16]. According to its developers, Tapped In was motivated by the desire to understand how to initiate and manage large heterogeneous communities of educators, how such communities evolve, and the benefits that participants derive from their involvement. This network included activities that were sponsored by formal organizations (e.g., universities, school districts, and nonprofits) mixed with volunteer driven and other unsponsored activities, in both synchronous and asynchronous media, with participants from across all career stages and diverse occupations related to education. Thus, Tapped In is an opportunity to develop and test hypotheses, tools, and techniques for understanding heterogeneous networks. Cumulatively, Tapped In hosted the content and

activities of more than 150,000 education professionals (over 20,000 per year in our study period) in thousands of user-created spaces that contain threaded discussions, shared files and URLs, text chats, an event calendar, and other tools to support collaborative work. Over its 16-year history, more than 50 organizations, including education agencies and institutions of higher education, consulted with Tapped In staff and became "tenants" in the system with online courses, workshops, seminars, mentoring programs, and other collaborative activities. There were also approximately 40-60 public activities per month designed by Tapped In members and open to anyone in the community (including tenant members). Volunteers drove the majority of Tapped In community-wide activity [6]. Extensive data collection capabilities captured the activity of all members and groups, including chat data, discussion board interactions, and file sharing. The trend in current social media is for such data to be treated as a proprietary asset, so the Tapped In data corpus offers a valuable opportunity for study.

**Table 3. Sample chat from Tapped In**

Full names have been abbreviated to initials.

23:35 M: are all good teachers good mentors?  
23:38 A2: some people will take a while to get to that point  
23:42 A2: No..not all  
23:51 E: definitely not  
23:55 L: Training can help, but I think some is personality  
24:09 A: some people are excellent teachers but are horrible mentors  
24:09 E: some great teachers can not hold a decent conversation with an adult  
24:11 A2: i had to co-ops who would be awful mentors  
24:24 L: Nods  
24:27 D: That is an interesting question M, ... I would probably say 'yes' first off, and then wonder some more  
24:42 M: it is something I have thought about often D  
24:47 A2: I think its alot of personality  
25:17 D: one thing a mentor has to know is how to operate with a peer, and ow to be intentional about handing over, or encouraging greater independence  
25:18 M: observation has made me think that it takes an extra “special ingredient” to tip the scales

We selected a period from September 2005 through May 2007 for our research, and used smaller samples within this period to develop and test our methods. This period was chosen because graphs of activity showed peak usage during this time, and we wanted to study a successful network. (Usage tapered off after 2010, and Tapped In was shut down in 2013.) Here we report an example analysis of 3 days of data, which includes a session of particular interest in the second day. This session, a “Teaching Teachers” session on mentoring, had previously been chosen for micro-analysis due to its high quality of interaction. A brief sample of the chat is shown in Table 3. We wanted to test whether the methods described above would detect anything significant about this session and how it was embedded in its surrounding context.

The first step was to import data from the log files into our EEC format. Tapped In log files were in both database format and raw text files for chat transcripts: custom translators were written to import a sequence of events organized by time (boxes in Figure 1b).

## 4.2 Contingencies and Uptake

An XML script drives the contingency and uptake analysis, as discussed previously. Educator chats tend to be deliberative; hence, for our data we set the window for the Proximal Event contingency at 120 seconds before the contribution (Table 1). Shorter durations have been used for faster paced chats [15]. We are initially using conservative parameters for Keystroke Level Modeling, namely expert typist values

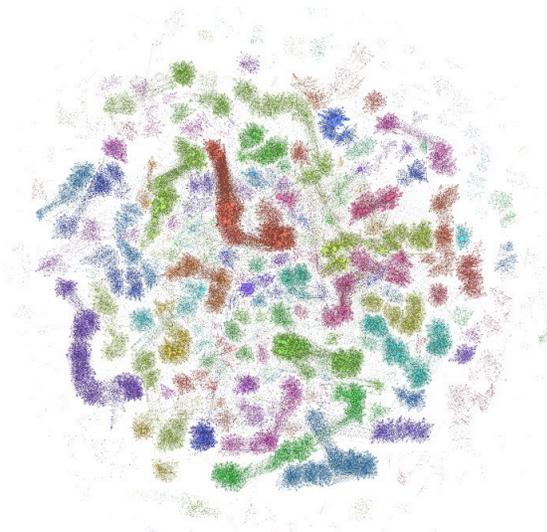
published in [13] of  $M=0.6$  and  $K=0.08$ , where  $M$  is the average time for a mental operation initiating the action and  $K$  is the average time to strike a key. For example, the contribution by A2 at 23:38 in Table 2 is 50 characters long:  $0.6 + (50*0.08) = 4.6$  seconds, so this cannot be taking up M’s question at 23:35 (the full transcript shows that it continues a prior question). In contrast, M’s question is within the proximal event scope of A2’s contribution at 23:42. Similarly, the simultaneous contributions at 24:09 cannot be contingent on each other, and 25:18 cannot be contingent on 25:17 even though D and M are clearly in conversation with each other.

Similar computations are used to limit the temporal scope of other contingencies, although a larger time window of 300 seconds was used (Table 2). An Address Contingency is placed between D’s 24:27 “That is an interesting question M” and M’s prior question at 23:35; while a Reply Contingency is placed between M’s “... thought about often D” at 24:42 and D’s 24:27. All of M’s later contributions are contingent on the earlier ones by Same Actor in this short excerpt. Although most contingencies are limited to events in the same room, we allow Same Actor to cross rooms. Exemplifying the Lexical Contingency, the lexical stems for A2’s contribution at 23:38 (“some people will ...”) are {peopl, tak, get, point} and for A’s contribution at 24:09 (“some people are ...”) are {peopl, excel, teach, horr, ment}. These overlap on “peopl”. (Stop words such as “some” have been filtered. The “some people” similarity of wording suggests adding a new contingency based on repeated bigrams or n-grams.)

The resulting graph can have multiple contingencies between a pair of events, and is too complex to visualize (Figure 1b offers a schematic). Many network algorithms can only operate on graphs with a single edge or arc between any two vertices (nodes), and we want to estimate the extent of uptake between acts. Hence, using weights like those in Table 2, we collapse the multiple contingencies between two events into a single weighted arc representing the extent to which the second is related to the first: the “uptake graph” of Figure 1c. All operations described below are on the uptake graph except where noted.

## 4.3 Sessions

The next step was to empirically identify sessions (temporally contiguous interactions between a set of actors) in the uptake graph. Although Tapped In had scheduled “calendar events” where participants met in a particular room at a particular time, there were also many other sessions that took place spontaneously or were not announced in the calendar, and sometimes

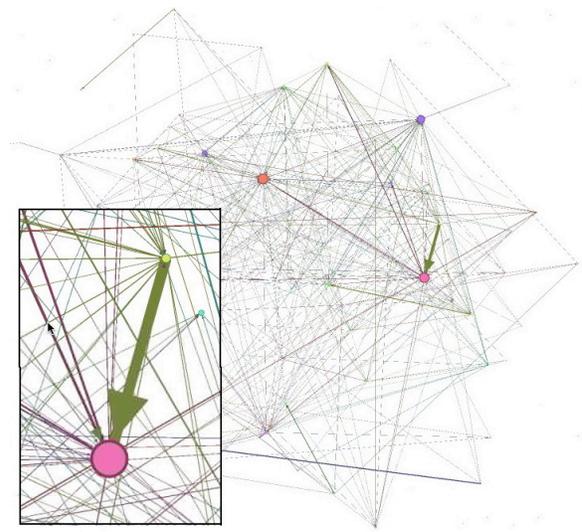


**Figure 3. Uptake Graph for Three Days**

even formal events would move between rooms. Therefore, we sought to empirically identify the sessions that actually took place. We compared algorithms for detection of cohesive subgraphs (“community detection algorithms”, [7]), applied to the uptake graph to identify sessions. A future publication will report the relative merits of several algorithms and the definitions of “session” that they imply. The example reported here used a modularity partitioning algorithm [4] implemented in iGraph, which implies that sessions are partitions of the uptake graph that are more highly connected within each partition than expected at random [12]. Modularity partitioning can find clusters of activity even when there is continuous activity resulting in a weakly connected uptake graph. However, when sessions have clear beginnings and ends without overlap (i.e., contingencies do not span the gaps), simply finding weakly connected components is sufficient.

In Figure 3 we show the uptake graph for the three-day period. All visualizations are in Gephi (gephi.org, [2]). This particular visualization uses the OpenOrd layout algorithm [10], a hierarchical version of traditional force-directed algorithms that group nodes according to their interconnectedness. Nodes are individual chat or discussion contributions, and colors represent the modularity partitions.

Each colored connected cluster corresponds to a session. The visualization made clear that there were a surprising number of interactive sessions taking place in Tapped In over this three-day period: there are 112 partitions. Interestingly, some sessions crossed rooms: sometimes a “tour” would start in one room and move to others, or persons would meet in the Reception room and then move elsewhere. Although we find it to be



**Figure 4. Inter-session uptake graph with close-up view**

interesting and sometimes useful to inspect them, visualizations such as Figure 3 are in general difficult to interpret, so we use computational tools for further analysis as follows.

#### 4.4 Inter-Session Relations and Key Sessions

After finding modularity partitions of the activity graph, the analysis can take two paths, as was shown in Figure 2. Within a session, analysis of uptake structure between contributions and (by folding) between actors helps us understand interaction and roles, respectively. Analysis of uptake between sessions is of interest for understanding how actors and ideas move between settings across time and space in a socio-technical system. Both of these paths are exemplified below.

Figure 4 shows each session collapsed into a single node. Weights on edges crossing sessions were summed, and the weighted in-degree for each session is shown as node size. Several of the sessions have larger node size, indicating their apparent influences on subsequent sessions. Of particular interest is the larger pink session node with a large green arrow pointing to it from a smaller light green session node (inset). What is the relationship between these sessions? First we examined the pink session, the chronologically prior session being “taken up” by the later one.

#### 4.5 Session Example

It turned out that the modularity partition visualized as the large pink node almost exactly captured the Teaching Teachers session on mentoring in the schools. The events placed in this partition included all

of the Teaching Teachers session contributions, and only a few other events in other locations at the beginning and end (from the Same Actor contingency). Events in this partition included initial entries into the Teaching Teachers room and introductions, followed by an in depth discussion of peer mentoring of teachers in schools, facilitated by M, lasting nearly an hour (excerpted in Table 3). Near the end, M mentions that she needs to leave for another discussion, and the others thank her and say goodbye. Interestingly, the modularity-partitioning algorithm placed in this session the first few utterances of M showing up in the new session in another room, which we shall call an “In Training” group.

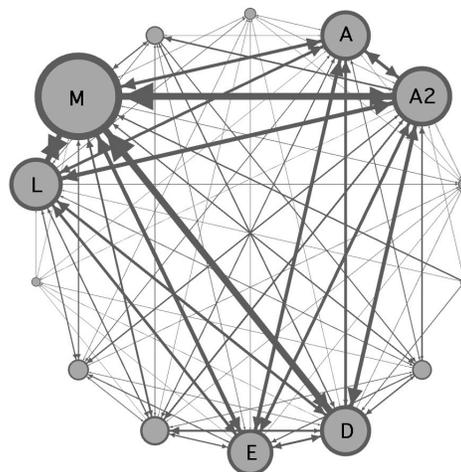
On inspection, we found that other partitions in this case study, as well as dozens of partitions from runs on other space-time spans of data, also corresponded to sessions with clear interpretations as to the boundaries and purpose of the gathering. This is remarkable because the partitions were derived purely with algorithmic methods: no information about sessions was provided to the algorithms. Thus the method shows promise as an automated way of identifying meaningful social events.

#### 4.6 Inter-Session Relations

As visualized in our graphs, uptake arcs are drawn from the chronologically later event to the prior event being taken up, and the same is true of the collapsed session graph in Figure 4. What is the nature of the session that is the source of the large green arrow in the inset of Figure 4, i.e., that it depends on the session we just examined? Looking at the event sequence for this session, we found that it is the In Training session that the facilitator M has just joined. Furthermore, two of the participants in her previous Teaching Teachers session, A2 and L, followed her there. This is the reason for the thickness (weight) of the green arrow: three actors have moved from one session to another, resulting in Same Actor contingencies connecting the sessions. This relationship suggests that it might be fruitful to see whether any ideas discussed in the Teaching Teachers session were carried over to In Training.

#### 4.7 Intra-Session Analysis of Key Actors

A preliminary analysis of the uptake structure of the Teaching Teachers session is reported in [18]. Here we focus on the sociogram level of analysis (Figure 1d). Folding the uptake graph for this session into an actor-actor sociogram, we obtained the graph shown in Figure 5. Node size is weighted in-degree, showing the relative importance of actors in terms of the extent to



**Figure 5. Session sociogram with weighted in-degree.**

which we estimate that others take up their contributions. (Eigenvector centrality can also be used to estimate transitive importance.) Not surprisingly, the most uptake was from contributions by the session leader M. Inspecting user profiles and prior activity, we found that two other prominent participants, L and D, were experienced members who sometimes facilitated their own sessions. Additionally, the sociogram helped us notice the importance of other actors: A, A2 and E. A2 was a “newbie” trying out Tapped In for her first time, and it seems significant that she followed M into the next session.

## 5. Discussion

### 5.1 Summary and Applications

This paper outlined an analytic approach in which log events are abstracted and merged into a single abstract transcript of events, and this is used to derive a series of representations that support multiple levels of analysis of interaction (contingency and uptake graphs, and session graphs) and of ties (associograms and sociograms). An example illustrated how our analytic framework and supporting algorithms can (1) find relationships between contributions in the relatively unstructured medium of chats; (2) parse the stream of activity into sessions defined as densely connected clusters of activity based on these relationships; (3) make influences between sessions across time and space visible; and (4) enable sociometric analysis of individual sessions.

This capability makes it possible for managers of socio-technical networks such as online communities, MOOCs, etc. to automatically find clusters of

interaction between participants and to identify important actors in these sessions. For example, the software could be adopted to take periodic (daily, weekly) log files, and generate reports of the size and locations of sessions and of prominent actors in these sessions. Tapped In community managers (personal communication) have indicated that, due to the volume of activity and the mixture of sponsored and unsponsored events, they could not keep track of everything taking place in their environment, and would have benefited from knowing about and offering support to unknown key actors.

The approach can be applied in any socio-technical network in which actors produce contributions visible to other actors, provided that it is possible to (1) obtain a log file and stream of events that identifies the actor, time, location, and content of each act; and (2) there is a basis for inferring which other actors potentially accessed (read) these contributions. It can be generalized to asynchronous discussions, although some rules for generating contingencies will need to be rewritten to acknowledge the different temporality of asynchronous discussion and to leverage other information such as thread structure and quotation practices [1]. We have begun such a generalization with Tapped In data, and plan to integrate it with the chat analysis for cross-media analysis.

## 5.2 Contributions

Our work addresses two analytic challenges arising from the nature of participation in socio-technical networks. First, since learning and knowledge production in socio-technical networks takes place through a synergistic mixture of individual and collective agency, we need to understand aggregate phenomenon (e.g., “ties”, “roles”, and “communities”) as both produced by and providing the setting of specific interactional events. Our framework addresses this with linked abstractions that coordinate multiple levels of analysis. Second, participant interaction is distributed across media, places and time in these environments, potentially resulting in separate traces of interaction that fragment their unitary experience. Our framework addresses this by building on an abstract transcript of interaction.

Algorithms for cohesive sub-cluster detection have been used extensively in studying socio-technical networks [7], but usually on graphs of ties between actors. Our innovation is to apply cohesive sub-cluster detection to graphs of relationships between events (such as chat contributions) to find “communities” of events in a time-space matrix; that is, sessions. Once a session is identified, we can fold the session graph into

a sociogram of ties between actors. Others have done this as well using temporal proximity (e.g., Rosen & Corbit [15]), but we also consider other relations between events to provide a richer basis for session identification and subsequent analysis of activity and actors within sessions. Work by Trausan-Matu on “polyphonic analysis” [22] has affinities to our use of multiple contingencies, but has only recently been abstracted to higher levels of analysis. A thesis by Charles [5] has provided an alternative implementation of our approach and extended the set of contingencies. Our approach dovetails with work that applies natural language processing methods for syntactic, semantic and pragmatic analysis of interactional structure, and indeed rules for generating additional contingencies could be derived from such research.

## 5.3 Future Work

Ongoing work includes experimenting with different algorithms for identifying sessions, and tuning weights on contingencies for capturing the interactional structure of sessions. Once these are well grounded, we can study how patterns of uptake and metrics on the derived sociograms relate to session quality, and use these to automatically identify sessions of interest. Two other near-term concerns include using inter-session relationships to trace the spread of ideas, and incorporating analysis of asynchronous discussions for analysis of cross-media influences.

Looking further ahead, automating the generation of interaction and social network graphs opens up several new research approaches for relating fine-grained interaction to more aggregated levels of analysis. One approach is to generate multiple social networks during a session, and track the change in actors’ relational properties (e.g., reciprocity, clustering coefficient, and centrality) in order to recognize significant changes to the group structure or role emergence in individuals. We could also automate the generation of social network graphs at significant points over the entire history of the online environment, to identify frequently interacting individuals and track the growth and fragmentation of overlapping communities. It might be possible to identify critical points in the formation of healthy or unhealthy communities.

All of these analytic options are products of our system’s abstraction away from media-specific forms and the automation of mapping between levels of analysis that this abstraction enables.

## 6. Acknowledgements

Many thanks to Mark Schlager, Patti Schank and Judi Fusco of SRI for sharing their data and expertise, and Kar-Hai Chu and Devan Rosen for their collaboration on this project. Reviewers provided valuable advice on improving the presentation. This work was supported by NSF Award 0943147. The views expressed herein do not necessarily represent the views of NSF.

## 7. References

- [1] F. Barcellini, F. D tienne, J.-M. Burkhardt and W. Sack, Thematic coherence and quotation practices in OSS design-oriented online discussions, Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work (GROUP '05), ACM Press, Sanibel Island, Florida, USA, 2005, pp. 177-186.
- [2] M. Bastian, S. Heymann and M. Jacomy, Gephi: An open source software for exploring and manipulating networks, International AAAI Conference on Weblogs and Social Media, 2009.
- [3] M. W. Berkowitz and J. C. Gibbs, A Preliminary Manual for Coding Transactive Features of Dyadic Discussion (unpublished manual), Ohio State University, 1979.
- [4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment, <http://dx.doi.org/10.1088/1742-5468/2008/10/P10008> (2008).
- [5] C. Charles, Analysis of Communication Flow in Online Chats, Department of Computer Science and Applied Cognitive Science, Unpublished Master's Thesis, University of Duisburg-Essen, Duisburg, Germany, 2013, pp. 90.
- [6] U. Farooq, P. Schank, A. Harris, J. Fusco and M. Schlager, Sustaining a community computing infrastructure for online teacher professional development: A Case Study of Designing Tapped In, Computer Supported Cooperative Work, 16 (2007), pp. 397-429.
- [7] S. Fortunato, Community detection in graphs, Physics Reports, 486 (2010), pp. 75-174.
- [8] S. Joseph, V. Lid and D. D. Suthers, Transcendent Communities, in C. Chinn, G. Erkens and S. Puntambekar, eds., The Computer Supported Collaborative Learning (CSCL) Conference 2007, International Society of the Learning Sciences, New Brunswick, 2007, pp. 317-319.
- [9] J. L. Lemke, Across the scales of time: Artifacts, activities, and meanings in ecosocial systems, Mind, Culture & Activity, 7 (2000), pp. 273-290.
- [10] S. Martin, W. M. Brown, R. Klavans and K. Boyack, OpenOrd: An Open-Source Toolbox for Large Graph Layout, SPIE Conference on Visualization and Data Analysis (VDA), 2011.
- [11] P. R. Monge and N. S. Contractor, Theories of Communication Networks, Oxford University Press, Oxford, 2003.
- [12] M. Newman, Networks: An Introduction, Oxford University Press, 2010.
- [13] J. Raskin, The Humane Interface: New Directions for Designing Interactive Systems, Addison Wesley, Reading, Mass., 2000.
- [14] B. Rogoff, Observing sociocultural activity on three planes: Participatory appropriation, guided participation, and apprenticeship., in J. V. Wertsch, P. D. Rio and A. Alvarez, eds., Sociocultural Studies of Mind, Cambridge University Press, New York, 1995, pp. 139-164.
- [15] D. Rosen and M. Corbit, Social network analysis in virtual environments, Proceedings of the 20th ACM conference on Hypertext and hypermedia (HT '09), ACM, New York, NY, 2009, pp. 317-322.
- [16] M. Schlager, J. Fusco and P. Schank, Evolution of an Online Education Community of Practice, in K. Renninger and W. Shumar, eds., Building Virtual Communities, Cambridge University Press, Cambridge, 2002, pp. 129-158.
- [17] D. D. Suthers, Technology affordances for intersubjective meaning-making: A research agenda for CSCL, International Journal of Computer Supported Collaborative Learning, 1 (2006), pp. 315-337.
- [18] D. D. Suthers and C. Desiato, Exposing chat features through analysis of uptake between contributions, Proceedings of the Hawaii International Conference on the System Sciences (HICSS-45), January 4-7, 2012, Grand Wailea, Maui, Hawai'i (CD-ROM), Institute of Electrical and Electronics Engineers, Inc. (IEEE), New Brunswick, 2012.
- [19] D. D. Suthers, N. Dwyer, R. Medina and R. Vatrappu, A framework for conceptualizing, representing, and analyzing distributed interaction, International Journal of Computer Supported Collaborative Learning, 5 (2010), pp. 5-42.
- [20] D. D. Suthers, J. Fusco, P. Schank, K.-H. Chu and M. Schlager, Discovery of community structures in a heterogeneous professional online network, Proceedings of the Hawaii International Conference on the System Sciences (HICSS-46), January 7-10, 2013, Grand Wailea, Maui, Hawai'i (CD-ROM), Institute of Electrical and Electronics Engineers, Inc. (IEEE), New Brunswick, 2013.
- [21] D. D. Suthers and D. Rosen, A unified framework for multi-level analysis of distributed learning Proceedings of the First International Conference on Learning Analytics & Knowledge, Banff, Alberta, February 27-March 1, 2011, 2011.
- [22] S. Trausan-Matu and T. Rebedea, A polyphonic model and system for inter-animation analysis in chat conversations with multiple participants, in A. Gelbukh, ed., Computational Linguistics and Intelligent Text Processing, Springer, Berlin, 2010, pp. 354-363.
- [23] S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications, Cambridge University Press, New York, 1994.