INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11
15 - 18 AUGUST 2011, TECHNICAL UNIVERSITY OF DENMARK

A PROTOCOL FOR CONNECTIVE COMPLEXITY TRACKING IN THE ENGINEERING DESIGN PROCESS

James Mathieson, Michael Miller, and Joshua Summers
Clemson University, SC, USA

ABSTRACT

The evaluation of design processes is often conducted after the given project has been completed or as a case study on a single process. These two approaches each cannot be used to improve an ongoing process and require a great deal of time to generate statistically significant samples. Presented here is a protocol for tracking the interconnection of design process elements as a mixed temporal hypergraph network which may evolve in real time. The protocol utilizes email and limited human reporting data to develop the time-stamped connections of the network. At any time, this network or a filtered subset of it may be subjected to an analysis of graph and network properties. The response of these properties may then be correlated to either events or performance metrics. Here, this approach is applied to emails generated in the course of an undergraduate mechanical engineering senior design project. This application demonstrates an ability to identify member roles, work schedules, and project phase changes from graph and network properties.

Keywords: Complexity, Design Process, Representations

1 COMPLEXITY IN ENGINEERING

Complexity is an aspect of engineering design which is not easily determined. Complexity is often addressed with the principles such as "designs should be simple" and "the design process should be simple" [1,2]. While these principles are valid, they do not allow designers to effectively quantify the complexity of a given design or course of action for comparison and decision-making purposes [3]. Complexity is too often determined by means of a subjective assessment by design engineers and project leaders. This subjective assessment introduces ambiguity to the process and decision reasoning. In order to alleviate such problems, a solid empirical basis for capturing and assessing project complexity is sought. The purpose of such a basis is to enable the mapping of complexity measurements to events and performance metrics in a predictive manner [4,5]. This, in turn, opens the possibility that designers may choose their actions along empirical projections of their implications. However, the work presented here addresses only the first stage of this process: the formalization of connective complexity tracking and identifying basic correlation.

Complexity is defined here in terms of systems [6,7,8]. A system is a set of interrelated elements which, through these relations, manifests a behavior which the individual elements do not exhibit independently [9]. A system can be made up of elements which can include anything capable of interaction, including molecules, machine elements, and people [10]. Our foundation for complexity is the human attempt to quantify our ability to understand these elements and interrelations which are counterintuitive [11]. Therefore, we can define complexity as the effort required to understand the properties of a given system [12,9].

There are several major classes in the measurement of system complexity: size, interconnectivity, centrality, and decomposability. The most common measurement used in formal complexity determination is size [13,8,7]. It is clear that when more elements are added to a system it becomes more complex [14]. However, system size does not capture the unique arrangement of the system which allows it to manifest higher order behavior. Interconnectivity, centrality, and decomposability capture this by analyzing how system elements are connected to each other [8,15].

2 PROJECT COMPLEXITY TRACKING

The first step in determining an objective means of tracking complexity is to further develop and refine a protocol for data collection. Here, we derive a new protocol in part from a previously developed protocol [16]. The existing protocol was developed for the purposes of tracking design complexity with respect to work hours addressing the requirements, functions, and components of the design, the coupling of elements within the same domain, and coupling of elements across domains. However, it has been observed during further research that additional complexity metrics exist, and that the complexity of both the design artifact and the design process impact outcomes. Therefore, the data collection protocol has been refined to collect connective data regarding both the artifact and the process. The new protocol was developed with the goal of being as minimally intrusive as possible, using documents and communications which were produced throughout the course of a senior design project at Clemson University.

One example of what was collected for complexity tracking, illustrated in Figure 1, would be a requirements document, the other documents which are related to that version of the requirements, who worked on making the change from the previous version, and the start and stop timestamps. This involves tracking who sent emails to whom and who attended meetings in regards to each update to the requirements. The same scheme is applied to CAD models, reports, design tools, and any other documents which were produced during the course of the project.

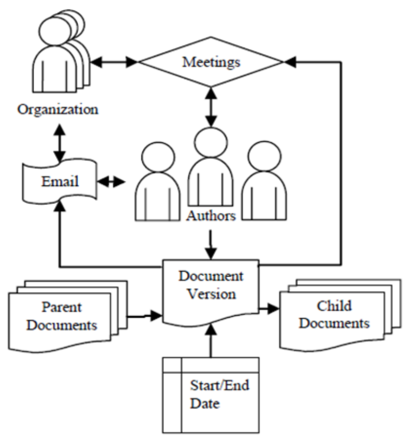
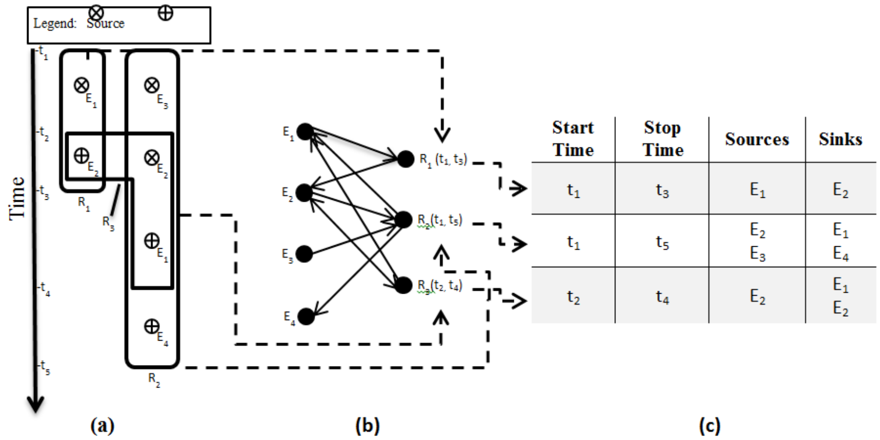


Figure 1: Design Complexity Tracking Scheme

3 DEVELOPED PROTOCOL

To extract connective information from the design process in an efficient manner, a protocol must be established which gathers complete data in a minimally intrusive manner. Email records are used as a primary data source. By carbon copying all transactions to a single email address, information regarding discussed topics, developed documents, and meetings may be extracted from the design group without requiring time consuming verbal reporting by participants. Subsequently, these email records are each made available to the analyzing system immediately upon their generation. One of the objectives in this protocol is for the required information to be within the identification capabilities of existing software tools.

The email records, once captured, are then processed to develop a graph of relationships within the project. This graph takes the conceptual form of a mixed temporal hypergraph, consisting of time stamped edges of multiple source (such as e-mail sender) and sink elements (such as e-mail recipient). Graphically, this may be represented as a mixed bi-partite graph where the secondary set of nodes is the edges of the hypergraph as shown in Figure 2.



Start Time	Stop Time	Sources	Sinks
t_1	t_3	E_1	E_2
t_1	t_5	E_2 E_3	E_1 E_4
t_2	t_4	E_2	E_1 E_2

Figure 2: Conversion from (a) Temporal Hypergraph to (b) Mixed Bipartite Graph to (c) Relationship Class

To enable the machine processing of the connective graph, the information is coded into a relationship data class. This data class contains a start and stop timestamp for the interaction being modeled, a set of source elements which provide information into the interaction, and a set of sink elements which receive information from the interaction. The sets of source and sink elements are not mutually exclusive, thus enabling a single edge to both transmit information from and to one element while only sending to or receiving from another.

Other derived classes of relationships include communication, topic, document, meeting, and organization. The elements of this graph are similarly divided into distinct classes. The classes addressed here are people, documents, emails, meetings, and topics. These derived classes are not trivial to the construction of the graph. Each class implies a specific set of connection rules to model the given instance of information flow and the distinct aspects of that exchange. We define these rules in the context of each of the relationship classes and how the different element classes are applied to them.

3.1 Organization

The organization of the process in question is of a key concern to how information flows. As such, the graph must capture how management duties are delegated and how individuals are divided in groups and hierarchies. This is usually explicitly defined in communications which initiate a given process. Figure 3 provides an example of how an organizational structure is translated into the graph. The organization given here is an undergraduate engineering senior design project. In each of the relationships, the start and stop time is coded as the start and stop dates of the project. However, if an organizational structure is altered mid-process, this should be captured by specifying the dates for which any given organization relationship was in effect. In coding the elements for each relationship, the supervising members are considered sources and supervised members considered sinks. This captures the path by which instruction may pass through the organization. It should be noted that this allows a broad range of organization structures to be coded including hierarchies, supervised groups, and autonomous teams.

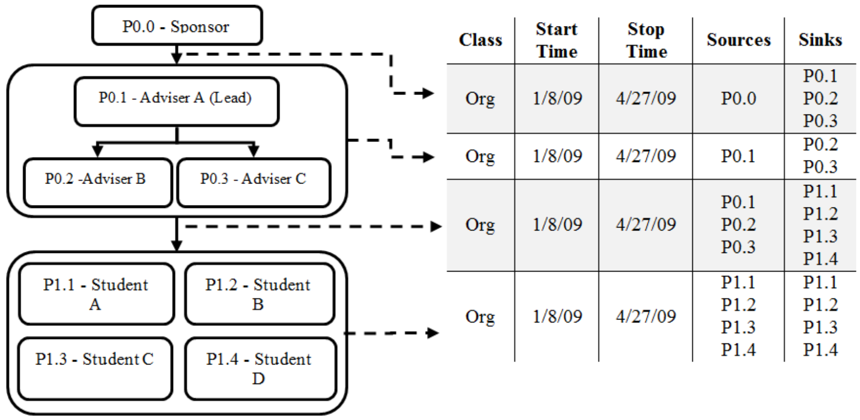


Figure 3: Example Organizational Structure Translation

3.2 Communication

Communication represents the bulk of available data on information flow. However, it is also a higher level class which defines the directionality of information flow and content but is not used to define the structure of the information. As such, communication relationships take two forms. The first form is the definition of the given communication instance, while the second is a definition of any self-contained information packets included in the communication, such as attached documents. Both of these may be illustrated with an email, as in Figure 4.

Here, one individual emails two others discussing three topics and attaching a document. This is translated into two relationships, one defining the email and the second defining the attachment. The time stamps for both relationships are identical, being the sent time on the email record for both the start and stop times. This is accurate given the instant nature of email and communication in general.

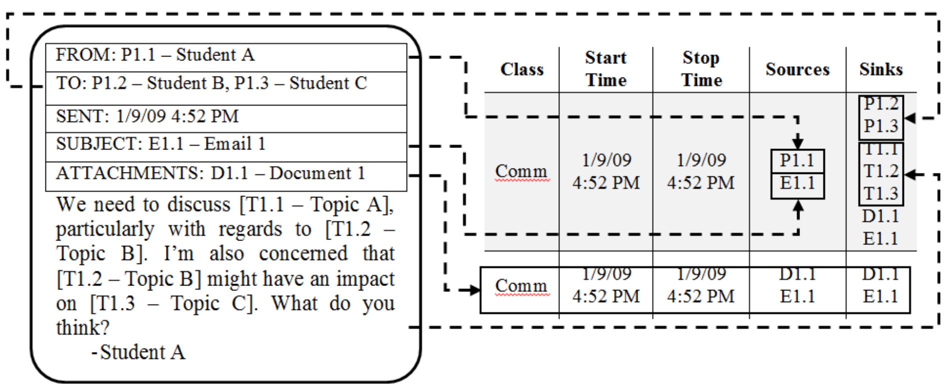


Figure 4: Translation of Email into Communication Relationships

The sender is placed in the source list as they are the source of this information. Likewise, the recipients are placed in the sink list. The topics discussed in the body of the email are placed in the sink list as this email has generated new information regarding these topics. The same is true of listing the attachment in the initial definition sink list. This captures the fact that the sender did something to this document. The attachment relationship captures the document being contained within the given email.

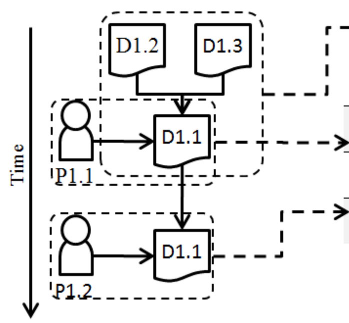
The email itself appears on both sides of all relationships involving it as the email is both physical data and the instance captured and is thus both acting and being acted upon in all relationships. In the case of a reply or forwarded communications, the referenced messages are included in the definition source list as these provide information into the current instance.

3.3 Document

Similar to communications, documents may have two different relationships forms based on information source. The first of these is authorship where information is added to the document by an individual, while the other is implied information through references, source documents, and previous document versions. For example, consider a document discussing the three topics addressed in the email from Figure 4. This document was co-authored by Students A & B at different times and references two other documents.

Authorship is defined in a manner very similar to communication. That is, the author and document are placed in the source list while the topics written about are placed in the sink list along with a repetition of the document. This is for the same reasons as with communication, the author has added new information to these topics. This is shown in coded form in Figure 5. The start and stop timestamps are derived from the digital file properties which record creation and modification times. In this case, Student A (P1.1) created the document at 1 pm and worked on it until 3:30 pm, writing information on topics T1.1 and T1.2. The document was then transmitted to Students B and C (P1.2 and P1.3) in the email from Figure 4. Student B (P1.2) then worked on the document from 5 pm to 8:30 pm, adding information regarding topics T1.2 and T1.3.

The third row in Figure 5 defines the existence of referenced documents within document. The referenced documents, D1.2 and D1.3 are listed as information sources while the current document is listed as a sink. This is consistent with the nature of referencing. The time stamps applied are the time span for which that source was used in generating the document. In this case, both are used over the course of writing the document.



Class	Start Time	Stop Time	Sources	Sinks
Doc	1/9/09 1:00 PM	1/9/09 3:30 PM	P1.1 D1.1	T1.1 T1.2 D1.1
Doc	1/9/09 5:00 PM	1/9/09 8:30 PM	P1.2 D1.1	T1.2 T1.3 D1.1
Doc	1/9/09 1:00 PM	1/9/09 8:30 PM	D1.2 D1.3	D1.1

Figure 5: Example of Document Authorship and Reference Relationships

3.4 Meeting

The recording of a meeting instance is treated as a special case of communication. Depending on the format of the meeting, you may have multiple senders and recipients. For example, a presentation by the student team from Figure 3 to the faculty advisory committee would be best captured with all of the students present as sources and all of the advisers present as sinks. Just as with communication and documents, the topics addressed are also listed as sinks. However, if the meeting recorded is a discussion amongst the student team, the members in attendance would be listed as both sources and sinks. Further, meetings containing both presentation and discussion phases should have these recorded as two distinct relationships. An example of this is given in Figure 6 where a single meeting consisting of a 20 minute presentation and a 10 minute discussion. While the topics of the presentation can be extracted from the presentation and executive summary documents, the content of any discussion requires the manual recording and reporting of meeting minutes.

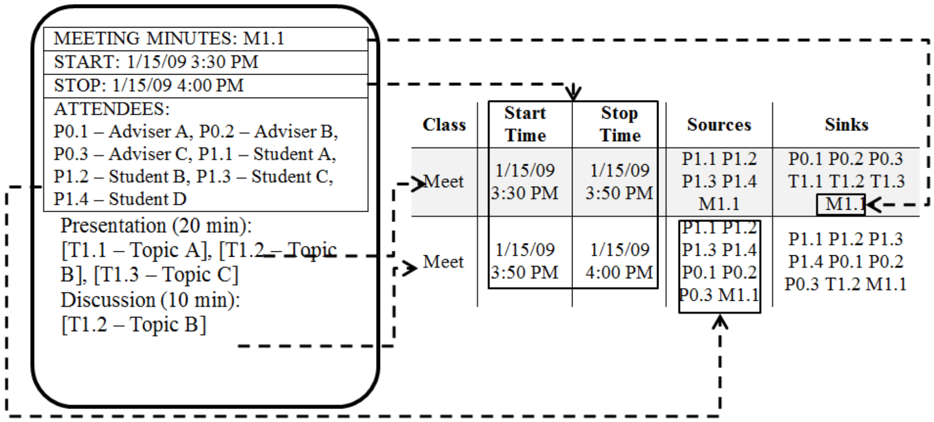


Figure 6: Example of Presentation and Discussion Meeting Relationships

3.5 Topic

The topic relationship class is used to detail the structure of information within the instance containers of communications, documents, and meetings. Particularly, this captures how topics are grouped together within the instance. As the appearance of two or more topics together within the instance has an effect on all topics in the grouping simultaneously, all topic relationships are modeled as bi-directional with identical source and sink lists. To illustrate how topic groupings in raw data are translated into topic relationships, we revisit the email presented in Figure 4.

Figure 7 presents the translation of the email body into two topic relationships. The first sentence contains references to Topic A and Topic B, forming one grouping, while the second sentence contains Topics B and C, forming the other. It should be noted that the topic relationships take on the same time stamp information as the instance definition. This is done in the same manner for documents and meeting minutes. The level of granularity that can be achieved in a practical amount of time is dependent on if the identification of topics and groupings in raw text is performed manually or computationally by linguistics software.

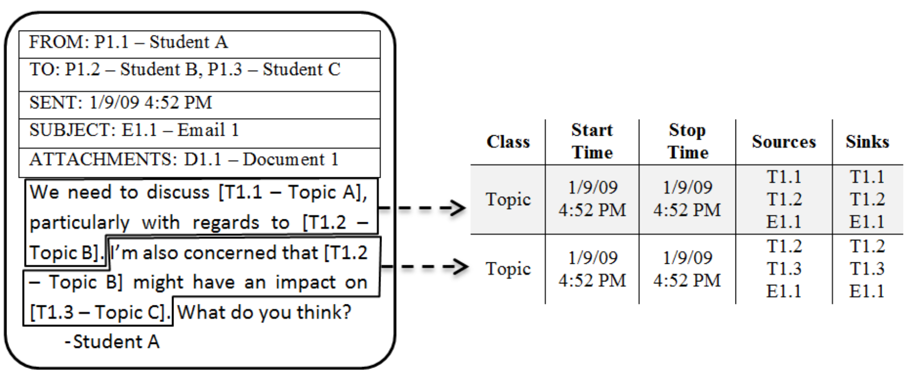


Figure 7: Translation of Text into Topic Relationships

4 APPLICATION CASE

The developed protocol was applied to data captured during the first month of an undergraduate engineering senior design project at Clemson University. The resulting coded graph data was then computationally analyzed for graph and network metrics previously identified. [15,17] These metrics include measures of size, interconnection, centrality, and decomposition. Many of these metrics are sum, maximum, mean, and density values derived from matrix or vector results on the individual elements.

Table 1 presents the results of this analysis as applied to the overall graph as well as sub-graphs filtered to relationships active within each week. Weekly values after the initial week are annotated with a change indicator for increase (+), decrease (-), or no change (=) from the previous week. Metrics of interest for identifying phenomena within the design process are those which do not present a continuous upward, downward, or constant change from week to week. Examples identified in this table are mean shortest path, mean flow rate, betweenness density, mean clustering coefficient, and mean core numbers. The total number of metrics discussed here is not exhaustive due to space limitations in this forum. We can explore the capture of process phenomena by looking at how these metrics vary when the graph is filtered to a 24 hour active view and rolled forward one hour at a time.

Table 1: Application Case Analysis Results

Class	Size				Interconnection							
Type	Dimentional		Connective		Shortest Path				Flow Rate			
Metric	Ele.	Rel.	DOF	Conn.	Σ	Max	Mean	Density	Σ	Max	Mean	Density
Week 1	40	52	333	233	3380	4	2.17	0.042	6818	24	4.26	0.0819
Week 2	69 (+)	97 (+)	775 (+)	461 (+)	10078 (+)	4 (=)	2.15 (-)	0.022 (-)	23150 (+)	29 (+)	4.86 (+)	0.0501 (-)
Week 3	76 (+)	130 (+)	783 (+)	532 (+)	12327 (+)	4 (=)	2.16 (+)	0.017 (-)	26051 (+)	45 (+)	4.51 (-)	0.0347 (-)
Week 4	110 (+)	168 (+)	1335 (+)	791 (+)	26924 (+)	4 (=)	2.25 (+)	0.013 (-)	59680 (+)	57 (+)	4.93 (+)	0.0294 (-)
Cumulative	211	439	3165	1975	103034	4	2.33	0.005	224651	104	5.05	0.0115

Class	Centrality							
Type	Betweenness				Clustering Coeff.			
Metric	Σ	Max	Mean	Density	Σ	Max	Mean	Density
Week 1	1820	434	46	0.875	21	0.8	0.53	0.010
Week 2	5590 (+)	858.2 (+)	81 (+)	0.835 (-)	39 (+)	1 (+)	0.6 (+)	0.006 (-)
Week 3	6998 (+)	1300 (+)	92 (+)	0.708 (-)	40 (+)	1 (+)	0.52 (-)	0.004 (-)
Week 4	15370 (+)	2134 (+)	140 (+)	0.832 (+)	56 (+)	1 (=)	0.51 (-)	0.003 (-)
Cumulative	58934	9366	279	0.636	113	0.9	0.54	0.001

Class	Decomposition								
Type	Ameri-Summers	Core Numbers							
Dir		In				Out			
Metric		Σ	Max	Mean	Density	Σ	Max	Mean	Density
Week 1	244	168	5	4.2	0.08	187	5	4.7	0.09
Week 2	977 (+)	368 (+)	7 (+)	5.3 (+)	0.05 (-)	368 (+)	6 (+)	5.3 (+)	0.05 (-)
Week 3	1278 (+)	429 (+)	8 (+)	5.6 (+)	0.04 (-)	407 (+)	7 (+)	5.4 (+)	0.04 (-)
Week 4	1428 (+)	589 (+)	7 (-)	5.4 (-)	0.03 (-)	665 (+)	8 (+)	6 (+)	0.04 (-)
Cumulative	6498	1257	10	6.0	0.01	1392	9	6.6	0.02

The first metric we look at in the 24 hour rolling view is mean shortest path length, shown in Figure 8. This metric serves as a proxy for general activity level as it measures the number of edges that must be traversed on average to go from any node to any other node. However, when a connection cannot be made, this value evaluates to zero in taking the mean. Therefore, a mean value less than one is positively indicative of a disconnected graph. These regions of disconnection, below the dashed line, line up closely with the shaded periods indicating weekends. This demonstrates the ability of the graph to identify the general work pattern of the project members.

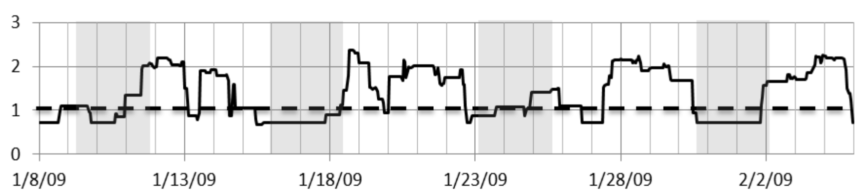


Figure 8: Mean Shortest Path Length, 24 Hour View, 1 Hour Step

Another metric which demonstrates an ability to identify key moments is betweenness density, shown in Figure 9. Betweenness, a centrality metric, measures the number of shortest paths which pass through any given element. This makes betweenness a good proxy for element importance. Betweenness density measures the average level of betweenness generated by each relationship. The spike in the graph represents a point at which the information generated was highly impactful. This point also corresponds to the beginning of the first week in which design decisions were made, the shift between project organization and generative design phases. It would be expected that similar spikes would also be observed at other major phase shifts in the process.

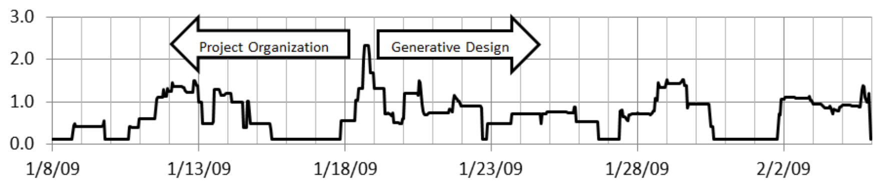


Figure 9: Betweenness Density, 24 Hour View, 1 Hour Step

An additional application of betweenness can be seen when the individual values for each element are taken for each of the 24 hour rolling analysis points. Figure 10 shows this for the four members of the student design group. Again, betweenness is taken as a proxy for importance. It can be seen that designer P1.2 starts out initially with a higher value than the other three during the first week. However, this promptly changes in the second week as P1.1 and P1.2 garner similar values, possibly indicating a power struggle for group leadership. By the third and fourth weeks, a hierarchy appears to form with group member values taking on distinct levels at critical points. However, it is also curious that designer P1.4 appears to maintain a more consistent level of importance though each work week than the other three. These observations demonstrate the ability to analyze group dynamics from the collected data.

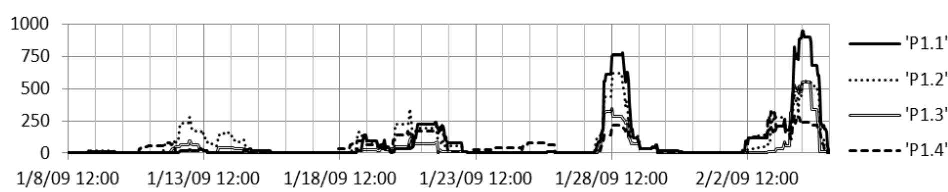


Figure 10: Designer Betweenness Values, 24 Hour View, 1 Hour Step

5 CONCLUSION

In this paper, we have presented a protocol developed for the recording and capture of connective information in a design process and an application case of this protocol. The protocol developed here captures information regarding organizational structure, communications, documents, meetings, and topics. This information is captured primarily through emails generated in the course of the analyzed project and supplemented by the reporting of meeting minutes. Once collected, these raw data sources are converted into a mixed temporal hypergraph, recorded in relationship data classes.

The application case presented here is analyzed to extract graph and network properties for various time windows. This analysis demonstrates potential abilities to identify work patterns, project phase changes, and group dynamics. Metrics worthy of further study for identifying project phenomena are also identified.

In the future, the possibility exists to extend this protocol into automated processes. Particularly, the use of linguistics software to identify the structure of topic relationships within raw text is a key point for extension. Another is further analysis of the application case to identify metrics indicative of phenomena and performance measurements of interest.

REFERENCES

[1] N.P. Suh, "Axiomatic Design Theory for Systems," *Research in Engineering Design*, vol. 10, no. 4, pp. 189-209, 1998.

[2] G Pahl and W Beitz, *Engineering Design: A Systematic Approach*, 3rd ed. London, England: Springer, 2007.

[3] G.A. Hazelrigg, *Systems Engineering: An Approach to Information-Based Design*, 1st ed. New Jersey, U.S.A.: Prentise Hall Inc, 1996.

[4] J. Mathieson, C. Sen, A. Shanthakuma, and J. Summers, "Complexity as a Surrogate Mapping between Function Models and Product Costs," in *Proceedings of the ASME IDETC/CIE 2011*, Washington, DC, USA, 2011, p. [To Be Submitted].

[5] J.L. Mathieson, B. Wallace, and J.D. Summers, "Assembly Time Modeling Through Connective Complexity Metrics," in *International Conference on Manufacturing Automation*, Hong Kong, China, 2010, p. 22.

[6] James Mathieson and Joshua D Summers, "Relational DSMs in Connectivity Compexity Measurement," , Greenville, SC, 2009.

[7] James L Mathieson, Chiradeep Sen, and Joshua D Summers, "Information Generation in the Design Process," in *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, San Diego, 2009, pp. 1-10.

[8] Joshua D Summers and Jami J Shah, "Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability," *Journal of Mechanical Design*, vol. 132, no. 2, p. 021004 (11 Pages), February 2010.

[9] H. A. Simon, *The Sciences of the Artificial*. Cambridge, U.S.A.: The MIT Press, 1981.

[10] I Barclay and Z Dann, "New-product-development performance evaluation: a product-compexity-based methodology," in *IEE Proceedings in Scientific Measurement Technology*, 2000, pp. 41-55.

[11] J Casti, *Connectivity, Complexity, and Catastrophe in Large-Scale Systems*, International Series on Applied Systems Analysis ed. New York, U.S.A.: John Wiley & Sons, 1979.

[12] L Lofgren, "Complexity of Descriptions of Systems," *International Journal of General Systems*, vol. 3, pp. 197-214, 1977.

[13] F Ameri, G.M. Mocko, and M. Porter, "Engineering design complexity: an investigation of methods and measures," *Research in Engineering Design*, vol. 19, no. 2-3, pp. 161-179, November 2008.

[14] Anthony J Masalonis, Michael B Callaham, and Craig R Wanke, "Dynamic Density and Complexity Metrics for Realtime Traffic Flow Management," Center for Advanced Aviation System Development (CAASD), The MITRE Corporation, McLean, VA, 2003.

[15] James L Mathieson and Joshua D Summers, ""Complexity Metrics For Directional Node-Link System Representations: Theory and Applications," in *ASME IDETC/CIE 2010*, Montreal, 2010, pp. IDETC2010-28561.

[16] I Pramanick and H Ali, "Analysis and experiments for a parallel solution to the all pairs shortest path problem," in *IEEE International Symposium on Circuits and Systems*, New York, 1994, pp. 479-82.

[17] James Mathieson, "CEDAR Complexity Analysis Capabilities," Clemson University, Clemson, Technical Report 2010.