Applying Software Product Line Technology to Simulation Modeling of Emergency Response Facility

Guodong Shao
Y. Tina Lee
Manufacturing Systems Integration Division
National Institute of Standards and Technology
100 Bureau Drive, Ms 8260
Gaithersburg, MD 20899-8260, U.S.A.
gshao@cme.nist.gov

This paper proposes an effective, reusable solution for modeling and simulation. The approach is to develop a Software Product Line (SPL) architecture that explicitly captures the commonality and variability in a family of similar simulation systems. First responders and incident management personnel need better training resources to prepare for possible catastrophic events, including natural disasters and terrorist attacks. Live exercises are often very expensive to organize and conduct. With modeling and simulation technology, simulation-based exercise and training models could be developed. The SPL technology enables the simulation system to easily adapt to multiple contexts and allows the simulation system to reconfigure quickly. SPL models can be developed at different levels of scope and depth to suit the emergency responder's needs. By reusing the system requirement analysis, software architectures, and design, the development time is shorter, the development cost is lower, and the quality is easily maintained.

This paper focuses on the application of SPL technology to simulation systems of emergency facilities, such as hospital emergency rooms, on-site emergency triage stations, decontamination stations, first-aid stations, and ambulances. The paper also introduces a prototype simulation of the SPL member system – a hospital emergency room simulation system that has been developed at the National Institute of Standards and Technology. The detailed modeling of a family of emergency response facility simulation systems is performed based on the Product Line Unified Modeling Language (UML)-based Software engineering (PLUS) method.

Keywords: emergency response, homeland security, simulation, Software Product Line (SPL), software reuse, Unified Modeling Language (UML)

1. Introduction

There is a growing need for improved preparedness of emergency response for both man-made and natural disaster events. Effective emergency response presents a number of challenges to the responsible emergency response providers. The Homeland Security Act [1] defines that emergency response providers as "Federal, State, and local emergency public safety, law enforcement, emergency response, emergency medical (including hospital emergency facilities), and related personnel, agencies, and authorities." These providers need to be trained and be ready to act in view of increased security threat. One major challenge is the lack of opportunities to train the emergency responders and decision makers in dealing with emergencies [2]. The responsible agencies have tried to meet these needs through the organization of live exercises, but such events are often difficult to organize and very expensive to conduct. Modeling, simulation, and visualization techniques can help to address this training challenge. The Software Product Line (SPL) technology [3] enables the simulation system to easily adapt to multiple contexts and allows the simulation system to reconfigure quickly.

This paper is organized as follows: This Section describes the importance of hospital emergency room (ER) as an emergency response provider (Section 1.1), the modeling and simulation for homeland security (Section 1.2), and background information of SPL technology (Section 1.3) and modeling and simulation technology (Section 1.4). Section 2 formulates the emergency response application problem and outlines the technical approach to solve the problem. Section 3 explains the detail design of the simulations' SPL architecture that includes the context model, use case model, feature model, static model, dynamic model, feature/class dependency model, and design model. All these models are presented using Unified Modeling Language (UML) diagrams [4]. Section 4 demonstrates the SPL architecture's reusability by presenting a prototype ER simulation system. The procedures to derive the ER simulation system from the SPL are described. The final section concludes the paper.

1.1 Homeland Security and Hospital Emergency Room's Role

After September 11, 2001, the nation's domestic incident management landscape changed dramatically. Today's security threat includes not only the traditional spectrum of man-made and natural hazards but also the deadly and devastating terrorist attacks with Chemical, Biological, Radiological, Nuclear, and Explosive (CBRNE) weapons. In March 2002, the Department of Homeland Security (DHS) was established to provide the unifying core for the vast national network of organizations and institutions involved in the effort to secure America.

Among many emergency response providers, ER plays a very important role to respond to emergencies, to save people's lives, especially in the event of a CBRNE attack. In fact, DHS is concerned with the nation's capacity to handle the "surge" that might emerge during a terrorist attack. Emergency rooms are on the frontline. The DHS University Center of Excellence for the Study of Preparedness and Catastrophic Event Response (PACER) is conducting research to improve the nation's preparedness and response to high consequence natural or man-made disaster. The intent of this research is to alleviate the effects of such events by developing and disseminating the best scientific practices. Surge capacity is one of the most important sub domains identified in the Homeland Security Presidential Directive [5]. The term 'surge capacity' is often restricted to health systems and their ability to absorb large numbers of additional patients. Regional, state, and national health systems are striving to increase health systems (hospitals and clinics) surge capacity in preparation for anticipated acts of terrorism, avian influenza pandemic, and a multitude of other public health threats that may result in mass casualties [6] [7].

DHS is concerned about the nation's ability to quickly detect evidence of biological and chemical terrorism. Emergency rooms will be among the first to detect these events.
ER physicians and nurses would likely be the first to detect exposure, identify the agent used, and treat the victims. Most victims of chemical or radiation attacks would be decontaminated by firefighters and emergency medical services workers at the scene and then be rushed to local

emergency rooms. Hospitals could probably handle many types of smaller-scale attacks such as bus bombings. However, experts say most U.S. emergency medical personnel are not properly trained or equipped to deal with a large number of victims of a chemical, biological, or radiation attack. After a CBRNE attack in a U.S. city, thousands of people - both actual victims and panicked bystanders - would flood local emergency rooms for testing, decontamination, and treatment [8]. Early detection of the agent is essential for ensuring a prompt response to a biological or chemical attack, including the provision of prophylactic medicines, chemical antidotes, or vaccines [9]. DHS is concerned about public response to a terrorist attack. Emergency rooms need to not only treat those in need in a timely manner but to be able to communicate with the public that the situation is being handed effectively. In his testimony for a hearing on "Risk Communication: National Security and Public Health" before U.S. House of Representatives, Kenneth Shine, President of Institute of Medicine, the National Academies, stated that "Key to the role of public health is education and information for the public and for health professionals. We know far too little about the availability of hospital beds, burn units, decontamination capability, and a variety of other parameters required by the health system to deal with terrorism. Moreover, the necessity for dramatically improved communications between the public health system, the medical care system, and law enforcement all require a high level of coordination and communication" [10].

1.2 Modeling and Simulation for Emergency Response Management

Modeling and simulation provides a non-destructive and non-invasive method of observing a system and also provides a way to test multiple inputs and evaluate various outputs [11]. Simulations allow users to reconstruct a comprehensive representation of real-world features during disaster response [12]. The limitations of live exercises can be overcome through use of simulation models that allow emergency response personnel across multiple levels in multiple agencies to be exposed to the same scenario. Simulation models can help the decision makers determine staff and resource levels in hypothetical terrorist attack scenarios. They can provide a wider range of training at a much lower expense. In addition, they can define the rules of operation, the probabilities of paths of action and time duration, and letting the events unfold. Simulations can also be used to study the impact of disaster events as a whole. For example, we need to understand how a radioactive plume released by terrorists would disperse. This knowledge would allow us to plan what traffic routes people would use to evacuate the affected areas, and what demands would be placed on the hospital resources in the area.

Sullivan identified the importance of disaster simulation and emergency response [13]. Simulation modeling has also been identified as one of the leading techniques for helping improve the incident management capabilities [14]. A recent survey [2] indicates that a number of modeling and simulation applications for analyzing aspects of various disaster events exist. For example, a simulation model of the ER for a hospital was developed at Texas Tech University. The model allows users to analyze patient flow throughout the treatment process, assess the utilization of ER resources, evaluate the impact of a hypothetical bioterrorist attack, and determine the appropriate resource and staff levels for a bioterrorism scenario [12]. Jain and McLean [15] proposed an architecture to identify the required groups of simulation and gaming models for incident management. They defined the model's scope in the solution space that covers all phases of incident management including prevention, preparedness, response, recovery, and mitigation. The proposed architecture allows the virtual environment to be highly configurable. The simulation models identified in the architecture include:

- The *social behavior simulators* simulate phenomenon based on actions of multiple individuals. These include modeling of crowd, traffic, epidemic, and consumer behavior.
- The *physical simulators* simulate the creation and growth of the emergency incident. These may include physical phenomena such as earthquake, explosions, fire, chemical, biological or radiological plume, etc.
- The *environmental simulators* include weather, watershed, indoor climate, and ecology. These simulators will model the above phenomena and provide the outputs to other simulators for modeling the impact.
- The *organizational simulators* include fire, law enforcement, health care, other government agencies, and the terrorist organization. The simulator will model the flow of information within the organization, flow of authority and decisions, and the resulting actions.
- The *infrastructure systems simulators* model the propagation of the impact of damage throughout the infrastructure system based on the damage to one part due to the emergency incident.

Together, these simulators are able to create an incident that includes all primary aspects and the responses from relevant agencies as well.

Simulation provides an effective and efficient way to train people for preparedness of emergency response for different incident scenarios. However, as a result of various difficulties and obstacles associated with simulation model development, most emergency response organizations do not take advantage of simulation technology. These organizations typically do not have enough technical resources. The emergency response simulation models from different responders share many similarities but also have some variations, such as the responder's functionality, resources, technologies, and facility layout. It is believed that using SPL to develop a family of simulation systems will be a more beneficial approach compared to the traditional, individual simulation development approach.

1.3 Software Product Line Technology

SPL is a family of software systems that have some common functionality and some variable functionality, that is, a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [3]. It is also referred to as family of systems, software product family, product family, or product line [16]. Each system derived from the SPL consists of the common components of the SPL, and tailors them through various mechanisms such as parameterization or inheritance, adding any new components as needed, and integrating the components based on the SPL architecture. Building a new system becomes more like block selection and assembly process rather than creating one from scratch. By reusing the system requirement analysis, software architectures, and design, simulation modelers do not have to develop each simulation model anew. The SPL family defines the overall simulation domains that are required for the development of applications of individual family members. The family product line concept could be used to create more robust and better-defined models than defining unique models for each simulation situation encountered currently allows [17].

Using SPL techniques, companies such as Nokia, Hewlett-Packard (HP), LSI Logic, Philips, and Cummins have improved time-to-market, engineering costs, portfolio size, and defect rates by factors of 3 to 50 [18]. Raffo et al claimed in [19] that "…a generalized simulation model as one that can be easily adapted to multiple contexts using significantly less effort than would be required to develop the original model." They adopt the "product family" concept as defined by Weiss et al [20] to develop the generalized process simulation model (GPSM) framework, their research work described in [21] demonstrates that GPSM can be built in hours rather than weeks and the framework was very favorably received by users who were well versed in alternative methods.

1.4 Modeling and Simulation Methodologies

The simulation modeling is simplification of real system to generate the behavior by specifying a set of instructions, rules, and equations to represent the structure and behavior of the system. The structural elements of a system include the components of the system and their interactions (e.g., input, output, and their connections). The behavioral elements include the sequence of the interactions, the timing constraint of the interactions, and the operation of each component [22].

The UML is a formal graphic tool used to describe the structure and the behavior of a software system conceived within the object-oriented paradigm [4]. UML provides a design tool to visually express common concepts, and take advantage of common patterns reuse [23]; commercial systems are available to translate UML diagrams into code for the popular object oriented programming languages. UML has been a popular and widely used modeling and simulation method. Such modeling approach allows the designer to use a unified tool to identify system requirements, plan system architecture and physical component features as well as their dynamic behavior [24].

In this paper, the SPL technology is applied onto the development of simulation modeling applications for a family of emergency response facility. The detailed modeling of the simulation systems is performed based on the Product Line Unified Modeling Language (UML)-based Software engineering (PLUS) method [16]. The PLUS method extends the UML-based modeling methods that are used only for single systems to address software product lines. With PLUS, the objective is to explicitly model the commonality and variability in a family of the simulation systems. PLUS provides a set of concepts and techniques to handle software product lines. The reuse of simulation models reduce the time and cost for model development. The UML notation supports requirements, analysis, and design concepts. The PLUS method separates requirements activities, analysis activities, and design activities for SPL. The PLUS differentiates requirements and analysis from design as follows:

- In the "Requirement" phase, the functional requirements of the product line, in particular, the commonality and variability of the SPL members are defined. The activities in this phase include product line scoping, use case modeling, and feature modeling.
- In the "Analysis" phase, the problem is broken down or decomposed. The activities in this phase include static modeling, object structuring, dynamic modeling, finite state machine modeling, and feature/class dependency analysis.
- In the "Design" phase, the solution is synthesized and composed. The goal is to develop a component-based software architecture for the SPL. The activities in this phase include software architectural pattern-based design and SPL architectural design.

2. Software Product Line Engineering

2.1 Problem Formulation

Emergency response facilities may include hospital emergency rooms, on-site triage centers, decontamination stations, and first-aid stations. For example, the Red Cross and other emergency response organizations may only require an ambulance as an emergency facility or a temporary on-site tent as an emergency triage station or a simple decontamination station. The emergency response facility's primary goal is to identify and stabilize the patient for further treatment by another department or hospital. The facility only provides emergency diagnosis and treatment and then release or transfer the patient for extended care [25].

Simulation modeling of an emergency response facility/unit allow users to construct a comprehensive representation of real-world features during disaster response. Each emergency provider has its own objective, scope, and policy to provide emergency-response support for the incidents. Thus, specific simulation-based exercise and training models are useful to serve different training purposes. For example, a hospital ER model could be developed to simulate patient flow through the hospital emergency rooms. As such, it would model the deployment of resources, actions for triage, treatment of injured, movement of casualties to other facilities, and transfer of patients to another hospital/facility. The model's logic should include relevant policies and procedures for emergencies, such as calling in medical staff, using temporary accommodations for the injured, and acquiring needed supplies and equipment.

Each simulation system consists of a simulation engine and generic modules including input module, output module, patient arrival module, patient departure module, triage process module, emergency treatment module, generic test data, and scenarios. Some incident management customers require 3-Dimensional (3D) animation for the simulation model. Some simulation models need to provide a log file from the simulation run. Some simulation systems need to read staff schedules from another information system. Some need to provide a document of simulation findings.

The hospital emergency rooms being simulated should have the capability of performing lab tests (such as blood and urine tests), x–rays, ultrasound, and/or Magnetic Resonance Imaging (MRI). Among the emergency response facilities discussed above, hospital is the only facility that can admit patients for further treatment.

The patient arrival being simulated in all the facilities could be either emergency patients who are brought in by an ambulance or walk-in patients who can go to the facility by themselves. The patient departure could be one of the following three cases: discharged, admitted, or re-directed to another facility.

The inputs, outputs, and data and user interface requirements for the emergency-response-facility-simulation system SPL are identified as follows:

Inputs may include the number, location, and type of casualties from an emergency incident, the availability of staff (at work or standby (on-call)), and resources (own and those that can be acquired quickly from surrounding jurisdictions). Other inputs are the time and resources required to attend to each casualty type, the probabilities of death from different casualty types over time, emergency center location, hospital layout, process stations and station capacities, processing times, patient arrival rate, staff shifts, medical resources, and symptom-treatment profiles.

Outputs may include the operation of the emergency facilities over time including the number of people treated and released, admitted, dead, or waiting for treatment. It may also include the status of the staff and facilities, system utilization, run time interactions, simulated clock time, the number of Emergency Medical Technicians (EMT) and ambulances dispatched over time, and the number of ambulance and casualty arrivals over time.

Data requirements

a) Support data required to build the model (e.g., number of beds, triage level, and flow).
b) Externally manipulated parameters during run-time (e.g., control of the allocation of resources).
c) Primary entities including the patient and facility resource.
d) Patient arrivals modeled as statistical probability distributions.
e) Emergency vehicle arrivals modeled using statistical distributions for their travel time from their entry point into the modeled area en-route to their assigned destination.

User interface requirements

a) 2D or 3D animation display of patient flow and hospital resources queuing and being processed in an emergency facility.
b) Data entry capabilities for modifying patient characteristics, generating patient volumes, and generating emergency vehicle starting points and destinations.
c) Periodic updates of the patient status, patient location, and facility status.

2.2 Technical Approach

The development of a simulation model has always been a challenging task. It requires expertise and is very time consuming. Currently, almost every customized simulation model needs to start from scratch [26]; the repetition of simulation modeling work increases development cost. This paper proposes a more effective, reusable solution using the SPL approach, as described in Section 3. The detailed modeling of a family of emergency response facility simulations is performed based on the Product Line UML-based Software engineering (PLUS) method [16].

A prototype ER simulation has been chosen to be developed to demonstrate the Emergency Response Facility Simulation System Software Product Line (ERFSSSPL)'s reusability. Emergency response facility is the location, permanent or temporary, where emergency response resources are available and emergency support activities can be conducted. The needs of the emergency response of the incident will determine the specific kinds of facilities required. Since September 11, 2001, the definition of first responder has been broadened to include those, such as public health and hospital personnel, who may not be on the scene of the emergency events, but are essential in supporting effective response and recovery operations [1]. Emergency medical has played an important role in the event of a terrorist attack or natural disaster. Simulations of ER operations in the event of a terrorist attack or natural disaster are especially useful.

3. Emergency Response Facility Simulation System Software Product Line

The modeling of ERFSSSPL includes the context model, use case model, feature model, static model, dynamic model, feature/class dependency model, and design model. The following subsections describe the design of each model in details.

3.1 Product Line Context Modeling

The product line context model defines the hardware/software boundary of the SPL. It includes external systems and external users to which the members of the product line need to interface. In the product line context model, each external system or user has a second stereotype to depict the reuse category: <<kernel>> or <<optional>>. Figure 1 shows the simulation system SPL context diagram. All members of ERFSSSPL must have the kernel external users, while only some of the members of ERFSSSPL have the optional external system. The external users (simulation analyst or incident management personnel) and external system of the ERFSSSPL are defined as follows:

- *Simulation Analyst*: The *Simulation Analyst* is the core user of the ERFSSSPL. The simulation analyst is responsible of executing the model and analyzing the simulation results on a daily basis. He/she might be involved in the simulation system development, and is the domain expert who is familiar with the problem. He/she must understand major requirement assumptions for the specific system, and is capable of performing data collection. The simulation analyst can define various scenarios for other users, verify the model based on the scenario, make suggestions regarding the length of the simulation run, the number of runs needed, and the initial conditions. He/she is responsible for analyzing the simulation results and documenting the findings.
- *Incident Management Personnel*: The *Incident Management Personnel* is the primary user of the system. By simulating different emergency events in a virtual environment using different settings, the incident management personnel is trained to respond to an emergency incident. The response actions may include the deployment of resources, actions for triage, treatment of the injured, movement of casualties to other facilities, and transferring patients to another hospital/facility under different event scenarios in the virtual world.
- *Information System*: The *Information System* is an optional external data system that is used only by certain simulation systems. The external information system is a data repository that maintains the data required to execute the model. Data maintained may include number of patients, number of resources, and resource schedules. The resource schedules may also be changed/updated in this system.

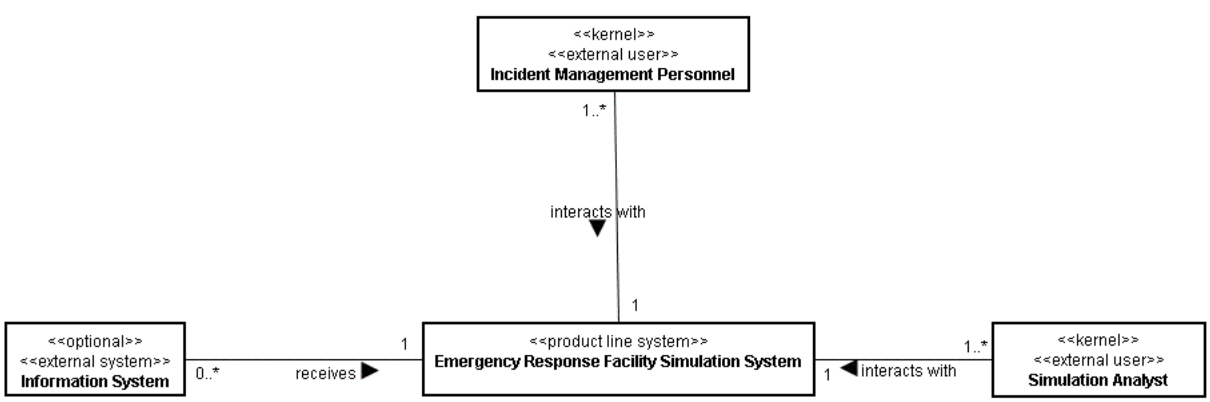

Figure 1: Context Diagram for Emergency Response Facility Simulation System Software Product Line

3.2 Use Case Modeling

Depicted in Figure 2 is the SPL use case model diagram for the emergency facility simulation system. Kernel use cases represent the core of the SPL's common part that is provided to all SPL members. In general, variability in the SPL is captured from the variation points in the kernel use case. For example, in the *simulate emergency treatment* use case, treatment can be decontamination only or can be combined with some other emergency treatments. The major variations are addressed by optional use cases, such as *simulate lab test and exam* and *document simulation findings*. Since they are optional, only certain SPL members apply these use cases. There are 14 use cases included in the use case model. There is a stereotype to depict the reuse category for each use case: <<kernel>> or <<optional>>.

As a sample, a brief introduction of the *simulate patient arrival* and *simulate lab test and exam* use cases is provided in Table 1 and Table 2. A variation point that specifies alternative functionality in the use case describes several possible alternative functions, one of which could replace the functionality at the same step location in the use case. One of the alternatives may be used as a default. Depending on the variation point, one of the alternatives must always be selected; this alternative is referred to as a mandatory alternative. With the other kind of alternative, one possibility is not to select any of the alternatives; this alternative is referred to as an optional alternative.

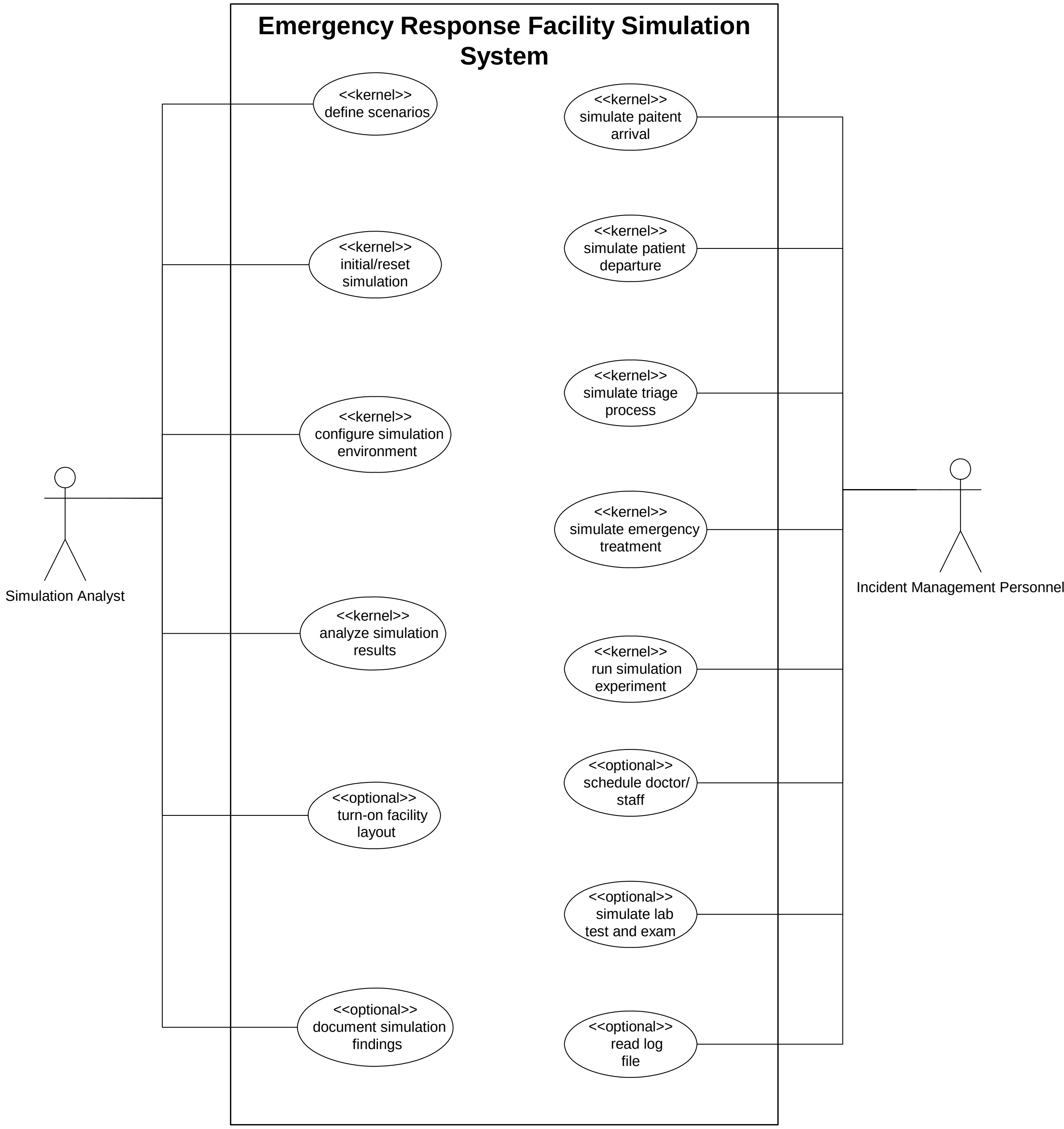

Figure 2: Use Case Model for Emergency Response Facility Simulation System Software Product Line

Table 1: Use Case for Simulate Patient Arrival

Use Case Name	**Simulate Patient Arrival**

ID	1
Reuse Category	Kernel
Summary	Patient arrival rate and other characteristics are being entered and simulated
Actors	Simulation user (Incident Management Personnel/Simulation Analyst)
Preconditions	• Simulation software is launched • Simulation model is loaded • Simulation scenario is defined
Description	1. Simulation user starts to run the simulation model 2. Simulation system prompts user to select type of patient from a list 3. Simulation user chooses the patient type 4. Simulation system prompts user to input number of patients 5. Simulation user inputs number of patients 6. Simulation system continues with the patient type and arrival rate
Alternatives	Line 2, 3, 4, and 5: based on the design, the user interface may vary, the way user inputs data may be different. User inputs are also different according to the scenario.
Post conditions	Patient arrival rate and type are entered into the system

Variation Point in the *Simulate Patient Arrival* Use Case

Name: Patient Arrival
Type of functionality: Mandatory alternative
Line number(s): 2, 3, 4, 5 of description
Description of functionality: The default prompt is a selection of patient type from the menu; an alternative is for the user to enter a patient code.

Table 2: Use Case for Simulate Lab Test and Exam

Use Case Name	**Simulate Lab Test and Exam**
ID	2
Reuse Category	Optional
Summary	Simulation of lab test and/or exam activities
Actors	Simulation user (Incident Management Personnel/Simulation Analyst)
Preconditions	• Simulation model is running • User interface is provided • There is an option to input or choose lab test/exam parameters
Description	1. Simulation prompts the input of parameters of test/exam

	2. User inputs parameters 3. Simulation model updates the input for the model 4. Simulation model continues with the lab test and/or exam functionality
Alternatives	Line 2: User skips the step, the simulation model runs without the option Line 3: Repeat 1 and 2 if the input is invalid
Post conditions	Simulation result about the lab test and/or exam is generated.

Variation Points in the *Simulate Lab Test and Exam* Use Case

Name: Lab Test
Type of functionality: Optional
Line number(s): 1 of description
Description of functionality: The lab test performs blood test, and urine test and generates Lab test result data. Only certain emergency response facilities such as hospital ER need to have this functionality.

Name: Exam
Type of functionality: Optional
Line number(s): 1 of description
Description of functionality: This includes a general examination and other necessary medical procedures using medical equipment such as x-ray, ultrasound, and MRI. Some emergency response facility such as a decontamination station does not need to have the exam functionality.

3.3 Feature Modeling

A commonality/variability analysis is performed to identify common, optional, and alternative features in the ERFSSSPL. The feature model is then developed based on the analysis result. The common features analysis identifies the common functionality in the product line; the optional and alternative features represent the variability in the product line as specified by the optional use cases and the variation points. The alternative feature must be one out of a group of alternatives. If an alternative is not chosen, the default is used. The relationship between the features and the use case are depicted in Table 3.

Table 3: Feature to Use Case Relationships

Feature Name	Feature Category	Use Case Name	Use Case Category / Variation Point (VP)	Variation Point Name
ERFSSSPL Kernel	Common	Define scenarios Initial/reset simulation Configure simulation	Kernel	

		environment Run simulation experiment Analyze simulation results Simulate patient arrival Simulate patient departure Simulate triage process Simulate emergency treatment		
Emergency patient	Default	Simulate patient arrival	VP	Patient arrival
Walk-in patient	Optional	Simulate patient arrival	VP	Patient arrival
Discharge	Default	Simulate patient departure	VP	Patient departure
Admitted	Optional	Simulate patient departure	VP	Patient departure
Re-directed	Optional	Simulate patient departure	VP	Patient departure
2D animation	Default	Configure simulation environment	VP	Visualization
3D animation	Alternative	Configure simulation environment	VP	Visualization
With animation	Default	Configure simulation environment	VP	Simulation display type
Without animation	Alternative	Configure simulation environment	VP	Simulation display type
Decontamination	Optional	Simulate emergency treatment	VP	Decontamination
Exam	Optional	Simulate lab test and exam	Optional	
Lab test	Optional	Simulate lab test and exam	Optional	
Log file	Optional	Read log file	Optional	
Recommendation	Optional	Document simulation findings	Optional	
Schedule change	Optional	Schedule doctor/staff	Optional	
General exam	Default	Simulate emergency treatment	VP	General exam
Trauma exam	Optional	Simulate emergency treatment	Optional	
Cardiac exam	Optional	Simulate emergency treatment	Optional	
OB/Gyn exam	Optional	Simulate emergency treatment	Optional	
Orthopedic exam	Optional	Simulate emergency	Optional	

		treatment		
Pediatric exam	Optional	Simulate emergency treatment	Optional	

The common features are determined based on the problem description and the kernel use cases. The following definitions are samples of the functionality that every emergency response simulation system must have

- **Patient:** A patient is someone who is injured or sick when the incident happens. The injury can be trauma or disease, in either critical or minor condition.
- **Doctor:** A doctor can be a general practitioner, cardiologist, orthopedist, pediatrician, or obstetrician/gynecologist.
- **Nurse:** A nurse is a medical staff member who is trained to assist doctors in patient assessment and treatment.
- **Help desk:** Help desk is a station that receives patients and prepares patient records, or a meeting place for doctors and nurses.
- **Waiting area:** Waiting area is a place for patients to queue up and wait for treatment.
- **Triage:** Triage is a process for sorting injured patients into groups based on their conditions and needs for or likely benefit from immediate medical treatment. Triage is used in hospital emergency rooms and at disaster sites when limited medical resources must be allocated.
- **Schedule of resources:** Schedule of resources is the planned time that the resource is on duty or occupied.
- **Distribution:** A probability distribution is used to estimate patient arrival rate (number of arrivals per unit of time) and treatment process time (duration of the treatment).
- **User interface:** A user interface is an interactive dialog window that allows the user to input, select, or change simulation parameters for different scenarios.

Some of the optional/alternative features are described as follows:

- **Emergency patient:** A patient who is in a critical condition and needs to be treated immediately, such as trauma and cardiac patients. Typically, this patient is brought in by an ambulance.
- **Walk-in patient:** A patient who is injured or sick but not with a life-threatening condition. The patient enters the hospital, proceeds through a triage process for an initial assessment of his condition.
- **Discharge:** After initial medical treatment, the patient is released.
- **Admitted:** After initial medical treatment, the patient is kept at the hospital for further evaluation and treatment, and transferred to an available room for a longer stay.
- **Re-directed:** After initial medical treatment, the patient is redirected to an alternate facility if all the rooms are occupied, a special required technology or equipment is not available in this facility.
- **Exam:** The assessment or diagnosis of the patients usually with the help of medical equipment such as x-rays or ultrasound.
- **Patient record:** Patient record is a patient file that maintains the patient's information including medical history, symptoms, treatment, and medications.

- **Decontamination:** A type of treatment used to make the patient safe by eliminating poisonous or otherwise harmful substances, such as noxious chemicals or radioactive material.
- **Lab test:** The assessment or diagnosis of the patients by analysis of blood, urine, or other bodily fluid or specimen.

Figure 3 is a feature-dependency model that describes how one feature depends on other features. There are stereotypes for the different kinds of features, such as <<optional feature>> and <<alternative feature>>. The root of the feature-dependency hierarchy is the common feature ERFSSSPL kernel, which every emergency response simulation system must carry. Several optional features that might be used by some members of ERFSSSPL are added to the ERFSSSPL kernel, include *Decontamination*, *Exam*, *Lab test*, *Log file, Recommendation,* and *Schedule change*. Related features are grouped into feature groups, which place a constraint on how the features are used by a given member of ERFSSSPL. Feature groups can be identified by means of stereotypes, such as <<exactly-one-of feature group>> and <<at-least-one-of feature group>>. In Figure 3, three *at-least-one-of feature* groups depend on ERFSSSPL kernel: they are *Specialist, Patient arrival, and Patient departure*. Two *exactly-one-of feature* groups are *Simulation display type* and *Visualization*. Each of these groups has two or more features. For example, the *Patient departure* feature group has the default *Discharge* feature and the optional *Admitted* and *Re-directed* features. For those *exactly-one-of feature* groups, the default and alternative features are mutually exclusive.

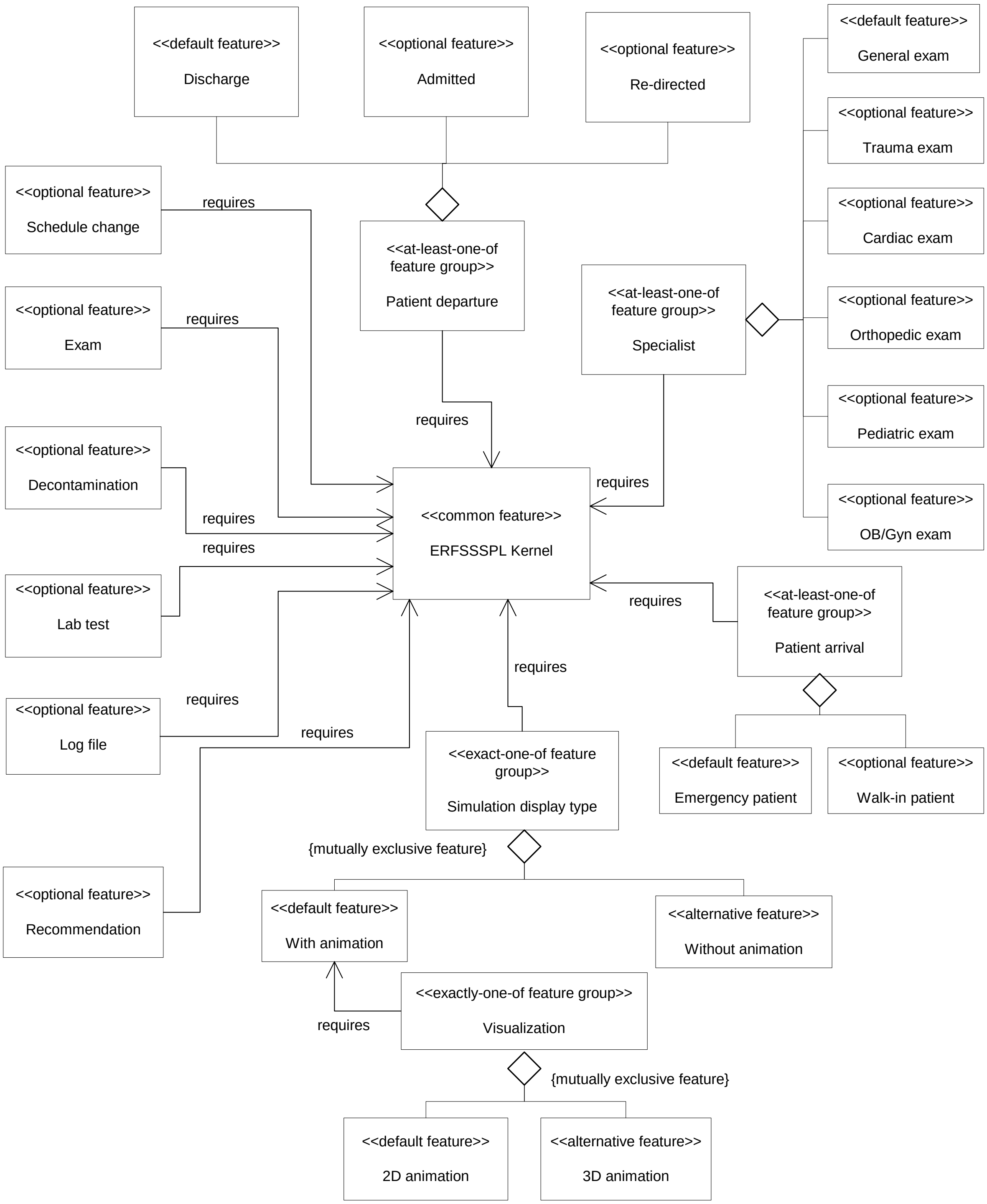

Figure 3: Feature Dependency Model for Emergency Response Facility Simulation Software Product Line System

3.4 Static Modeling

A static model of the ERFSSSPL is depicted in the entity class diagram shown in Figure 4. All the major entities, their attributes, and the relationships among these entities are presented. Because this static model depicts only entity classes, all the classes have the <<entity>> stereotype. In the product line entity class model, classes have a second stereotype to depict the reuse category of each class: <<kernel>> or <<optional>>.

The *Emergency_center* class has four subclasses: *Staff*, *Facility*, *Material*, and *Treatment*, In which *Staff* has three subclasses *Doctor*, *Nurse*, and *Tech*. *Facility* has five subclasses *Lab*, *Help_desk*, *Waiting_area*, *Bed*, and *Exam_rm*, where the *Exam_rm* has two subclasses *Triage_rm* and *Radiology_rm*. *Material* has subclasses *Medication* and *Linen*. *Treatment* is provided to the *Patient*. Each *Staff* has working *Schedules*. *Patient* has *Patient_record*, *Arrival* (superclass of *Emergency* and *Walk_ in*), and *Release* (superclass of *Go_home*, *Admitted*, and *Reroute*). The *Simulation_engine* provides *S_clock*, *Animation*, *User_interface*, simulation *Result*, and probability distribution (*P_distribution*). Simulation engine is provided by simulation software.

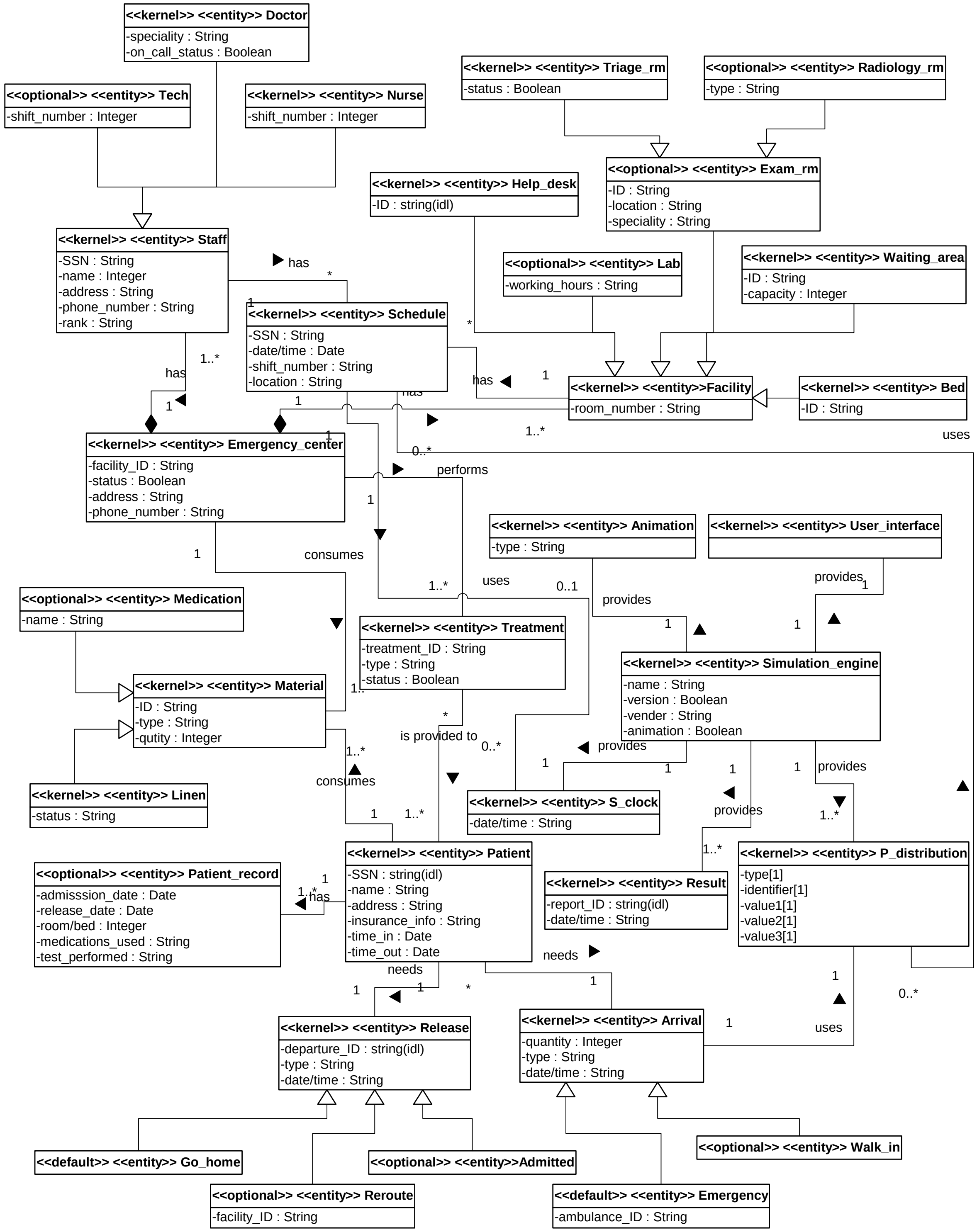

Figure 4: Static Model for Emergency Response Facility Simulation System Software Product Line

3.5 Dynamic Modeling

With the kernel-first approach [16], a set of kernel communication diagrams is developed based on the kernel use cases. For those state-dependent use cases, a set of state charts, which describes the state scenarios for the state-dependent object - *Simulation control*, is also developed. For example, Figure 5 is the communication diagram and Figure 6 is the state chart for the *simulate patient arrival* use case, with the default object chosen where such a choice is required. The simulation user here may be a S*imulation Analyst* or an *Incident Management Personnel*.

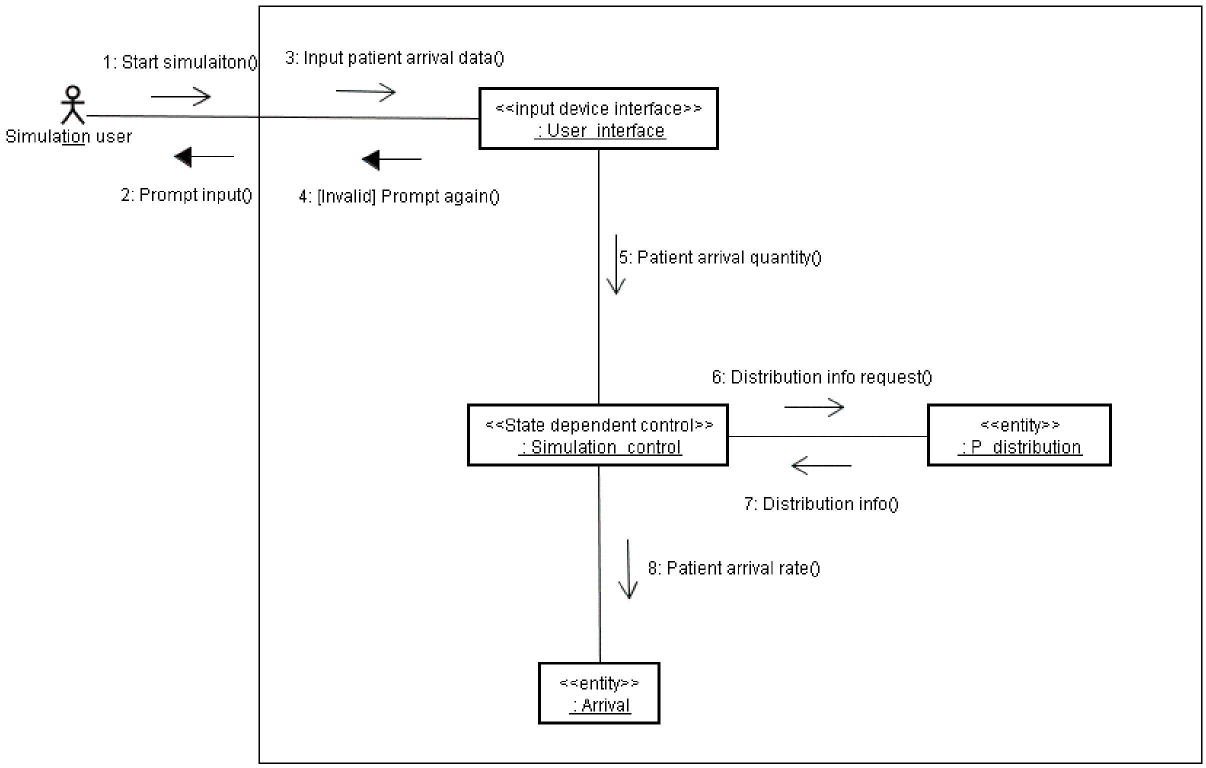

Figure 5: Communication Diagram for Kernel Use Case: Simulate Patient Arrival Use Case

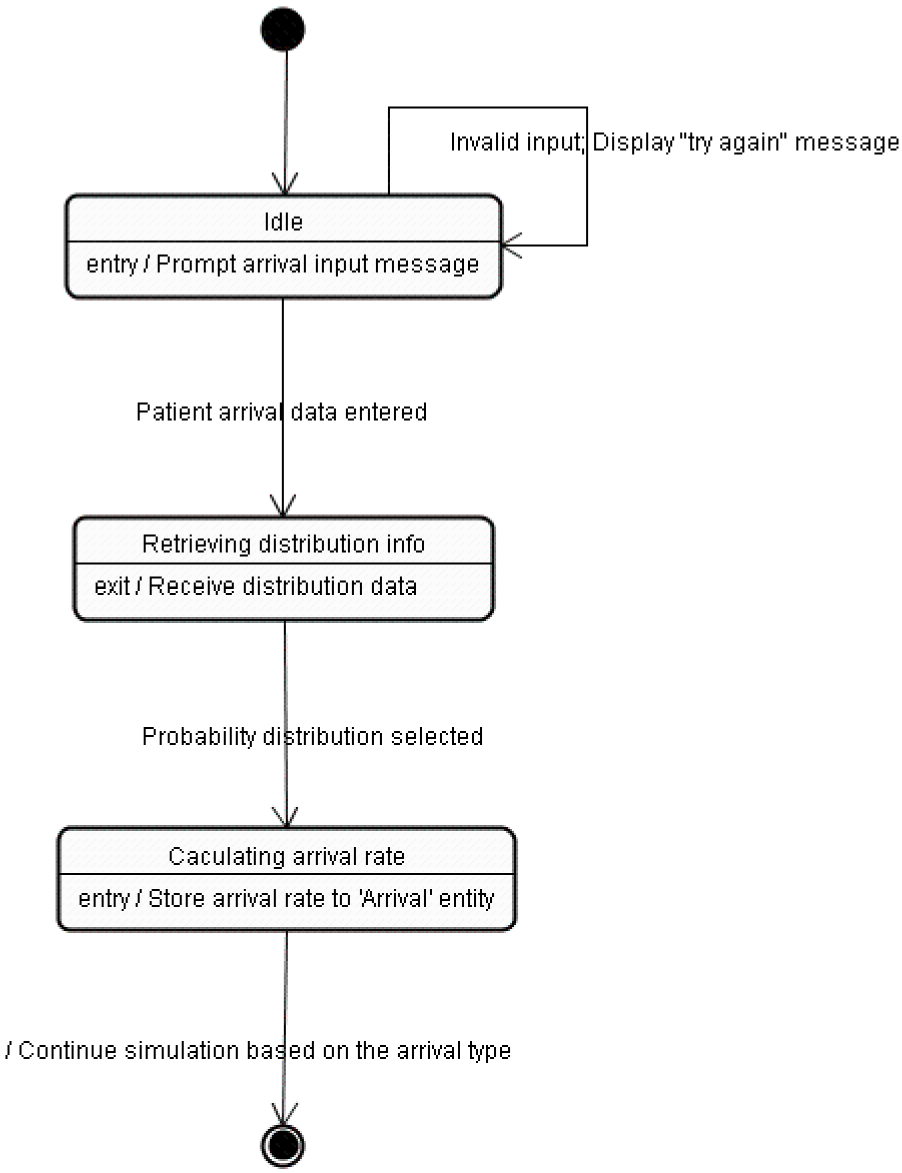

Figure 6: State Chart for Simulation Control: Simulate Patient Arrival Use Case

3.6 Feature/class Dependency Modeling

Table 4 depicts the feature/class dependencies; it shows the features and classes that realize the functionality described by the feature. For example, *Walk-in patient* is an optional feature, supported by two classes: *Walk_in* and *Simulation_Control* where *Walk_in* is an optional class, and *Simulation_Control* is a kernel-parameterized class. The feature condition is stored as a Boolean class attribute of *Simulation_Control* called *walkin*, which is set to true or false, depending on whether the feature is selected (walkin =True) or not selected (walkin = False).

Table 4: Feature/class Dependency Model

Feature Name	Feature Category	Class Name	Class Reuse Category	Class Parameter
ERSSPLS Kernel	Common	Patient Arrival Release Patient_record P_distribution Facility Staff Schedule Material Simulation_engine Result S_clock User_interface Simulation control	kernel kernel-abstract-vp kernel-abstract-vp kernel kernel kernel-abstract-vp kernel-abstract-vp kernel kernel-abstract-vp kernel kernel kernel kernel-param-vp kernel-param-vp	
Emergency patient	Default	Emergency Simulation control	default kernel-param-vp	emergency:Boolean
Walk-in patient	Optional	Walk_in Simulation control	optional kernel-param-vp	walkin:Boolean
Discharge	Default	Go_home Simulation control	default kernel-param-vp	discharge:Boolean
Re-directed	Optional	Reroute Simulation control	optional kernel-param-vp	redirected:Boolean
Admitted	Optional	Admitted Simulation control	optional kernel-param-vp	admitted:Boolean
2D animation	Default	Animation	kernel-param-vp	type: 2D
3D animation	Alternative	Animation	kernel-param-vp	type: 3D
With animation	Default	Simulation_engine	kernel-param-vp	animation:Boolean
Without animation	Alternative	Simulation_engine	kernel-param-vp	animation:Boolean

Exam	Optional	Exam_rm Simulation control	optional kernel-param-vp	exam:Boolean
Lab test	Optional	Lab Simulation control	optional kernel-param-vp	labtest:Boolean
Log file	Optional	Result Simulation control	optional kernel-param-vp	logfile:Boolean
Recommendation	Optional	Result Simulation control	optional kernel-param-vp	recommendation:Boolean
Schedule change	Optional	Schedule Simulation control	optional kernel-param-vp	schedule_change: Boolean
Decontamination	Optional	Treatment Simulation control	kernel-param-vp kernel-param-vp	type:decontamination decontamination: Boolean
General exam	Default	Doctor Simulation control	kernel-param-vp kernel-param-vp	specialty: General generalexam:Boolean
Orthopedic exam	Optional	Doctor Simulation control	kernel-param-vp kernel-param-vp	specialty: Orthopedic orthopedicexam: Boolean
Pediatric exam	Optional	Doctor Simulation control	kernel-param-vp kernel-param-vp	Specialty: Pediatric pediatricexam:Boolean
OB/Gyn exam	Optional	Doctor Simulation control	kernel-param-vp kernel-param-vp	specialty: OB/Gyn OB/Gynexam:Boolean
Trauma exam	Optional	Doctor Simulation control	kernel-param-vp kernel-param-vp	specialty: Trauma traumaexam:Boolean
Cardiac exam	Optional	Doctor Simulation control	kernel-param-vp kernel-param-vp	specialty: Cardiac cardiacexam:Boolean

3.7 Design Modeling

The ERFSSSPL is designed as a component-based software architecture based on the Centralized Control architecture pattern [16]. One control component provides overall control of the system, receives messages from input components that contain events causing the control component to change state and send action messages to output components. Figure 7 shows the Centralized Control architectural pattern for the ERFSSSPL, in which the concurrent components are depicted on a generic communication diagram. The *SimulationControl* component is a centralized control component, which executes the state chart that provides the overall control and sequencing for the simulation. *SimulationControl* receives messages from two input components – *UserInterface* and *InfoSystemInterface*, when either component detects input from the external environment. *SimulationControl*'s actions are sent to the output component *DisplayInterface*, the action may be to prompt for user input or to display simulation animation as well as the simulation result. The type of system configuration is determined when the system is deployed, which is not discussed in this paper. The component architecture is developed starting with the design of the kernel system, which contains the kernel and default components, and then the message communication between components. The processes are repeated for the whole product line, until finally the optional and variant components are added.

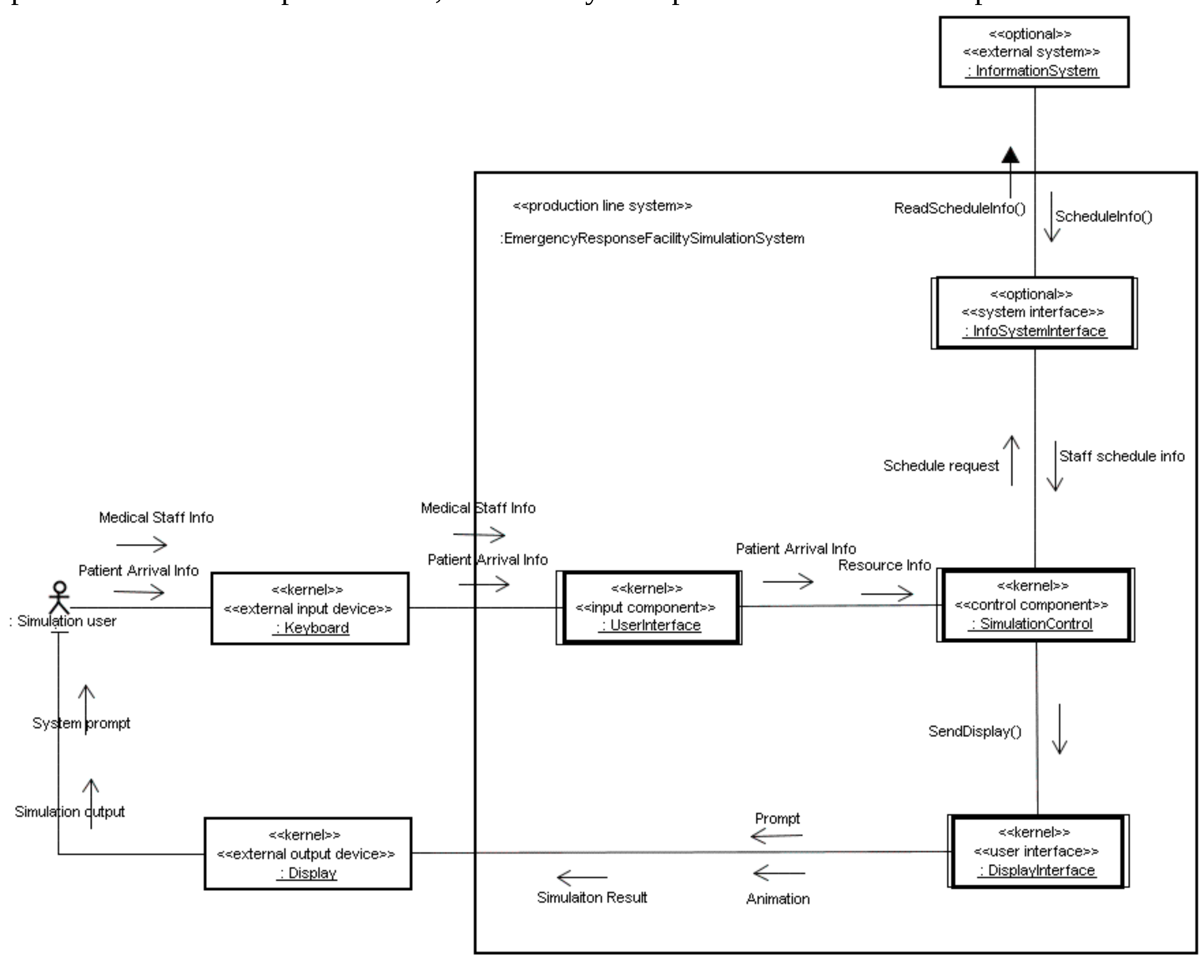

Figure 7: Centralized Control Architecture Pattern for ERFSSSPL

4. A Prototype Emergency Room Simulation System

This section presents a prototype of a hospital ER simulation system (the ER model) that is a software application derived from the ERFSSSPL discussed in previous section. A hypothetical emergency incident scenario has been developed by the National Institute of Standards and Technology for demonstrating the applicability of integrated simulation and gaming for incident management. A number of simulation and gaming modules have been utilized to model the major aspects of the hypothetical scenario. The modules demonstrate the value of utilizing simulation for incident management applications. They can be used to highlight the value of training using simulation. The scenario is based on a dirty bomb attack in Washington DC on the evening of July 4. The fireworks on the National Mall on July 4 attract a large crowd. The scenario did not consider the feasibility or means of getting a dirty bomb to the identified location. The probability of such an occurrence is expected to be very low with the typical high security surrounding such an event. The focus of the scenario was on the consequences if such an incident occurs. The near term consequences of a dirty bomb explosion include the casualties and radiation exposure among the crowd in the immediate vicinity and in the area covered by the plume, and response by police, fire department and emergency medical technicians. The major consequences of the incident and the response need to be modeled for incident management purposes [27]. The ER model simulates the actions of the hospital emergency room near by in response to such emergency incident. The operation of hypothetical emergency room for handling the casualties from the incident is simulated. Casualties arriving at the emergency room include serious cases of trauma and cardiac cases brought in by ambulances and walk-ins with minor injuries and the worried well.

This prototype ER simulation system is a discrete-event-simulation model of an emergency patient's flow through a hospital. The purpose of this simulation is to provide a small but realistic model of resources, patient flow, and congestion in the ER of the hospital. The model will demonstrate how the incident affects the dispatch of emergency vehicles to transport the injured to hospital and the waiting time in different areas. The model will also evaluate the resources needed. The simulation will train incident management personnel and/or hospital management teams to respond to the crises that affect ER flow. By trying out scenarios, the simulation can help the team to evaluate the impact of their decisions on the deployment of resources, actions for triage, treatment of the injured, and transfer patients to other hospitals if needed. The prototype simulation model includes trauma rooms, cardiac rooms, specialty treatment rooms, and the general and ambulance entrances for patient arrival. The model uses an interactive user interface, allowing the user to make modifications to the model parameters. The user may change the patient arrival quantity and the number of trauma and cardiac patients. Users are allowed to add specialists such as pediatricians, orthopedic surgeons, and obstetrician/gynecologists as needed. The arrival of a critical patient such as a cardiac or trauma patient will use extra resources, which normally will cause a backlog of non-critical patients. A scale layout of an ER is used as a background for the model. A bird's-eye view of treatment areas and the flow of patients, staff and support departments such as patient transport are modeled. Required data of the ER's operations are loaded into the model. The model then can be used to analyze factors such as the arrival of the patients, staffing levels, room layout design and utilization, treatment times at each station, ancillary departments such as lab and imaging, physician practice patterns, and other variables that have an impact on ER operations. The primary entities in the simulation model are patients, medical staff, emergency vehicles, and other resources. Ambulatory patients are modeled as first-in first-out queues.

The procedures of hospital ER simulation system development are as follows:

- Determine the scope of the ER model within the product family.
- Identify and acquire the data required to build the ER model, such as hospital layout map, number of medical staff, and schedule of the doctors' shifts.
- Determine what parameters may be manipulated externally while the ER model is running.
- Determine the architecture of the ER model based on the product line simulation models by considering the commonality and variability.
- Construct a preliminary design of the ER model and its graphical interface using SPL approach that allows the system be expanded or adapted to other settings.
- Test and improve the hospital simulation system model.

4.1 SPL Architecture

The feature model, use case model, analysis model, and design model (as shown in Figures 1-7 and Tables 3-4) in the SPL are adapted to derive the architecture of the ER model (as shown in Figures 8-11 and Tables 6-7). Due to limitations on paper length, not all detailed models are presented here. All the kernel features and all the optional features except the *Log file, Recommendation, Schedule change,* and *Decontamination* are included in the ER model. The simulation engine is ProModel, a discrete-event-simulation software that provides 2D animation [28] [29]. At run time, the *With animation* feature can be turned on. Table 5 presents mappings between the models of the SPL and the ER model. The diagram/table pairs have the same structure; however, some items in the features, use case, and entity models are eliminated when applied onto the ER model since these items are not needed. For example, Figure 9 depicts the use case model for the ER model; it is derived based on the SPL use case model shown in Figure 2 with the selected use cases presented in solid line and the unselected use case presented in dotted line.

Table 5: Diagrams and Tables Mapping between the SPL and the ER Model

	Context Diagram	**Use Case Model**	**Feature Dependency Model**	**Static Diagram**	**Feature to Use Case Relationships**	**Feature/Class Dependency Diagram**
SPL	Figure 1	Figure 2	Figure 3	Figure 4	Table 3	Table 4
ER	Figure 10	Figure 9	Figure 8	Figure 11	Table 6	Table 7

In the following sections, the ER model input, output, and logic are discussed.

- As showed in Figure 10, the context model of the ER model is derived from the context model of the ERFSSSPL by removing the optional external system "Information System". It is used to define the boundary of the ER model.
- The *use case* model (Figure 9) and the *feature dependency* model (Figure 8) of the ER model defined the functional requirements and characteristics. These models not only determine the functional logic of the ER model, but also determine the contents and formats of inputs and outputs for the ER model.

- The *static* model (Figure 11) is a structural view of the information aspects of the ER model. The class, their attribute and the relationships are used to model the operation logic of the ER model.
- The *feature to use case relationships* model (Table 6) and the *feature/class dependency* model (Table 7) are used together with the *use case* model (Figure 9), and the *feature dependency* model (Figure 8) to further determine the detail logic for the ER model.

Production in software product lines can be fully automated, completely manual, or mixture of the two. If appropriate UML modeling tool, simulation software, and programming language are selected, the models introduced above can be used for automatic (or at least semi-automatic) simulation model generation. This is also part of potential future work.

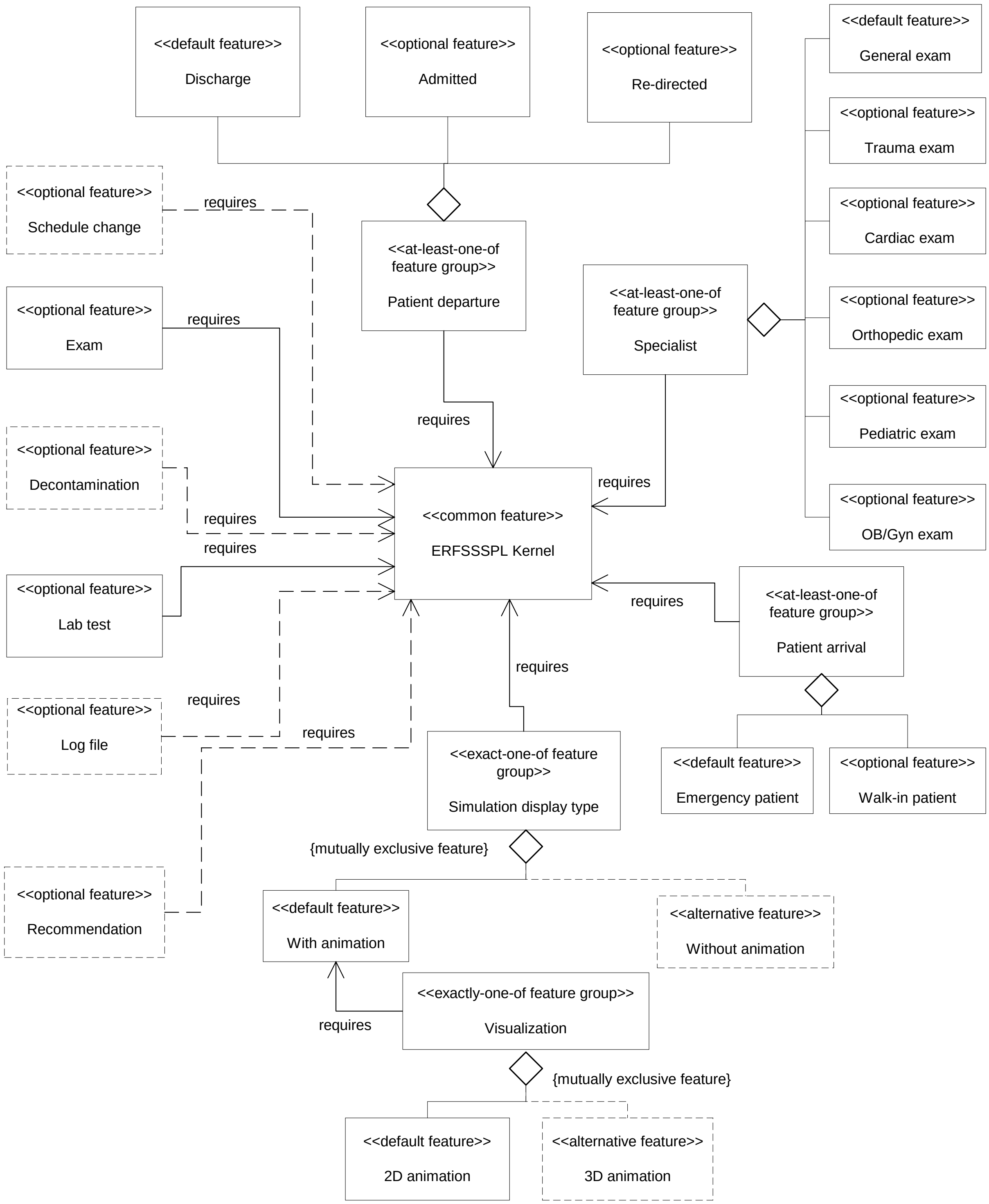

Figure 8: Hospital Emergency Room Application: Feature Dependency Model

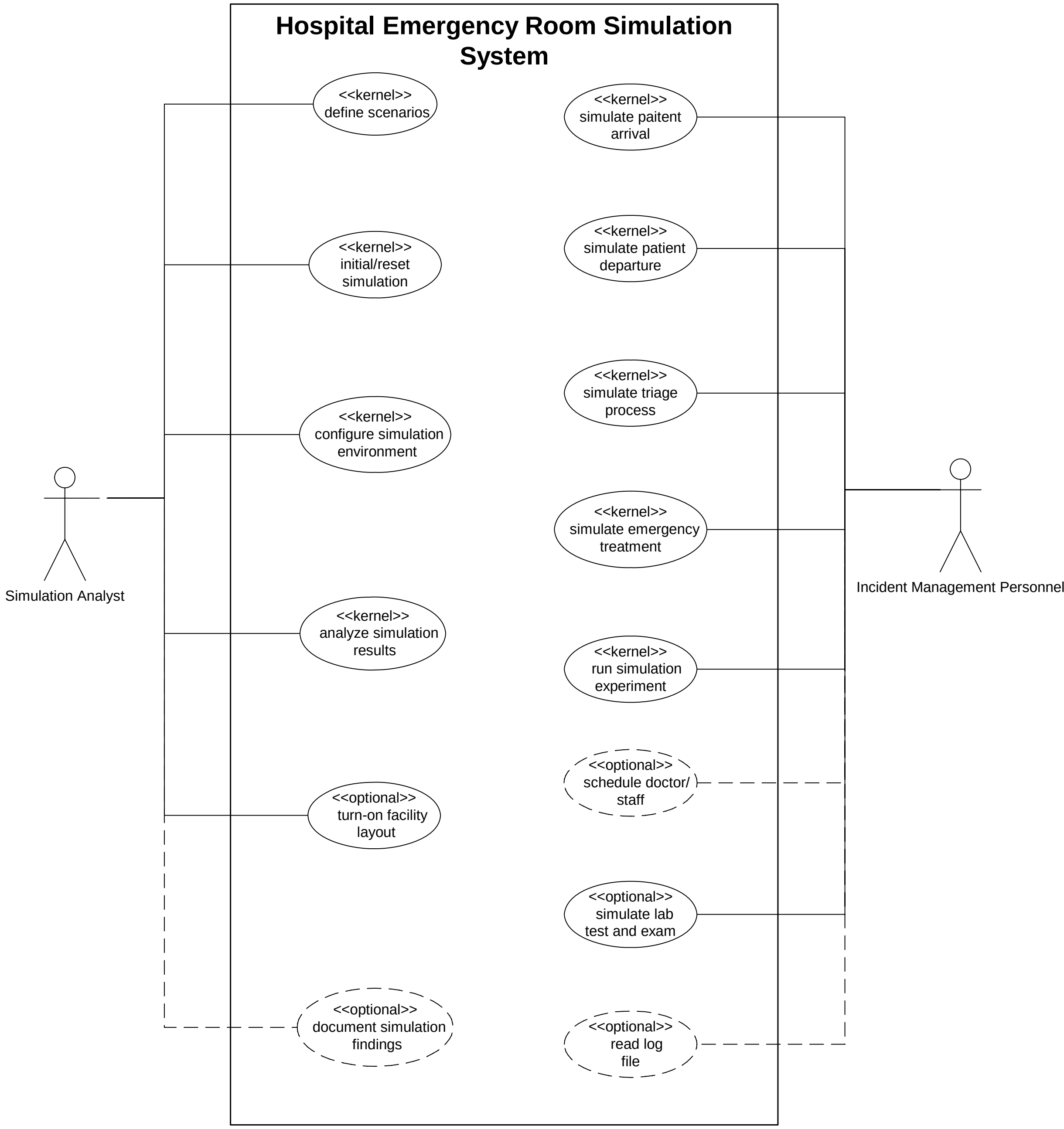

Figure 9: Hospital Emergency Room Application: Use Case Model

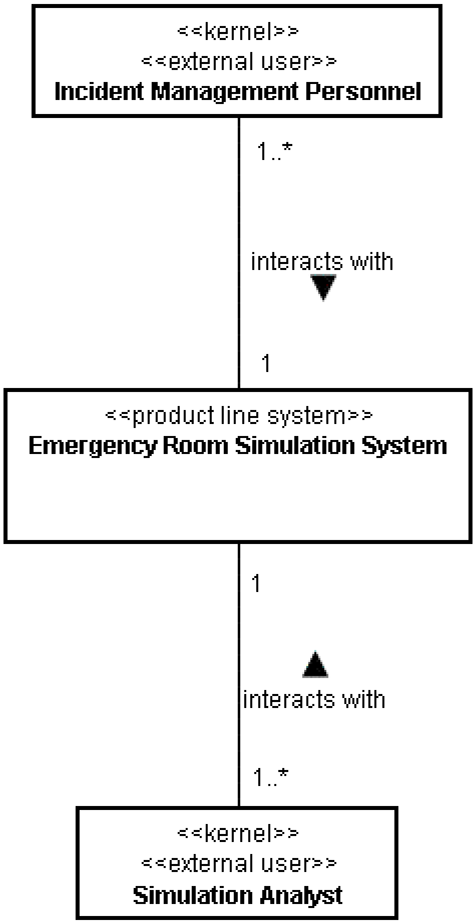

Figure 10: Hospital Emergency Room Application: Context Diagram

Table 6: Hospital Emergency Room Application: Feature to Use Case Relationships

Feature Name	**Feature Category**	**Use Case Name**	**Use Case Category / Variation Point (VP)**	**Variation Point Name**
ERSSPLS Kernel	Common	Define scenarios Initial/reset simulation Configure simulation environment Run simulation experiment Analyze simulation results Simulate patient arrival	Kernel	

		Simulate patient departure Simulate triage process Simulate emergency treatment		
Emergency patient	Default	Simulate patient arrival	VP	Patient arrival
Walk-in patient	Optional	Simulate patient arrival	VP	Patient arrival
Discharge	Default	Simulate patient departure	VP	Patient departure
Admitted	Optional	Simulate patient departure	VP	Patient departure
Re-directed	Optional	Simulate patient departure	VP	Patient departure
2D animation	Default	Configure simulation environment	VP	Visualization
With animation	Default	Configure simulation modes	VP	Simulation display type
Exam	Optional	Simulate lab test and exam	Optional	
Lab test	Optional	Simulate lab test and exam	Optional	
General exam	Default	Simulate emergency treatment	VP	General exam
Trauma exam	Optional	Simulate emergency treatment	Optional	
Cardiac exam	Optional	Simulate emergency treatment	Optional	
OB/Gyn exam	Optional	Simulate emergency treatment	Optional	
Orthopedic exam	Optional	Simulate emergency treatment	Optional	
Pediatric exam	Optional	Simulate emergency treatment	Optional	

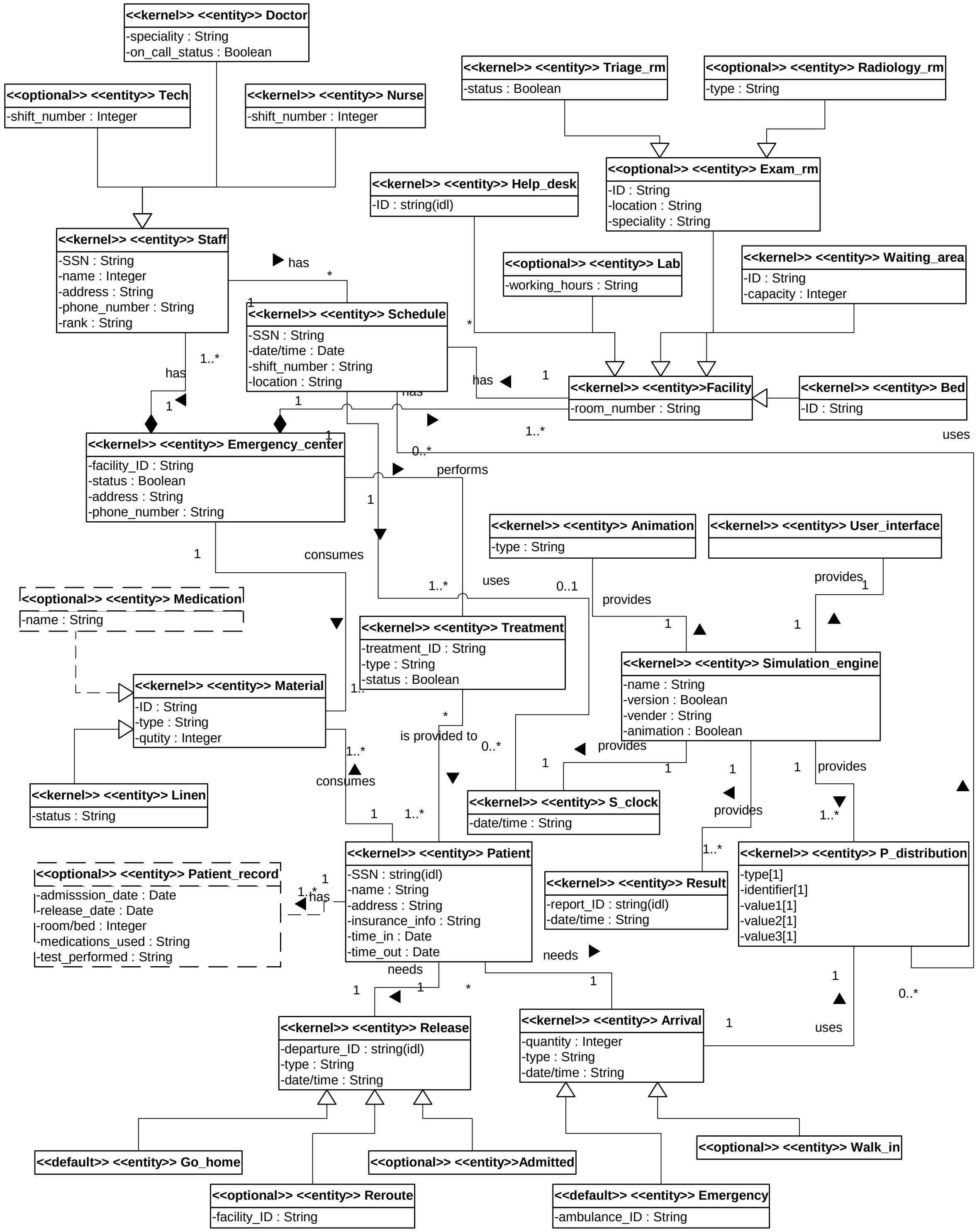

Figure 11: Hospital Emergency Room Application: Static Model

Table 7: Hospital Emergency Room Application: Feature/Class Dependency Model

Feature Name	Feature Category	Class Name	Class Reuse Category	Class Parameter
ERSSPLS Kernel	Common	Patient Arrival Release Patient_record P_distribution Facility Staff Schedule Material Simulation_engine Result S_clock User_interface Simulation_control	kernel kernel-abstract-vp kernel-abstract-vp kernel kernel kernel-abstract-vp kernel-abstract-vp kernel kernel-abstract-vp kernel kernel kernel kernel-param-vp kernel-param-vp	
Emergency patient	Default	Emergency Simulation_control	default kernel-param-vp	emergency:Boolean
Walk-in patient	Optional	Walk_in Simulation_control	optional kernel-param-vp	walkin:Boolean
Discharge	Default	Go_home Simulation_control	default kernel-param-vp	discharge:Boolean
Re-directed	Optional	Reroute Simulation_control	optional kernel-param-vp	redirected:Boolean
Admitted	Optional	Admitted Simulation_control	optional kernel-param-vp	admitted:Boolean
2D animation	Default	Animation	optional-param-vp	type: 2D
With animation	Default	Simulation_engine	kernel-param-vp	animation:Boolean
Exam	Optional	Exam_rm Simulation Control	optional kernel-param-vp	exam:Boolean
Lab test	Optional	Lab Simulation_control	optional kernel-param-vp	labtest:Boolean
General exam	Default	Doctor Simulation_control	kernel-param-vp kernel-param-vp	specialty: General General exam:Boolean
Orthopedic exam	Optional	Doctor Simulation_control	kernel-param-vp kernel-param-vp	specialty: Orthopedic Orthopedic exam:Boolean
Pediatric exam	Optional	Doctor Simulation_control	kernel-param-vp kernel-param-vp	specialty: Pediatric Pediatric exam:Boolean
OB/Gyn exam	Optional	Doctor	kernel-param-vp	specialty: OB/Gyn

		Simulation_control	kernel-param-vp	OB/Gyn exam:Boolean
Trauma exam	Optional	Doctor Simulation_control	kernel-param-vp kernel-param-vp	specialty: Trauma Trauma exam:Boolean
Cardiac exam	Optional	Doctor Simulation_control	kernel-param-vp kernel-param-vp	specialty: Cardiac Cardiac exam:Boolean

4.2 Model Inputs

The input data of the ER model include:

- The layout of a hospital ER
- Number, location, and type of casualties
- Availability of staff at work and off
- Availability of resources
- Time and resources required for attending to each casualty type
- Hospital location
- Process stations
- Station capacities
- Station processing times
- Patient arrival rate
- Hospital shifts
- Medical resources
- Symptom-treatment profiles

4.3 Model Outputs

The output data of the ER model include:

- Status of the patient
- System utilization
- Resources utilization
- Number of people treated, released, admitted, dead, and waiting for treatment
- State of the staff and facilities (to determine their capability to deal with another incident)
- Run time interactions
- Simulation clock time

4.4 Model Logic

The ER model logic includes relevant policies and procedures for emergencies including calling in medical staff, using temporary accommodations for the injured, and acquiring needed supplies and equipment. Figure 12 depicts the model overview. Patients arrive in the ER either through the general entrance or via ambulance. An ambulance is for those patients who are in critical condition, such as trauma and cardiac. The hospital groups patients into different categories based on the severity of their injuries. Trauma or cardiac patients are considered the most critical and need to be treated immediately. Non-critical patients go through a triage process whereby the hospital makes an initial assessment of their injuries. The patient is then transferred to an available room for further exam and treatment. They go through a registration process either before or after the treatment, depending on the severity of their injury. After initial medical treatment, the hospital can release the patient or assign the patient to a room in the hospital for a longer stay. Because of the limitations of the hospital's space and equipment, the patient may be redirected to an alternate facility.

After a patient is taken into the examination room, the nurse and technician start the initial assessment or treatment. The doctors then examine the patient and make decisions. The patient is moved to the Nurse Unit when all the necessary procedures are carried out. Generally, most patients are ambulatory and can walk into the hospital, where they are initially triaged and sent to the waiting area to anticipate being called into one of the various exam rooms. Different categories of exam rooms cater to the type of exam that will take place, such as general physical exam, orthopedic exam, pelvic exam, pediatric exam, and critical care. If the condition is not critical, the patient can be discharged after the initial treatment. If further emergent tests or radiological exams are indicated, patients enter the queue for these procedures. Figure 13 details the ambulance entrance of the ER.

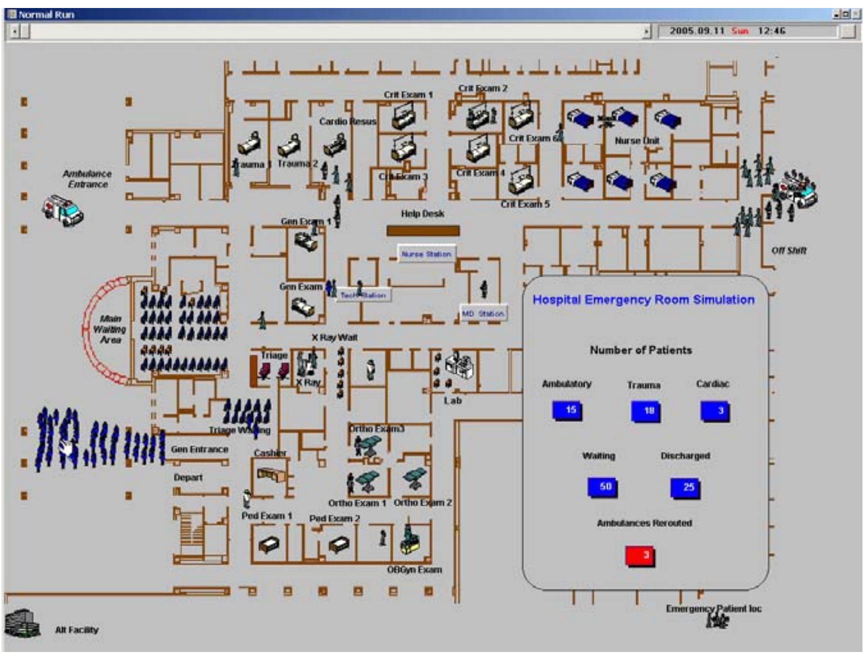

Figure 12: Hospital Emergency Room Simulation - Model Overview

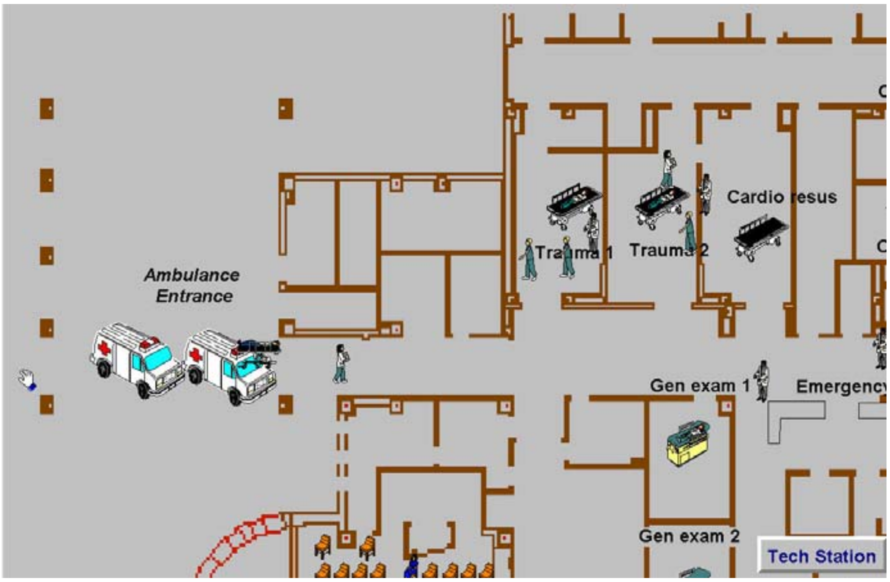

Figure 13: Hospital Emergency Room Simulation Model - Ambulance Entrance

4.5 Analysis of the Simulation Result

Through the analysis of the statistical results of the simulation output, the simulation model of the hospital ER can help management to find potential problems in current methods. Incident management personnel can use the simulation model to look at different possible scenarios. For instance, they can tell the model how long it takes to perform particular steps in the triage processes, where the processes take place, and how many staff members are available. Simulation can show them visually the impact they have on the ER's processes, as they can:

- Run the model and observe the values calculated on-screen.
- Examine the statistical results and plot/chart of the probabilities for different scenarios.
- Discuss the conclusions as well as any recommendations for improving the models performance.
- Lessons learned from the simulation allow a better decision to be made in case of a real emergency. This can result in a better design of the hospital ER, preventing an unwise large investment on a not-optimal design.

5. Conclusions

There is a growing need for preparedness for emergency response both for man-made and natural disaster events. The man-made disaster risk has increased due to a rise in possibility of terrorist attacks against the United States. Effective emergency response presents a number of challenges to the responsible agencies. Simulation-based acquisition is a powerful tool for developing functional solutions to handle crises and training solutions that are essential to prepare people for situations that go far beyond normal. Simulations would support decision-making through all phases of incident management, including prevention, preparedness, response, recovery, and mitigation.

Since simulations are often developed from scratch, there is little opportunity for the analyst to build upon the work of others since each simulation is built as a custom solution to a uniquely defined problem. A better solution would be to simplify the design through re-usable processes. The reusability will help to reduce the costs associated with simulation model construction and thus make simulation technology more affordable and accessible to a wide range of potential emergency responders. Most importantly, it will provide better training to incident management personnel from different responders.

This paper has proposed applying the SPL technology to develop simulation-based training modules for supporting incident management. The beauty of the SPL is to capitalize on commonality and manage variation in order to reduce the time, effort, cost, and complexity of creating and maintaining a product line of similar software systems. Companies such as Nokia, Hewlett-Packard (HP), LSI Logic, Philips, and Cummins have successfully applied the SPL techniques [30]. The SEI case studies of the SPL adoptions provided several successful stories [31] [32].

This paper discussed the requirements, analysis, and design models of a family of simulation systems for emergency response facilities. The modeling of ERFSSSPL was performed based on the Product Line UML-based Software engineering (PLUS) method. UML was used to describe components during the requirements, analysis, and design stages, and to capture their characteristics and relationships.

Considering that hospitals have little practice in dealing with emergencies due to incidents such as terrorist attack or natural disasters, making simulation-based training in such cases especially important. This paper demonstrated reusability of SPLs by presenting a prototype of the SPL member system; a hospital ER simulation system that was derived from the SPL design. The prototype demonstrated how easily the SPL member model could be derived once the SPL architecture had been built. Future work is intended to use SPL to build applications for various emergency response activities that support incident management for different scenarios.

6. Disclaimer

7. References

[1] United States Government Accountability Office. *Emergency Preparedness and Response, Some Issues and Challenges Associated with Major Emergency Incident: Testimony before the Little Hoover Commission, State of California.* GAO-06-467T, February 23, 2006.

[2] Jain, Sanjay, and Charles R. McLean. "Modeling and Simulation of Emergency Response: Workshop Report, Relevant Standards and Tools." National Institute of Standards and Technology, NISTIR-7071. 2003.

[3] Clements, Paul, and Linda Northrop. *Software Product Lines: Practices and Patterns*, Boston: Addison-Wesley Professional, 2002.

[4] Booch, Grady, James Rumbaugh, and Ivar Jacobson. *Unified Modeling Language User Guide 2nd ed.*, Boston: Addison-Wesley Professional, 2005.
[5] Office of the Press Secretary: "Homeland Security Presidential Directive/Hspd-8." The White House. http://www.whitehouse.gov/news/releases/2003/12/20031217-6.html.
[6] Coolahan, James E. "Planning for an Integrated M & S Framework for Catastrophic Event Response." http://www.pacercenter.org/pdf/coolahanPublication.pdf.
[7] PACER. "National Center for the Study of Preparedness and Catastrophic Event Response." Center of Excellence for the Study of Preparedness and Catastrophic Event Response (PACER). http://www.pacercenter.org/pages/research_rn.aspx.
[8] CFR. "Responding to a Terrorist Attack: Hospital Emergency Rooms." Council on Foreign Relations (CFR). http://www.cfr.org/publication/10199/ responding_to_a_terrorist_attack.html.
[9] Khan, Ali S., and Michael J. Sage. "Biological and chemical terrorism: Strategic Plan for Preparedness and Response." http://www.cdc.gov/mmwr/preview/ mmwrhtml/rr4904a1.htm.
[10] Institute of Medicine, The National Academies. *Risk Communication: National Security and Public Health:Hearing before the Subcommittee on National Security, Veterans Affairs, and International Relations Committee on Government Reform U.S. House of Representatives.* 107th cong., 1st sess., November 29, 2001.
[11] Robinson, C. Donald, and Donald Brown. "Development of Metrics to Evaluate Effectiveness of Emergency Response Operations." 10th International Command and Control Research and Technology Symposium. 2005.
[12] Patvivatsiri, Lisa. "Simulation Model for Bioterronrism Preparedness in an Emergency Room." Proceedings of the 2006 Winter Simulation Conference. L. F. Perrone, F.P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds. 2006.
[13] Sullivan, Thomas J. "Modeling and Simulation for Emergency Response." Lawrence Livermore National Laboratory, Report No. UCRL 92001 Preprint. 1985.
[14] National Research Council (NRC). "Making the Nation Safer: The Role of Science and Technology in Countering Terrorism," The National Academies. http://www.nap.edu/openbook.php?isbn=0309084814.
[15] Jain, Sanjay, and Charles R. McLean. "An Integrating Framework for Modeling and Simulation for Incident Management." *Journal of Homeland Security and Emergency Management*, 3, iss. 1, Article 9 (2006). http://www.bepress.com/jhsem/vol3/iss1/9.
[16] Gomaa, Hassan. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*, Boston: Addison-Wesley Professional. 2004.
[17] Raffo, David, M. "Getting the Benefits from Software Process Simulation." Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE'99), June 1999. http://www.sba.pdx.edu/faculty/davidr/draccess/WEB/publications/conference/Seke99.pdf.
[18] Software Product Lines. "Welcome to Software Production Lines." http://www.softwareproductlines.com/.
[19] Raffo, David M., G. Spehar, and Umanatha Nayak. "Generalized Simulation Model: What, Why and How?" Proceedings of the 4th International Workshop on Software Process Simulation Modeling (ProSim'03). May 2003.
[20] Weiss, David M., and Chi Tau Robert Lai. *Software Product Line Engineering: A Family-Based Software Development Process*, Boston: Addison-Wesley Professional, 1999.

[21] Raffo, David M., Umanatha Nayak, and Wayne Wakeland. "Implementing Generalized Process Simulation Model." Proceedings of the 6th International Workshop on Software Process Simulation Modeling (ProSim'05). May 2005.
[22] Choi, K.-S., S.-C. Jung, H.-J. Kim, D.-H. Bae, and D.-H. Lee. "UML-based Modeling and Simulation Method for Mission-Critical Real-Time Embedded System Development." IASTED Conf. on Software Engineering. (2006): 160 -165.
[23] Gamma, Erich, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design patterns: elements of reusable object-oriented software,* Boston: Addison-Wesley Professional, 1994.
[24] Barbarisi, Osvaldo, and Carmen Del Vecchio. "UML Simulation Model for Hybrid Manufacturing Systems." Proceedings of the 13th Mediterranean Conference on Control and Automation. 2005.
[25] Amodie, Jeffrey S. "Simulation modeling for hospital emergency department resource analysis." Florida Atlantic University. 1995.
[26] Shao, Guodong, Charles McLean, and Swee Leong. "A Simulation System with Neutral Data Interface for Shipbuilding Operation." Proceedings of the Industrial Simulation Conference. (October 2005): 137-141.
[27] Jain, Sanjay, and Charles R. McLean. "A Concept Prototype for Integrated Gaming and Simulation for Incident Management." Proceedings of the 2006 Winter Simulation Conference. L. F. Perrone, F.P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds. 2006.
[28] ProModel Corporation. *ProModel Manufacturing Simulation Software Reference Guide,* Orem: ProModel, 1996.
[29] ProModel Corporation. *ProModel Manufacturing Simulation Software User's Guide,* Orem: ProModel, 1996.
[30] Software Engineering Institute. "Software Production Lines." http://www.sei.cmu.edu/productlines.
[31] Brownsword, Lisa, and Paul Clements. "A Case Study in Successful Product Line Development." Software Engineering Institute (SEI) Technical Report, CMU/SEI-96-TR-016. Carnegie Mellon University, Pittsburgh, PA. 1996.
[32] Clements, Paul, and Linda Northrop. "Salion, Inc.: A Software Product Line Case Study." Software Engineering Institute (SEI) Technical Report CMU/SEI-2002-TR-038. Carnegie Mellon University, Pittsburgh, PA. 2002.

Author Biographies

Guodong Shao is a computer scientist in Manufacturing Simulation and Modeling Group, Manufacturing Engineering Laboratory at National Institute of Standards and Technology (NIST). His e-mail address is gshao@cme.nist.gov**.**

Y. Tina Lee is a computer scientist in the Manufacturing Engineering Laboratory at National Institute of Standards and Technology (NIST). She is currently Secretary of the SISO's Core Manufacturing Simulation Data (CMSD) Product Development Group.