A Pattern for the Integration of Conceptual Models in Support of Multidisciplinary Efforts to Develop Software Simulation Tools

Joan Peckham [&*], Natacha Thomas *, Jean-Yves Hervé [#1]*, Ron Hutt *#, Angel Castro [*], Charles Collyer*, Elizete Fernandez*, Feng Han*, David Kurowski *, Colin Liebermann, María del C. Rey, Lisa Ricci, Katherine Wray*
The 3-D Group
University of Rhode Island
Kingston, RI 02881
&
Gabriel Santos *
Benigno E. Aguirre *
Disaster Research Center
University of Delaware
Newark, Delaware 19716

Abstract: Conceptual models have been developed for many domains and application types. Within each discipline, the mode and purpose of conveying the design is slightly different. For example, in the computing disciplines, the conceptual models are comprised of constructs that enable the domain expert and software engineer to represent an application with notations that permit understanding of the application domain and at the same time facilitate the mapping from this conceptualization to the code that implements it. Clear specification of the computational model along with the objects, structures, and rules is essential. In the engineering domain statistical and mathematical models are used to permit concise and precise representation of the application to enable accurate analysis of the characteristics of the objects being modeled as well as an analysis of their performance. In the social sciences models utilize diagrammatic and verbal notations and attempt to describe the processes that effectively explain and predict human individual and group behaviors. The constructs used are not necessarily tied to computational or data driven constructs. In this paper we describe a pattern that describes an effective means of operating within a multidisciplinary group to develop a model for a common software prototype development goal The purpose is to provide a model integration pattern or technique that assists multidisciplinary teams to develop clear, precise and correct software design. Our particular

---

[&] This researcher is partially supported on this project from NSF grant ***** and its REU supplement.

[*] This researcher is partially supported on this project by NSF REUgrant ******

[#] These researchers are partially supported by the URI 3-D Partnership ***

[1]

example is the development of a simulation tool to be used by social scientists and security experts to analyze the evacuation of buildings under emergency and non- emergency situations. The team is made up of engineers, computer scientists and social scientists.

Related Research: Architects and software engineers have used patterns to describe reoccurring problems and possible standard solutions for the design and management of buildings and software code. The first oft cited example of pattern definition is in the architectural domain as put forward by Christopher Alexander's (Alexander, et. al., 1977). In his book he points out that there are many patterns describing best practices and good designs that can be communicated to others for reuse. The benefit of these patterns is that once captured and described with a set of problems for which they provide solutions, future designers and managers need not "re- invent the wheel". They are efficient, elegant and pleasing solutions that can be reused by the architect. Computer scientists had already begun to archive standard algorithms, many of which are the foundations of modern computer science education (Knuth, 1971). Following Christopher's example for the description and archival of software architectures, Gamma and co- authors described a set of frequently used patterns that combined data structure as well as algorithmic information. Recently patterns for software development and management have also been identified, an example is (Coplien and Harrison, 2005). In this paper we describe a set of patterns that have emerged in our own multidisciplinary work. The goal is to develop a set of procedures and constructs that can help researchers employing different model types to communicate with each other and to integrate their conceptual results to a common software design.

The benefits of pattern identification and adoption include:

1. Saves time – **** talk about the conceptual ease of abstraction and naming of a particular approach****
2. Fewer errors - ****constructs used identify problems and solutions more precisely. Fewer mistakes made in communicating such and defining solutions***

The Building Evacuation Project The goal of our project is to develop a model and software tool that simulates the movement of pedestrians through TF Greene Airport during emergency and non- emergency situations. The research team is multidisciplinary from civil engineering, manufacturing engineering, computer science, social psychology, and psychology. Building evacuation simulation tools have been developed with each project using different models of pedestrian movement. Most have focused upon the spatial aspects of the building and the explicit barriers for movement without much focus upon the psycho- sociological aspects of human motion. For example, SIMULEX (Thompson and Marchant, 1996) models individual pedestrians by assigning attributes to each pedestrian and then using these attributes to navigate them from their current location to exits in the building. This approach has been described a coordinate based. Special attention is paid to the geometric modeling of the human body (with a bird's eye perspective of head and two shoulders) and the flow rates through the exits. Although the program works several other factors such as different walking speed attributed to individuals into its motion- related calculations, all of which have "social significance" these are not based upon concepts or well-founded assumptions regarding *social relations, culture, or group integration*. For example, the walking rates are not dependent upon the density of the groups associated with the individual nor do they take into account factors such as "physical harm" in which calculations would differentiate the location of the pedestrian relative to the source of a disaster in progress or the person's perception of the level of danger associated with a dangerous incident. Other factors that might be considered in such models are:

****
****may not need as much detail below … but will move pieces of it to the examples given in the ******elements of the pattern later in the paper *****

****

- The influence of control agents, leaders, keynoters, and other leaders that might be available to the pedestrians during the evacuation process.
- The presence of information whether disseminated via leaders or other announcements or rumors in the environment.
- The presence of other sensory cues such as the smell or visual presence of smoke, fire, the sound of explosions, excitement and behavior of other groups.
- The influence of social bonds during an evacuation. These will lead people to consider staying behind to wait for members of their social groups. Thus group membership has a direct influence upon the behavior and thus movement of individuals in the groups.
- The decision process that might influence individual or groups of pedestrians to (Drabeck ****):
    i. Evacuate immediately
    ii. Actively seek confirmation
    iii. Await confirmation
    iv. Do nothing

- The consideration of categories of personal space. Different pedestrians have varying calculations of personal space (the space the enforce between themselves and others) that is influenced by such factors as the social situation, their perceived relationship with others, the tone or type of conversation, gender of the pedestrian, cultural norms, age (Berkowitz, 1971)
- The influence of a pedestrian's membership in different types of groups such as primary, secondary, and nested secondary groups (Canter, 1980 (VERIFYREF)).
- Other attributes of the pedestrian such as their familiarity of the setting, whether or not they are transients, etc (*****)

Many of the authors of this paper have been involved in multidisciplinary research projects in the past. Certain effective patterns have emerged that describe the problems of model integration and definition that we believe would be helpful to participants in other such projects. Here in this paper we talk about these patterns and use our building evacuation project as a rich example of a project where the adoption of this approach is appropriate and useful.

The pattern: This pattern describes a set of steps that can be taken to help integrate the various disciplinary views into a coherent one for the purposes of the final goals of the project. This is not a "waterfall" model in that it is not a set of steps that are to be carried out in a linear fashion. It is instead a spiral model in which each of these steps is repeated to the level of refinement needed in the project. (These are models that are familiar to computer scientists [Pressman 2005].) The aspects of this multidisciplinary model are described in the sections below:

Describe what (and only what) must be captured. In communicating the constructs from a given discipline, there may be many things that have to be mutually agreed upon early in the project. For example, in our project, there are social science models that capture the way in which pedestrians might privately converse with themselves and others that are not meaningful in simulating their movements. So in the model of Figure 1, pedestrians might spend some time consulting with their groups in making decisions, but the process that describes these conversational and mental decision processes are not of interest here. However, the milling behavior of a group during a decision period is important because it influences pedestrians' movements. So we restrict ourselves to only those models that directly influence pedestrian motion. This is a decision that was articulated early on in the project.

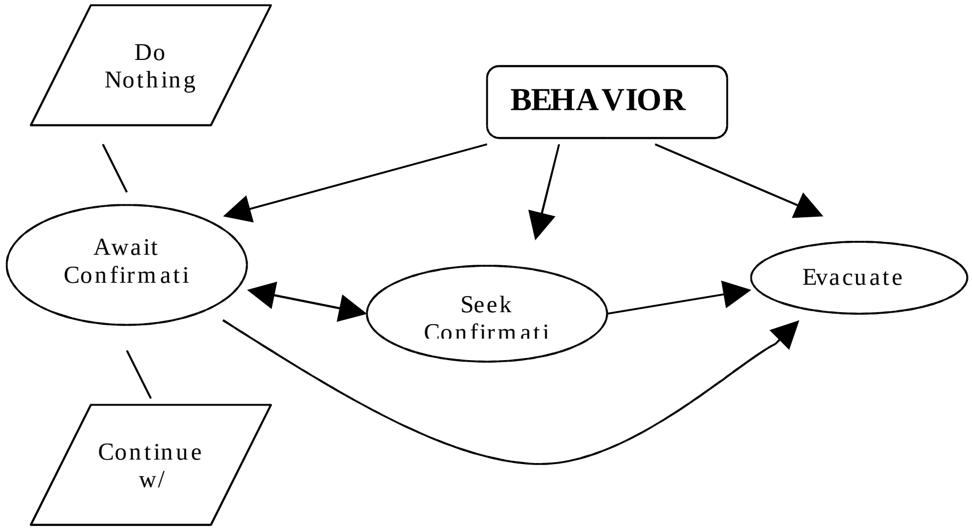

Figure 1: Social & Individual communication

In our project we strive to integrate the social science model that is concerned with individuals and groups of individuals and with the engineering model that has extended our understanding of vehicular traffic flows to the pedestrian situation to specify a robust simulation model that includes the best of both. Below is a diagram from our original project proposal. It describes our early understanding of the tasks ahead of us. First we are planning video capture of pedestrians in buildings. From this, we develop a model of what we are seeing in this captured information. This information is combined with information from the structural model describing the geometric characteristics of the building we are simulating, and the behavioral model we have developed with the social scientists. All of this is again combined and integrated into the engineering model that simulates the pedestrian flow (taking into account the geometry of the building, the social scientific information, and what we have observed in the visual capture). This is in turn mapped to the software model needed to specify the software implementation. Notice that this model is somewhat linear without iterations. We no longer advocate this; most complex problems are not solvable in such a linear fashion. Feedback and re- visiting the problem is usually required. It also hints that at each step, developers will work in isolation to develop their own piece and someone will integrate the output into the next step. The problem with this is that there is a lack of experts who are literate in both disciplines and able to map well from one model to another.

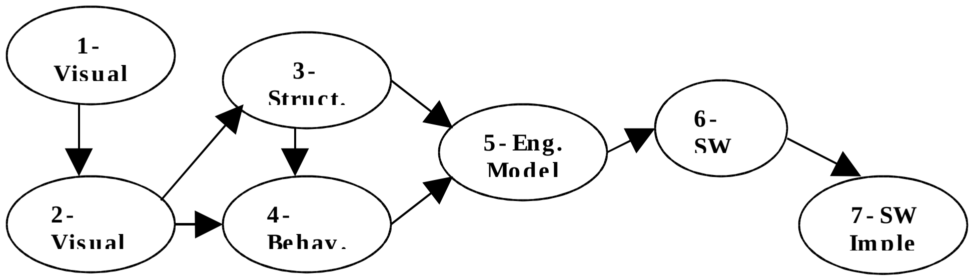

Figure 2: Project Management – Flowchart of Project Tasks

We have instead found that it is first important to assemble all experts into one room to identify the important constructs that need to be captured. This can be carried out in several productive ways:

1. Brainstorming
2. Ask experts to write literature surveys about developments in their own disciplines that pertain to the problems at hand. It was particularly helpful in our project to hear from the social scientists and civil (transportation engineers) in this way. It helped the computer scientists to hear what the important modeling issues were. For example, panic is assumed by those who have not studied evacuation incidents to be present far more often than it is in reality [Aguirre & Santos, 2004]. And "crowd" is not a term that is recognized by the social scientists as carrying important information for the purposes of group analysis (a "crowd" is just an aggregation of a collection of groups that in themselves carry information about relative behaviors of pedestrians. These constructs tend to "trickle down" to computational data structures and algorithms in the code. If they are not well understood early on, the software prototype will be flawed.
3. Presentations *******
4. View existing prototypes **********

****

Now show how Angel and JY's model of video capture and validation emerged with an iterative aspect to it. ******

Describe the important construct types (objects, attributes, procedures and rule) Next a uniform and well defined set of terms with definitions that can be used to communicate among the various modelers is needed. Some examples include group, threat, and panic. Once defined, these can be used as a common set of objects, descriptors, and relationship in constructing each of the models. Our integrated model is comprised of the following features. Details of this draft model can be found at www.*****: More is said about how to map from other models to these constructs later in this paper.

- Objects – representing pedestrians, groups of pedestrians, geometric building features and other inanimate objects (obstacles), event (such as a fire or other catastrophe), activity agenda (of a pedestrian), etc.
- Rules – used to determine the next state of the system. The input to at rule is the current state of the system and the output is the next state.
- Computational model – specifies the order with which the states of objects are updated. At any given time in the system, each object has a given state. We can think of time in the system as a sequence of discrete intervals of time. At the end of each interval the all object states are updated. This is a standard approach used in most other simulation tools (REFS).

Construct the diagrams ******Show some from each of the domains ….. speak on a meta model level about similarities and differences.*******

Explain the important aspects of the project that must be captured for your model It is sometimes difficult to decide if a part of a specialized model is necessarily private to the experts of the domain in which it is being developed, or if it should be elaborated to the rest of the group.

Traditionally, we try to abstract certain parts so that members of the group are not lost in the unnecessary detail. However, there are times when the detail is so important to the overall design and implementation of the system, it needs to be revealed and discussed in the group. For example, the overall control structures for the software needed to be discussed in the group because it would influence the outcome of the simulation and the presentation of the multidisciplinary models that are integrated to provide the end product. So, for example, in our project, the following questions needed to be explained, discussed, and answered in the multidisciplinary environment:

- At the end of each time interval for updating pedestrian positions, what should be the ordering for updating each object? The state of each object (including a pedestrian's position) is determined by the states of other objects in the system. Thus different object update orderings will yield different next states in the system. Should the objects be randomly be selected for update or should there be a specified ordering that takes into account the social-psychological states of the objects?
- Should the control of the system be centralized with a controller specifying the update ordering, or should there be distributed competing objects that update according to their ability to acquire resources (such as a slice of system time and a piece of memory).
- What should be the ordering of the firing of the rules that dictate the next states of the objects? There could be potentially many rules that determine the next step for each object. The order or weighting of each rule is important in determining the outcome.

The answers to these questions influence the coding of the system and must be captured early on as a procedural piece of the conceptual design.

Describe your model (Teach others to be literate in your domain notations) This makes the final integrated model easier to derive. An example of a draft model that was constructed by our social scientists in consultation with the engineers early in the project is shown in Diagram 1. This captures the behaviors pedestrians with respect to decision making after a hazard is present and known in the environment. It was constructed after the literature was consulted with respect to the expected alternative behaviors. The problem for the computer scientists is that different notations in their models specify well defined and specific constructs for code implementation. So the differentiation among the various icons such as squares, ovals, and parallelograms is very important. After analyzing the diagram below and other similar ones, the computer scientists concluded that parallelograms in the diagrams are sub- constructs of objects (sometimes attributes making up the objects and other times sub- objects). In this diagram below, states of pedestrian objects are being described. The Behavior box shows that behavior is influenced by these states (the pedestrian upon finding a potentially dangerous incident in the environment, usually enters one of three states: await confirmation, seek confirmation, and evacuate. While awaiting confirmation, one of the following sub- states can be entered: do nothing or continue with the agenda.

****Also refer to the vulnerability diagrams in the paper(Turner, 2003) an example of a published Soc. Model and talk about the mismatch between what is needed computationally for the soc. scientists….**********.

*****Talk about how we got hung upon "static versus dynamic" in the CS model (hard to explain to the Soc. Sci. folks **** as well as "global versus local" in the soc. Sci. model … hard to explain to the CS folks ******

In designing the system, we needed to differentiate between emergent and prescriptive characteristics. For example, one of the important objects that capture human behavior is that of the group. Pedestrian are members of prescribed groups such as family and loved ones, and emergent groups (we shall from now on refer to these as clusters). An individual pedestrian will be assigned to prescribed groups (we refer to as simply groups) at a particular time interval, and will not be removed until they complete a certain task. For example:

- Family groups traveling together will be assigned to this group and not leave it for the duration of the simulation.
- Groups of pedestrians disembarking from an airline and who have luggage to retrieve will be members of a prescribed group that will try to stay together (although not as tightly as family groups).

Pedestrians moving along a corridor in the same direction with an agenda that continues to propel them in the same general direction might find themselves in a cluster (emergent group). This group might cease to exist at any time and members will not be quite as compelled to stay in the same group (although some research indicates that there might be some affinity for some pedestrian types).

Decide what is important from one model to yours and why

Devise a mapping to the goal model – The social scientist is trying to model human behaviors. This is a very complex description and is rich in feedback loops and the organization of the model is not particularly organized. This is the nature of the individual (this problem arises in the biological sciences as well). In order to map to the software model, notations must be consistent and easily map to important computing constructs such as data structures and attributes and processes. The design should be elegant and as simple as possible. The more complex the design, the more likely the implementation will contain errors, and the more complicated it is to analyze for computational feasibility (*******talk briefly computationalcomplexity somewhere***). So the mapping process is one of folding the most important information from the application domain into the computational model. For example --- show a mapping.

Iterate (the spiral method)

****

This stuff below has to be added to the pattern above

****

Model Integration Issues: As we have seen, there are many modeling activities and constructs that are similar. So let's first investigate what can be done in a consistent manner and in collaboration among the groups early in the project. First, the primary objects and constructs and their definitions must be agreed upon. Once this is done, each team can construct their models, but there needs to be a spiral technique for these developments in which each team consults with the other. This means that each modeling group must consult with the others to communicate their modeling notations and newly emerging constructs. Eventually, they must be merged into a conceptual model that is suitable for describing the system code.

Integrated Model: It is easy for the developers to imagine that they will receive the models from the domain experts and translate them into the conceptual model needed to specify the software. We have seem examples above that indicate this is not possible. Below in the integrated model, we see that these constructs cannot be completely decoupled or neatly mapped from the social science or engineering models.

Computational Model: The above summarized many features that influence the movement of pedestrians in a given setting. This information must be mapped to the appropriate computational model. With regard to each of the above, we do not seek to simulate the thought or conversational processes involved in each of these activities, but instead the resulting movements associated with each. For example, resulting milling behaviors present during ii or iii might be simulated. A clear description of the computational model is essential for the design and development of the simulation code. In this application, there are two models of control that were defined by the computer scientists, centralized and distributed. These describe the means by which the objects and rules are used to compute the movement of the pedestrians. In the centralized model, there is a piece of code called the *controller* that directs the motion of the objects and orchestrates the order with which objects are updated and rules are evaluated. The advantage of this method is that the controller can choose to include social science considerations into the object state evaluations. The second or distributed model assumes that each object has its own thread of execution within the system and the operating system will take care of the ordering of the firing. One advantage might be a pseudo- random ordering with respect to the evaluation of object states. However, the operating system schedules processes for other reasons than fairness among processes (threads). Efficiency, response time, turnaround, and throughput are also important considerations {Tanenbaum and Woodhull, 1997).

While the above seems to be a very low level consideration, the decision to develop the software with a centralized controller or a set of distributed threads is part of the conceptual design and cannot be totally separated from the implementation in this type of application. In any case the distributed versus centralized decision should be made early on in the design.

References:

Aguirre, B. & Santos, G., 2004. "A Critical Review of Emergency Evacuation Simulation Models", Disaster Research Center, University of Delaware, Newark, Delaware 19716.

Alexander, C., Ishikawa, s., et. Al. 1977 *A Pattern Language*, Oxford: Oxford University Press.

Berkowitz, 1971, "A Cross National Comparison of Some Social Patterns of Urban Pedestrians".

Canter, D (Ed.). 1980. *Fires and Human Behavior*. New York: John Wiley & Sons.

Coplien, J.O. and Harrison, N.B.2005 *Organizational Patterns of Agile Softare Development*, Pearson, Prentice Hall.

Drabek, E. T., 1996. "Disaster Evacuation Behavior Tourists and Other Transients", Institute of Behavioral Science, University of Colorado Boulder, CO, USA.

Gamma, E., Helm R., 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*, Reading, MA, Addison-Wesley.

Pressman, R. S., 2005. *Software Engineering, A Practitioner's Approach*, Sixth Edition, McGraw Hill.

Tannenbaum, A.S, & Woodhull, A.S. 1997, *Operating Systems, Design and Implementation*, Second Edition, Prentice Hall, p. 83.

Thompson, P.A., Wu., J., and Marchant E.W., 1996, "Modelling Evacuation in Multi-storey Buildings with Simulex." Fire Engineers Journal, Vol. 56, no. 185, pp. 6-11

Turner, ,B. L., Kasperson, R. E., et.al., 2003, "A Framesork for Vunerability analysis in Sustainability Science", PNAS, (100,14), p. 8074-8079.