Frame-Based Models of Communities and Their History

Robert B. Allen

Research Center for Knowledge Communities, University of Tsukuba, Japan
rba@boballen.info

Abstract. Previous models of communities and their history have focused on the entities in those communities such as their locations and people. We introduce models which incorporate behaviors and processes. We propose that approaches based on object-oriented modeling are particularly useful. Specifically, we explore the feasibility of developing object-oriented models which employ linguistic frames adapted from the FrameNet corpus. We apply these models to relatively straightforward and self-contained historical scenarios. We implement the models in Java and analyze some of advantages and challenges in that approach. Historical newspapers are particularly rich sources of natural language descriptions about communities but there are many sources of non-linguistic information about communities which may also be incorporated. We consider the possibilities of developing more coherent models of communities based on modeling processes, partonomies, systems, and situations. Finally, we consider enabling greater interactivity with the structured models and alternative architectures for the models.

Keywords: Behavior, Descriptive Modeling, Events, Functionality, FrameNet, Indexing, Information Organization, Java, Object-Oriented Modeling, Processes, Social Modeling, Digital Humanities.

1 Modeling History and Communities

History has rich sources of evidence but they are challenging to organize and access. In an approach we describe as model-oriented information organization, we propose that explicit behavior-based modeling could improve indexing and be more effective than traditional descriptions of historical documents, artifacts, or communities. Systematic models used in software engineering are particularly appealing because many tools have been developed to implement and interact with them.

We focus on modeling communities Yet even because they are relatively self-contained. Even a relatively small town will have hundreds of residents and each of them is involved in myriad activities. Moreover, modeling history meaningfully requires a great amount of detail and the specification of a great deal of context. Fortunately, digitized historical newspapers and other historical records contain a vast amount of detailed information relating to communities.

Some of our earlier work explored supporting access to massive collections of digitized historical newspapers and the challenge of organizing that newspaper content. Because there are few standardized structures in historical newspapers, it is difficult even to consistently identify newspaper articles, let alone index them. As an

A. Nadamoto et al. (Eds.): SocInfo 2013 Workshops, LNCS 8359, pp. 110–119, 2014.

alternative to applying simple metadata labels for the articles, we propose organizing the material by developing models of the content [1, 8].

Explicit modeling could support access by improving the accuracy of human retrieval from historical newspapers and in turn greater accuracy should drive refinement of the models. Similarly, the models can improve the accuracy of the text extraction and, in turn, text extraction can populate the models.

There are different levels of detail and coherence in community models. We developed a prototype interactive directory of the entities in a Midwestern US town for 1899 to 1900 [6]. That directory combines data about people, businesses, and locations obtained from census records, historical city directories, and digitized historical newspapers. However, the interactive directory does not describe either typical processes or the details of specific events. Here we explore implementing a richer community model which includes behaviors specified with Java classes based on text descriptions. We also provide a conceptual framework and some additional world knowledge for structuring models of the entities and events related to the community.

2 Entities, Frames, and World Knowledge

Frames are flexible structures employed in artificial intelligence and software engineering for describing complex objects. While the entities we describe are often far more complex than those in most software engineering applications, frames offer a fruitful approach to describing them. For modeling textual descriptions we employ FrameNet [10, 12] which uses frames to capture concepts expressed in natural language. FrameNet was based on a cognitive model in which the linguistic frames were determined by typical usage. We use the collection of frames as a resource for information organization. Importantly, FrameNet includes frames for verbs. These describe how the frame values are changed as a result of an event [2, 8].

We modeled the attributes of individuals and organizations. Our entities are often variations of common structures; people have predictable biological structures and somewhat more variable, but still often predictable, social structures. We combined entities with a conceptual model knowledgebase and world knowledge beyond what was in the newspapers we sampled. The knowledgebase classified entities and facilitated inheritance. The world knowledge included both details about instances (e.g., street names, the names of businesses, the names and roles of government officials (cf. [3]), and the names of children in the 4^{th} grade class in the local school), and about processes (e.g., local laws, rituals, train schedules, etc.). Some of the world knowledge overlaps with the information in an interactive city directory [6].

We focus on description rather than inference, because of the complexity of making automated inferences. We allow low-level inferences for actions which are implied but omitted in written language, but we particularly avoid automated inferences about motives, trends, or theories because of the difficulties of large scale AI systems in doing so, rather we emphasize successful large-scale descriptive systems such as UMLS.

3 Frame-like Models with Java

Many software and knowledge engineering tools have been developed to model specific activities in communities, modeling a limited aspect of the entire community. For greater robustness and support, we adopted a general-purpose programming language and coded the natural-language-like structures needed. We selected an object-oriented programming language so that we could also model the behavior of entities and relationships with other entities. We adopted Java. Because sequences of events unfold across time, we allow the representation to unfold. That is, we run the model to determine the states rather than try to statically spatialize the context for each event.

Natural language is the basis of the modeling in this paper, in that we apply limited preprocessing of texts to show a coherent set of actions. In doing so, we remain as close to the source text as possible so as to remain as faithful to the history (as it was written) as possible. It is useful to think of many concepts as inheriting and behaviors from broader classes. A person is a physical object with mass and shape as well as an animal which breaths, moves, and eats. Similarly, the shops in the community such as the bakery and dry goods shop share attributes as retail businesses. We defined entity classes for geographic features, governmental units, businesses, and individuals. Once the classes were defined, we instantiated specific entities. Those instantiated entities were recorded in an EntityLog array (Fig. 1).

```
// main:
    entityLog[entityLogCnt] = new EntityLog();
    entityLog[entityLogCnt].typeCnt=person_cnt;
    entityLog[entityLogCnt].type="Person";
    entityLog[entityLogCnt].instanceName="FatherHennipin";
    entityLogcnt++;
```

Fig. 1. Entry in the EntityLog to record that a Person entity has been instantiated. There is a similar log for events.

Verb frames were implemented as Java classes with methods. A specific event was described by instantiating the verb frame class and passing entities to it as arguments. As a result of the methods, the values of one or more entity attributes are changed (cf., [2]). Fig. 2, shows a Java class implementation of the verb frame for releasing a captive. In this case, the state of a variable indicating whether the captive person is a prisoner is set to *false*. We recognize this is a simplistic quasi-attribute and we discuss alternatives in [7].

```
class V_Release {
        public V_Release(Person Captor, Person Captive){
            Captive.isPrisoner=false;}
```

Fig. 2. Java class for "release" frame

4 Some Solutions and Some Challenges

4.1 Sample Encoded Text Passage

We focused on encyclopedia-level descriptions because they generally use straightforward language with relatively little metaphor. Such texts still include unstated gaps and assumptions (see [7]) but these seemed fairly uncontroversial and we used our best judgment in coding them. As an example text, we selected a passage from Wikipedia about the early history of Minneapolis, Minnesota.[1] We instantiated Person entities for the explorers Daniel Greysolon, Father Hennipin, and two unnamed individuals. [2] Hennipin and the unnamed individuals formed an AdHocGroup which had been captured and was being held by the Dakota Tribe. The Dakotas were instantiated as a FormalGroup. In a sequence of events described in the Wikipedia entry Greysolon requested the Dakotas to release Hennipin and the others. After being released, Hennipin "discovers" and names St. Anthony Falls. A later segment of the Wikipedia passage briefly describes the development of the area around the Falls into a mill district.

4.2 Model Elements: Partonomies, Processes, and Systems

Because the entities we consider are highly complex and because they often interact in predictable ways we develop model elements. The entities complex entities can be separated into parts and can be described with partonomies. Geographic locations are parts of other locations. As noted in the previous section, many activities involve either ad hoc groups or formal groups. Grouping is a partonomic relationship [14]. For AdHocGroups, we developed a transient entity which contains an array of instances. FormalGroups also have an array of members but the group is also a formal entity in itself and has its own attributes.

Many events are part of an established sequence and that sequence itself can be treated as an entity. In earlier work, we suggested that predefined or expected sequences could be modeled with processes. Several types of processes can be distinguished. A process may simply iterate through an ordered list. Or, there may be Petri-Net-like sequences in which the order can be relatively unimportant. Moreover, these processes may be enhanced with constraints on the entities involved. For instance, a Recipe specifies both quantities and qualities of each of its ingredients (e.g., "take two large eggs").

Systems and Situations combine entities and processes. The entities are often partonomic and a system may itself be an entity in which the processes describing the relationship of the parts. Systems include organizations and machines but they can also include static relationships such as the walls of a house holding up the roof. A Situation is an interaction of specific entities and it may be either in stasis or evolving.

---

1 `http://en.wikipedia.org/wiki/History_of_Minneapolis` retrieved on August 3, 2013.

2 Recently, the text of the Wikipedia entry has been modified to include the names of the un-named individuals.

We found it difficult to distinguish between classes and instances for complex entities. For instance, a given house may be an instance of the house class but instantiation of the high-level class does not specify all the details of that house (e.g., whether it has a separate dining room). We may need multi-leveled inheritance for complex object concepts and instances of those objects. Moreover, the typical parts of complex entities may change and considerable care is needed to distinguish between changes in the relationship of classes (the information structure) and changes in the instances.

4.3 Capturing the Nuance and Ambiguity of Natural Language

Language is highly nuanced. Consider the relationship of Baker, Bakery, and BakedGoods. The pattern of Occupation, Shop, and Product of Occupation is mirrored by Pharmacist, Pharmacy, and Pharmaceuticals and by Printer, PrintShop, and PrintedMaterials. However, it breaks down in other cases such as Librarian and Library where a service is provided rather than a product. The distinction between product-oriented organizations and service-oriented organizations is well understood in economics but those concepts have not been integrated with an overarching framework for communities.

There are nuances in the usage of even relatively a relatively simple term such as Baking. "To bake" may mean simply to heat to a high temperature, may refer to a particular phase of cooking, or may mean the preparation of an entire food item (e.g., "he baked some bread"). The simplest word sense of baking involves one state change. More complex forms are a sequence of actions which include the simple action of baking. This extension of a simple action into a flow of actions is related to but richer than the inheritance of the method because the embedded action is integrated with the other actions. Likewise, a Baker could be someone who bakes a single item or it could be a person who has the occupation of Baking; further that occupation is associated with many other concepts.

As another example of complexity in language, consider that a Library is often, but not always, a CulturalInstitution. Indeed, an entity may change classifications and we need to be able to describe it during the transition when it may have mixed attributes. For instance, a Library may initially belong to a group or individual and later be opened as a public CulturalInstitution.

Verbs will apply across entire classes of entities which may not be congruent with the established class hierarchies. In the example passage, for instance, a Person (Greysolon) made a request of the Formal Group (the Dakota Tribe). A request can be made by any Agent to any other Agent whether a Person or a Formal Group. However, because Person is part of the sentient organism hierarchy we cannot also make it part of an Agent hierarchy which includes Groups. We considered overloading constructors but using Java Implements for an Interface was simpler. Still, there are additional cases where a more flexible approach is needed. A Baker may have several different roles. The Baker may also be the Manager and the Owner of the Bakery. While multiple inheritance is generally discouraged in object-oriented design with languages such as Java, it seems difficult to avoid when the goal is modeling natural language usage.

Ultimately, we need standards to encourage consistency and a flexible modeling structure. Beyond nuance, natural language is also ambiguous. Even in textbook-level histories, there is considerable ambiguity. For instance, in reading the passage about Greysolon we assume that the Dakotas granted Greysolon's request to release Hennipin (rather than, say, Hennipin escaping) but that is not stated specifically. Such low-level inferences are largely uncontroversial and are accepted by most readers but should be noted.

There are also challenges in modeling generalizations. While we may know that most 10 year old American children spend their days at school it is obviously not always the case that a given child will be in school on a given day. We can state it as a generally true statement and as an expectation but, ideally, we would also identify the situations when it might not hold.

5 Broad Issues and Next Steps

A brief passage has been coded and the models we have implemented thus far are fairly basic. We need to expand the coverage and address more complex passages. Here, we consider some broader issues.

5.1 Modeling People: Mental Events, Knowledge, and Intentions

The entities describing People and Organizations are the most complex entity types in this implementation. Other parameters may have been described simplistically. For instance, when Father Hennipin was captured we noted his being held as change of state. But, should we have a separate state associated with every different verb? In [7] we suggest the need to explicitly model Situations. Some of the states of a person could be determined by the reference to the Situations they encounter.

Following the distinction made in [4, 5], in this initial study we focused on observable entities and events. Thus, we have excluded consideration of mental events and explanations including descriptions of people's reasoning. However, many explanations are based on attributions to a person's mental state. The attribution of mental events is so common for causal explanations that we need general strategies for modeling them. In any event, to the extent that we capture the gist of the words used to describe the mental states and the explanations based on descriptions of the mental states, we should be able to model them without being able to make generalizations.

Mental states which include beliefs, knowledge, and intentions are common in mental events such as inferences and decision making. Presumably a Baker both does the action of Baking and has knowledge of how to bake. But, it is unclear exactly what specific skills should be enumerated to cover knowledge of baking, let alone more complex skills such as knowledge of singing or of a foreign language. We may more readily represent abstractions which could include likely behaviors associated with having a given skill, instead of representing each person's knowledge in detail. For instance, knowledge of a foreign language is reflected in a person's ability to communicate in a country where that is the local language. Groups such as organizations, institutions and communities present analogous challenges. Groups have expectations, norms and knowledge. The challenge of representing these parameters is increased by the fact that groups may lack a consensus among their members on them.

Furthermore, human language users have widely differing understandings of concepts such as knowledge and norms. It remains an open question about the extent to which an information organization system should try to replicate mental models of individual human language users. In many cases, it may be better for the information organization system to approximate a typical language user. Essentially, the question is the extent to which the system should model a knowledgeable person or should it attempt to model every possible detail about every known entity and event. We need flexible multi-level representation management systems.

5.2 Object Instance Metadata and Other Annotations

While in the present study all of the text comes from one source, we envision weaving together text from multiple sources. Ideally, the source of all claims would be documented. Potentially, parallel versions of history could be woven together, especially if the versions were consistent. For instance, a user could decide whether to browse a simple biography of Daniel Greysolon or a more detailed one. If the descriptions are inconsistent, version management will be needed.

5.3 Explanations, Narrative, and Argumentation

As suggested by [5] discourse structures such as for narrative, explanation, and argumentation may be added over the base entity-event fabric. We noted the potential for narrative earlier in this paper. Explanations are often based on mental events (Section 5.1) to account for a person's actions. Argumentation may evaluate evidence or contrast explanations. Additional rhetorical structures may be defined to handle formal scholarly analysis.

5.4 Incorporating Richer and More Complex Representations

Here, we have focused on reproducing text descriptions with frames and with general knowledge about the entities described. However, merely modeling natural language statements is likely to be incomplete unless the text is unusually comprehensive. For example, much of the information in newspapers is not in the text of articles but in tables, pictures, and advertisements. To capture this information and to incorporate sources such as diaries, letters, and other types of records, we may need to augment the text representations with non-linguistic frames or other types of knowledge structures. A wide range of models which incorporate increasing levels of external knowledge are possible. Among those are architectural description logics, shopping models, and models of relevant norms and legal concepts. It would also be helpful to expand beyond communities and coordinate our models with some of the GIS-based approaches from area studies and "smart cities" projects from urban planning and eventually community models could be merged and extended to become national or international models. Eventually, standards bodies could determine preferred approaches. We need to develop a representation environment that allows representations to be manipulated and used at different resolutions for different applications.

5.5 Interacting with the Structured Models

Structured complex representations should support novel types of user interaction. For instance, [2] described how multiple trails of causally related events could be presented through an interactive timeline. The model-oriented approach many also support question answering. Moreover, we might generate narratives and other types of discourse from the EntityLogs and EventLogs. More ambitiously, we might also develop interactive historical environments. We might teach students about the French Revolution by allowing them to explore simulations of communities involved in the French Revolution.

User interfaces are also needed to help authors create a structured corpus. We envision a workbench for historians to organize evidence and pursue hypotheses in their research. Indeed, the structure provided by the models can facilitate the evaluation of evidence and inferences and should highlight gaps and inconsistencies.

5.6 Incorporating Knowledge of Cultures and Broad Social Theories

The models developed here are intended to apply equally across cultures though we have focused on the US Midwest because its communities are especially well documented in historical newspapers. Even in the short passage about Minneapolis described above, we considered the social organization of the Dakotas. Specific conceptual frameworks could be developed bottom up, but cultural and social modeling may also be implemented top-down (e.g., [11]). Moreover, while these theories can be applied to help modeling, eventually the large amount of data being modeled may in turn be useful for evaluating the theories.

5.7 Alternative Architectures

While Java was effective for implementing behavior-based frames, several of the issues we have encountered are due to Java's design. Among these issues is ambiguity about the distinction between classes and instances; there is a need for several inheritance paths depending on context and for readily extensible representation of classes. As we discuss in [7], future modeling efforts should explore other object-oriented languages such as Self and Slate [13].

While our frame-based approach remains faithful to the text of historical sources in a way that is important for historians, for other applications some liberties may be allowed. Much in the way that historical fiction takes liberties with historical evidence, simulations may fill in gaps and provide engaging interaction with the users. Indeed, some historians would say that the point of studying history is finding insights into plausible underlying processes. Simulations might help with that. Simulations could also be implemented as multi-agent interactions. Further, we believe that simulations could be the basis of history-oriented games.

6 Conclusion

We have proposed that frame-based object-oriented modeling is useful as an approach to historical descriptions of communities. We explore issues and implications of this proposal by modeling a community which is comparable to many communities described in digitized historical documents. Specifically, we have explored models with frames which track English natural language statements and extended them with conceptual and real world knowledge. We focused on natural language because such models may help with indexing and access to the historical records. However, a range of other related models is possible. We could develop models of social interactions not based on natural language.

While there is understandable caution in dealing with situations as complex as community models, there would be considerable value in having an overarching framework. Indeed, a focus on a broad "architecture of cognition" has led to considerable progress in understanding cognition and a similar broad impact may be expected for modeling communities.

Many challenges remain, but we believe there is a great value in highly structured representation of histories for interaction with that content. By and large, these are not algorithmic problems; rather, they are problems of representation, robustness, and scale. We urge a coordinated initiative of digital historians, social scientists, and information scientists to develop and apply broad conceptual models. We believe that these substantial challenges would yield to a concerted effort.

References

1. Allen, R.B.: Improving Access to Digitized Historical Newspapers with Text Mining, Coordinated Models, and Formative User Interface Design. In: IFLA International Newspaper Conference: Digital Preservation and Access to News and Views, New Delhi, pp. 54–59 (2010), `http://boballen.info/RBA/PAPERS/IFLA2010/IFLA2010proceedings.pdf`
2. Allen, R.B.: Visualization, Causation, and History. In: iConference (2011), `http://dx.doi.org/10.1145/1940761.1940835`
3. Allen, R.B.: Developing a Knowledge-base to Improve Interaction with Collections of Historical Newspapers. In: IFLA WLIC, San Juan, PR (2011), `http://conference.ifla.org/sites/default/files/files/papers/ifla77/188-allen-en.pdf`
4. Allen, R.B.: Model-Oriented Information Organization: Part 1, The Entity-Event Fabric. D-Lib Magazine (2013), doi:10.1045/july2013-allen-pt1

5. Allen, R.B.: Model-Oriented Information Organization: Part 2, Discourse Relationships. D-Lib Magazine (July 2013), doi:10.1045/july2013-allen-pt2
6. Allen, R.B.: Toward an Interactive Directory for Norfolk, Nebraska: 1899-1900. In: IFLA Newspaper and Genealogy Section Meeting, Singapore (2013), http://arxiv.org/abs/1308.5395
7. Allen, R.B.: Rich Semantic Modeling with Object-Oriented Approaches, ArXiv (in preparation)
8. Allen, R.B., Japzon, A., Achananuparp, P., Lee, K.-J.: A Framework for Text Processing and Supporting Access to Collections of Digitized Historical Newspapers. In: Smith, M.J., Salvendy, G. (eds.) HCII 2007. LNCS, vol. 4558, pp. 235–244. Springer, Heidelberg (2007)
9. Engeström, Y.: Innovative Learning in Work Teams: Analyzing Cycles of Knowledge Creation in Practice. In: Engestrom, Y., Miettinen, R., Punamaki, R. (eds.) Perspectives on Activity Theory: Learning in Doing: Social, Cognitive, and Computational Perspectives, pp. 377–404. Cambridge University Press, Cambridge (1999)
10. Fillmore, C.: Frame Semantics and the Nature of Language. Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech 280, 20–32 (1976), http://www.icsi.berkeley.edu/pubs/ai/framesemantics76.pdf
11. Mohr, J.W., White, H.C.: How to Model an Institution. Theory and Society 37, 485–512 (2008), http://www.soc.ucsb.edu/ct/pages/JWM/Papers/How%20to%20Model%20an%20In./How%20to%20Model%20an%20Institution.pdf
12. Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J.: FrameNet II: Extended Theory and Practice (2010), https://framenet.icsi.berkeley.edu/fndrupal/the_book
13. Salzman, L., Aldrich, J.: Prototypes with Multiple Dispatch: An Expressive and Dynamic Object Model. In: Gao, X.-X. (ed.) ECOOP 2005. LNCS, vol. 3586, pp. 312–336. Springer, Heidelberg (2005)
14. Winston, M.E., Chaffin, R., Herrmann, D.: A Taxonomy of Part-Whole Relations. Cognitive Science 11, 417–444 (1987), doi:10.1207/s15516709cog1104_2