LA-UR-03-2754

Approved for public release;
distribution is unlimited.
04/25/2003

c. 1

Title:

SIMULATION OF HUMAN FACTORS FOR MATERIAL SAFETY

Author(s):

Andrew Koehler
Johnell Gonzales-Lujan
George Tompkins
Robert J. Burnside
Drew Kornreich

Submitted to:

INMM Workshop
Oak Ridge, TN

Los Alamos
NATIONAL LABORATORY

Simulation of Human Factors for Material Safety

Andrew Koehler, D-1
Johnell Gonzales-Lujan, D-2
George Tompkins, D-2
Robert J. Burnside, D-2
Drew Kornreich, D-2

Los Alamos National Laboratory
MS F609
Los Alamos, NM 87545

The D-1 (Statistical Sciences) and D-2 (Stockpile Complex Modeling and Analysis) groups frequently collaborate to analyze production capabilities at Los Alamos National Laboratory. The facilities in question run the gamut from traditional machining to the fabrication of Plutonium components. This paper documents our efforts to extend our modeling capabilities from traditional discrete event simulation modeling to include agent based models.

1 Why Move Towards an Agent Based Approach?

Material safety in nuclear operations depends both upon what people do, as well as how they do it. For example, small differences in how a worker removes his or her hand from a glove-box greatly determine the likelihood of a skin contamination event. The accumulated effect of many such small procedural issues, along with the combined effects of a host of specialized work-rules, have been observed by our groups over many years to often play a much greater role in determining the efficiency of our process operations than do actual machine throughputs, input-output strategies, or process queuing logic.

All process simulation models are idealized abstractions of actual systems. Frequently one of the most abstracted parts of existing process simulations is that of human factors. Typically, process models treat persons in the system as simply an 'operator' or 'resource' whose availability constrains throughput. Such abstraction is appropriate and useful when the goal is to model and optimize throughput or another quantitative performance measure. In general, however, this approach fails to predict system behavior when such behavior depends upon how workers deal with the uncertainties inherent to compliance with complex (and sometimes conflicting) work rules and physical limitations. As a result, traditional modeling makes it very difficult for analysts to address questions involving "socio-technical" system factors such as: what might happen when time pressure causes human operators to hurry, and what might be the unforeseen consequences of their work rules and safety polices? More importantly, how could these work rules and policies be improved?

These questions are vital in any industrial setting, but even more so in settings where nuclear/hazardous materials are involved. Discrete event simulation modeling which can incorporate the types of uncertainties mentioned above provides an avenue for the safe identification of consequences, and a controlled environment for testing changes. Discrete event simulation also provides an avenue for identifying interdependencies and interrelationships between system components.

Based on these considerations, we have begun development of a generic agent model of an operator. This development effort is part of a broader effort to introduce high fidelity visual and physical models of process operations at existing and future facilities at Los Alamos. The remainder of this paper will discuss our requirements for the operator, the software implementation and system architecture we are in the process of developing, and our progress to date.

2 Requirements

Our initial requirements for the operator agent include:

- Compatible with a decentralized approach to process model architecture
- Relative efficiency in handling "human like" decision-making
- High degree of reusability across process operations
- Modular approach separating "high" and "low" cognitive functions

One of the most significant limitations we have found with existing process simulation packages, is that they assume an unrealistically high degree of information and decision-making centralization. For example, operators and parts are routed via optimized path logic, or through the use of "God's eye view" routing tables. We want to avoid, if possible, centralized information and decision making for the operators. One of the most elemental areas this impacts has to do with navigation; how will an operator move from machine to machine and room to room? Our goal is for the operator to possess internally, as completely as possible, the information about the environment around it. (e.g. where walls, doors, and equipment are located). The operator has the ability to observe (with the option to follow) special room/equipment requirements and work rules associated with the work environment. This would include the type and amount of material allowed in a given area and any relevant standard operating procedures. This decentralization of decision-making logic will also allow workers to be given (or gather) knowledge sets that may be erroneous or incomplete in a way that is more like behavior in the real world.

The discrete event paradigm dictates that events must be generated for the operator to cause it to check its proximity to the objects in its area. However, too many such events will slow model execution, therefore the approach for detecting proximity must be efficient. Additionally, although we want to allocate enough CPU time to each agent to enable “human-like” goal analysis and behavior, this is not an artificial intelligence project, but rather a pragmatic attempt to get our process models to behave more like what we observe occurring on the shop floor.

Further, we would like the operator’s logic to be as modular as possible. For example, we want to separate the “autonomic” or “low” logic for functions like navigation, and basic work rule compliance, from “higher” level decision logic (e.g. ‘what do I do now’). The goal is an operator that, by default, will not require logic or software modification to navigate through whatever environment we establish, so that we can focus on customization of higher level logic in future models. This will allow us to use the operator to explore complex, interactive, behaviors without having to reinvent the agent in each type of work setting.

3 Software Implementation and System Architecture

Rather than attempt to develop a software tool from scratch we are working with a tool called Flexsim (www.flexsim.com). Flexsim was elected for two reasons; 1) it provides an existing, object oriented, discrete event simulation engine which is readily extensible using C++, 2) it provides (by default) a high-fidelity 3D visualization of the system. Since many of the behaviors we are interested in are most readily detected by observation the ability to visualize output is paramount for evaluation and communication.

The object oriented nature of the software lends itself to model decentralization—the alignment of physical and decision logic with the actual level of object in the model comparable to real actors. This is especially important to us, as our modeling efforts are concerned both with prediction of output under ideal circumstances, but also seek to identify potential process disruptions more efficiently than we have been able to in the past. With experience, we have come to the recognition that previous modeling efforts have at times prevented the recognition of potential “surprises” before they occurred because the modeling tools used did not permit sufficiently accurate representations of how things actually work in the physical world.

3.1 Decentralization of Decision Making

Decentralization of decision-making is inherent in the Flexsim architecture, however designing models while employing this as a design philosophy has required us to greatly rethink how we approach process simulation.

We have taken decentralization of decision-making we to mean two things: first, that no entity in the model will be allowed to know more than what it would in reality, and second, that entities only have ability to “decide” actions based on what their actual decision authority is in reality. While we are not always completely successful at following these rules, we have attempted to make every effort to allow “agency” in our models only in cases where it really exists.

For example, in some simulation environments, routing information is considered to be an inherent property of each part that moves through a production system. This is a sometimes useful abstraction. However, this abstraction places too much information at the level of an inanimate object for purposes of our Flexsim effort—and by so doing makes overly difficult the introduction of worker behavior such as part mis-identification or errors in routing. By contrast, in the real world, an individual part knows nothing about where it has come from or where it is going—indeed it "knows" nothing at all except for a very basic set of low level behaviors such as "I will succumb to gravity if not on or being held by something" or "I will burn if put into an oxygen atmosphere." Likewise, our worker agent does not know what is going on unless he/she is specifically told by another process, can see the event in question, or is told by another worker or entity in the model.

This decentralization of decision-making and information in some ways makes the modeling task more challenging up front. For example, inherent to the idea of decentralization of decision-making in the context of the operator agent model is the idea that agents cannot know what each other are doing unless they are in visual sight, or have some other means (radio, telephone etc) of querying each other. This has meant we must take more time to develop "low" level behaviors for operator agents. Rather than routing tables, we must create an independent ability for operators to communicate (and the underlying logic as to when they will seek information from each other), some degree of vision for operators (so that they can find out what objects and worker are around them), a collision detection system (so that they can develop their own pathway routing), as well as different sets of decision logic and preemption of goals (so that operators know how to pursue both long term goals and return to those goals when short term tasks interrupt them).

However, as these infrastructures become more complete, we have increasingly found benefits that we would not have been able to enjoy through traditional, centralized, modeling. In our process operations, much of the work is done following a "two man rule"—which basically means that operators must maintain visual contact with one another throughout work activities. By designing the agents from the start to deal with limited information, it has been relatively simple to implement behavior that takes into account not only whether two agents are near each other, but indeed, whether or not they can actually "see" each other given equipment, room layout, and operational constraints. Furthermore, since behavior can be inherited easily, once we have written the necessary code for one worker, all workers in the model can quickly be given either identical behaviors, or whatever deviations from that behavior we want to use to simulate incomplete compliance.

Additionally, because this modeling logic forces development of basic capabilities, we have found advantages in reusing and recombining these simple components in ways that we did not anticipate at first. The method we use for collision avoidance serves multiple duty as a basic component to our physics, vision, and facility navigation system. With relatively little development work we intend to expand upon this to use it to provide real-time dosimetry as well.

3.2 Decomposition of Physical Logic

Similarly to our efforts to decentralize model decision-making, we have also aggressively sought to model physical objects and the logic of physical object interactions from the ground up.

In standard process simulation, or functional block flow modeling, objects are highly abstracted and generally are given only the minimal set of characteristics necessary for them to be linked to the simulation kernel. This approach makes sense if modelers are only concerned with calculating the nominal time and efficiency results that may be obtained from different system configurations. Our problem has (over time) developed into being better able to understand why we get the facility performances we observe and to explain why these differ from our nominal T&E results. Performance has, through observational data, been found to depend critically upon a host of small "cascading" physical interactions between objects. Because of this we have sought to capture a higher level of detail about objects in our model—including some that our existing models have no present use for, but for which we envision a later need.

For example, at a most basic level, every object in our models has 3-D spatial characteristics. We are presently working on several approaches to provide a representation of the object's material properties, including weight, radioactive, shielding, strength, and chemical traits. While this has again required greater up-front efforts, we believe it will provide long term benefits—particularly in understanding unintended consequences of layout/work-rule interactions.

The purpose of this effort is to develop a means by which we can more fully understand whether or not certain physical model components/constraints have an impact on agent behavior. We also envision moving towards facility simulation logic that is entirely driven from the bottom-up. Rather than having a central logic that prevents (for example) the over pressurization of a glove-box, we instead want to be able to simulate a realistic chain of failures: failure to follow a work-rule, leading to a valve being left partially on, leading to flow of steam into a glove-box, leading to the gloves being blown off their ports. Each step in this chain should happen "naturally" and without centralized event triggers. Just as in reality, no component in this chain has any knowledge of what the upstream components are—rather the valve simply "lets" whatever it is hooked up to through at a flow rate when open, the glove box simply "knows" it's internal pressure is x, and the gloves "know" that they fail whenever over pressurized (regardless of whether that over pressurization is due to steam (which we might anticipate) or a worker agent deciding to overfill a box at some other point with cream-cheese due to a conflict in work-rules (which we might not anticipate).

4 Path Forward

At present, our development work is focused on construction of several key components to the agent model. We are presently using the example of the Tokaimura MOX fabrication plant (see Figures 1 and 2) criticality accident as a physical framework within which to develop the necessary building blocks to simulate several low level behaviors, including basic work-rule compliance, navigation, avoidance, and interaction with process logic as part of goal completion (Figure 3). This effort should be finished within a couple of months.

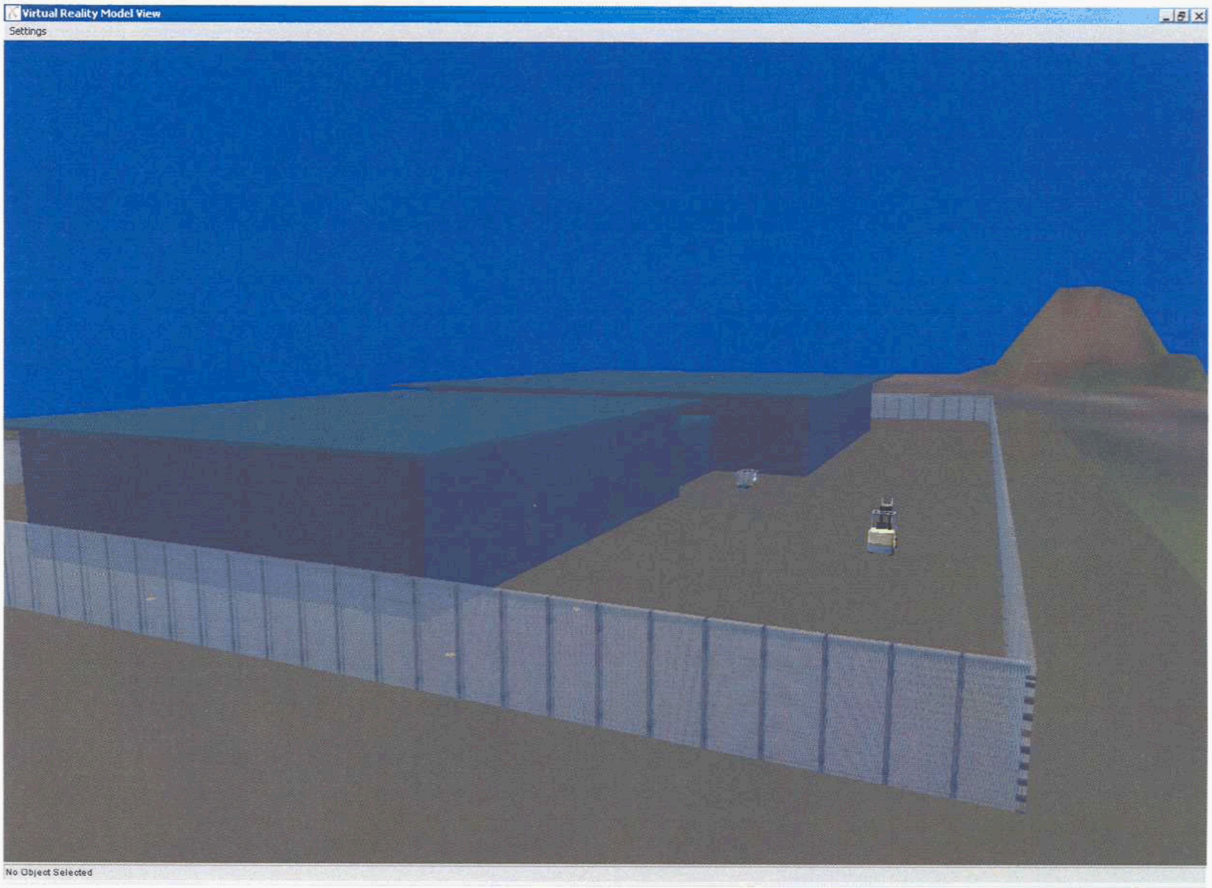


Figure 1: External View of Tokaimura Plant

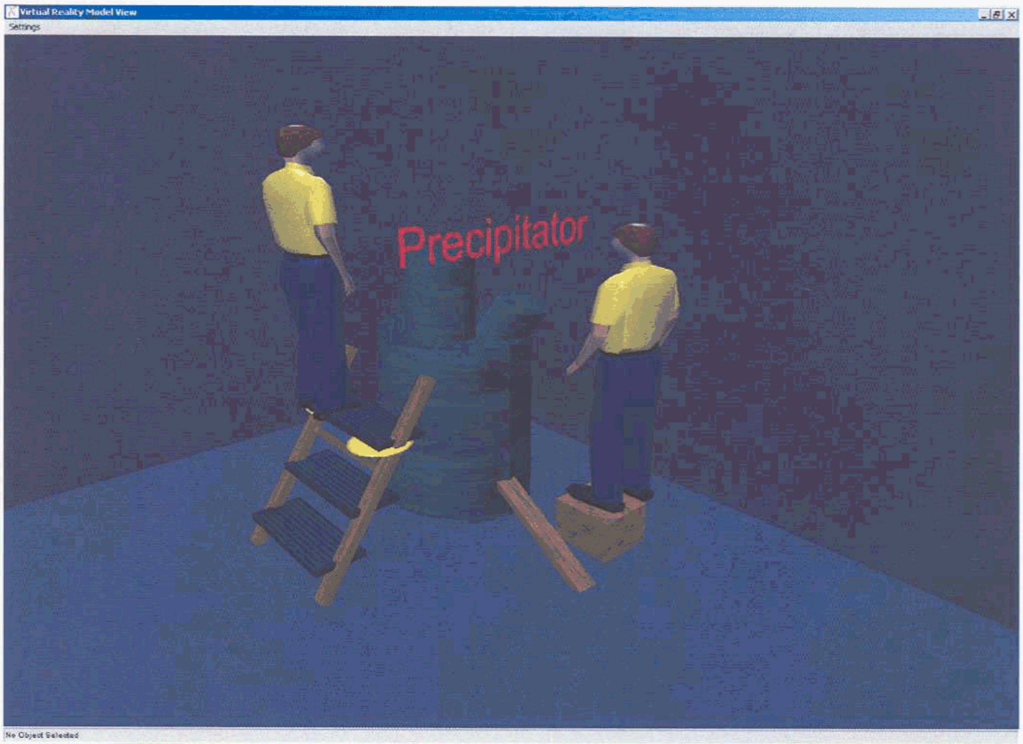
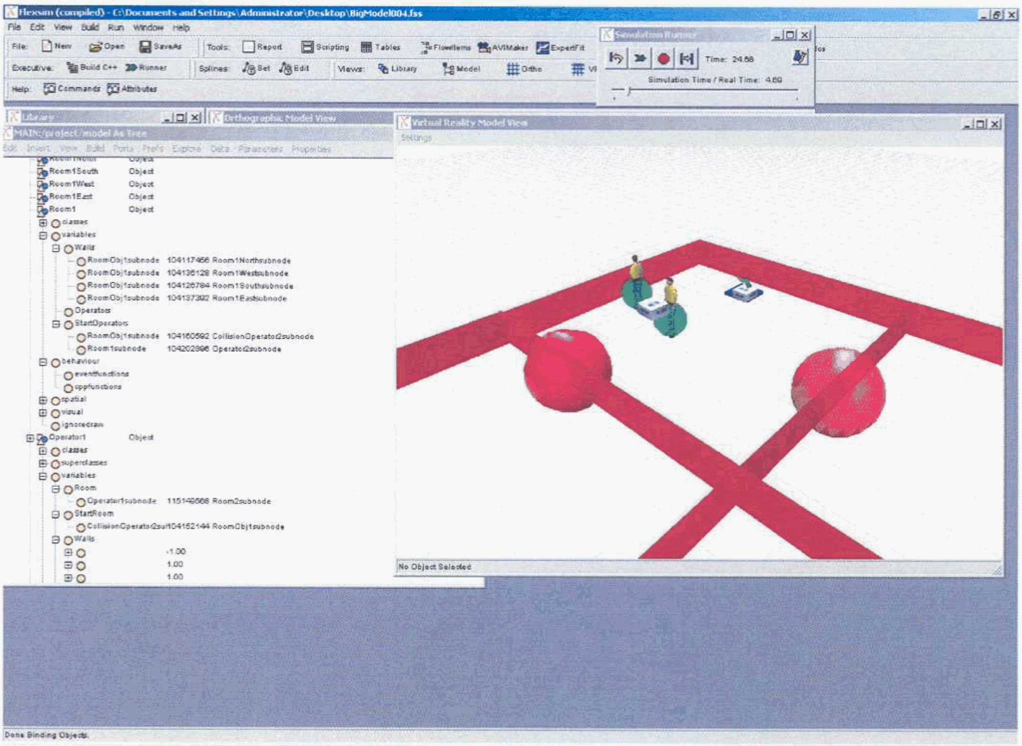


Figure 2: Location of the Criticality Accident



```
Room1South          Object
Room1West           Object
Room1East           Object
Room1               Object
  classes
  variables
    Walls
      RoomObj1subnode   104117456 Room1Northsubnode
      RoomObj1subnode   104136128 Room1Westsubnode
      RoomObj1subnode   104126784 Room1Southsubnode
      RoomObj1subnode   104137392 Room1Eastsubnode
    Operators
    StartOperators
      RoomObj1subnode   104160592 CollisionOperator2subnode
      Room1subnode      104202896 Operator2subnode
  behaviour
    eventfunctions
    cppfunctions
  spatial
  visual
  ignoredraw
Operator1           Object
  classes
  superclasses
  variables
    Room
      Operator1subnode  115149568 Room2subnode
    StartRoom
      CollisionOperator2su104152144 RoomObj1subnode
    Walls
                        -1.00
                        1.00
                        1.00
```

Figure 3: Example of Navigation and Collision Detection Infrastructure

Armed with these basic tools, we are then going to undertake a set of modeling projects at Los Alamos starting in the next fiscal year. As part of these efforts we will be completing work on an integrated chemistry/physics module, real-time radiation dose components, and development of higher level work-rule logic.