Abstract

SEGALL, NOA. Design and Prototyping of a Cognitive Model-Based Decision Support Tool for Anesthesia Provider Management of Crisis Situations. (Under the direction of David B. Kaber.)

This research involved the prototyping of a decision support tool (expert system) for use by anesthetists in crisis situations, in order to promote prompt and accurate patient diagnosis, care, and safety. The tool alerts anesthetists to a developing crisis, manifested by changes in certain patient physiological variables, and provides them with a list of potential causes and preventive measures for dealing with the crisis. The tool provides advice in an unobtrusive manner. Information is presented in a format requiring minimal interaction with the system interface.

Decision support tools for managing patient crisis situations may be useful in large hospitals where an attending anesthesiologist supervises multiple nurse anesthetists or anesthesiology residents that are delivering drugs to patients across operating rooms. Such a tool can provide support to nurses and residents when the attending physician is not present, and can warn of potential crisis situations that would prompt the anesthesia provider to contact an attending physician. The attending physician may also use the tool as a quick method of learning patient status when entering an OR. In addition, the tool could be used by practitioners working alone to deliver anesthesia.

A novel approach was applied to the development of the decision support tool to support anesthesiology decision-making. First, a hierarchical task analysis was conducted to identify the procedures of the anesthetist in detecting, diagnosing, and treating a critical incident,

specifically, myocardial infarction. Second, a cognitive task analysis was carried out to elicit the necessary goals, decisions, and information requirements of anesthetists during crisis management procedures. The results of these analyses were then used as bases for coding a cognitive model using GOMS (goals, operators, methods, selection rules), a high-level cognitive modeling language. EGLEAN (error-extended GOMS language evaluation and analysis tool), an integrated modeling environment, was used as a platform for developing and compiling the GOMS model and applying it to a Java-based simulation of a patient status display. After the anesthetist's decision-making process was captured in GOMS, a basic interface for the decision support tool was prototyped (extending traditional OR displays) to present output from the computational cognitive model by using ecological interface design principles. Finally, a preliminary validation of the tool and interface (patient state and cognitive model output displays) was performed with samples of expert anesthesiologists and human factors professionals in order to assess the usability and applicability of the decision support tool. The anesthesiologists indicated that they would use the decision support tool in crisis situations and would recommend its use by junior anesthesia providers. The human factors experts provided comments on the interface's compliance with usability principles, such as providing prompt feedback and preventing errors.

This research has provided insight into anesthetist decision-making processes in crisis management. It resulted in a prototype of a cognitive model-based decision support tool to augment anesthetist decision-making abilities in these situations.

DESIGN AND PROTOTYPING

OF A COGNITIVE MODEL-BASED DECISION SUPPORT TOOL

FOR ANESTHESIA PROVIDER MANAGEMENT OF CRISIS SITUATIONS

by

NOA SEGALL

A dissertation submitted to the Graduate Faculty of North Carolina State University

In partial fulfillment of the requirements for the Degree of Doctor of Philosophy

INDUSTRIAL ENGINEERING

Raleigh, NC

2006

APPROVED BY:

Dr. Christopher Mayhorn	Dr. Regina Stoll
Dr. Melanie Wright	Dr. Robert St. Amant
Dr. Gary Mirka	Dr. David Kaber Chair of Advisory Committee

Dedication

To my dear husband, Gideon, without whose help, encouragement and love I would not be where I am today.

Biography

Noa Segall was born in Haifa, Israel. She completed her high school education in Haifa in 1993. Following service in the Israel Defense Force as a computer operator, she studied Mechanical Engineering at the Technion – Israel Institute of Technology and graduated in 1999 with a Bachelors degree. She received her Masters degree from Oregon State University's Industrial and Manufacturing Engineering department in 2003, where her research focused on the usability of hand-held devices for exam administration. Since 2003, she has been working towards a Doctoral degree in the department of Industrial and Systems Engineering at North Carolina State University under the guidance of Dr. David Kaber. Her current research interests include human factors in medical systems, human-computer interaction, cognitive modeling, and human factors in automation design.

Acknowledgements

It is difficult to overstate my gratitude to my advisor, Dr. David Kaber. His continuous guidance and support made this work possible. I also wish to acknowledge my committee members for their time and comments on this document.

I would like to thank Dr. Melanie Wright and Dr. Jeffrey Taekman of the Duke University Human Simulation and Patient Safety Center for their insightful input during the decision support tool development process and for providing access to Duke University Hospital personnel and facilities. I would also like to thank Dr. Wright for her help and guidance during the expert interview process and Dr. Taekman for taking part in multiple interviews. I am grateful to the physicians who volunteered to participate in the interviews conducted as part of this research and to evaluate the decision support tool. Finally, I appreciate the continued support of my colleagues at the North Carolina State University Cognitive Ergonomics Lab.

Table of Contents

List of Tables

List of Figures

Acronyms

ABCD	Airway, Breathing, Circulation, Drugs
ACLS	Advanced Cardiac Life Support
ADST	Anesthesia Decision Support Tool
AH	Abstraction Hierarchy
AI	Artificial Intelligence
AIQ	Artificial Intelligence Quotient
ASA	American Society of Anesthesiologists
ATC	Air Traffic Control
BP	Blood Pressure
CAD	Coronary Artery Disease
CDM	Critical Decision Method
CNC	Computer-Numerical Controlled
CO	Cardiac Output
COPD	Chronic Obstructive Pulmonary Disease
CPR	Cardiopulmonary Resuscitation
CTA	Cognitive Task Analysis
CVP	Central Venous Pressure
DST	Decision Support Tool
ECG	Electrocardiogram
EGLEAN	Error-extended GOMS Language Evaluation and Analysis
EID	Ecological Interface Design
$ETCO_2$	End-Tidal CO_2

FMS	Flexible Manufacturing System
HSPSC	Human Simulation and Patient Safety Center
HTA	Hierarchical Task Analysis
GDTA	Goal-Directed Task Analysis
GLEAN	GOMS Language Evaluation and Analysis
GOMS	Goals, Operators, Methods, Selection rules
GOMSL	GOMS Language
GUI	Graphical User Interface
HR	Heart Rate
ICU	Intensive Care Unit
MAC	Monitored Anesthesia Care
MAP	Mean Arterial Pressure
MI	Myocardial Infarction
NMB	Neuromuscular Block
NTG	Nitroglycerin
OR	Operating Room
PA	Pulmonary Artery
PAWP	Pulmonary Artery Wedge Pressure
PVC	Premature Ventricular Contractions
SPO_2	Arterial Blood O_2
TEE	Transesophageal Echocardiography

1. Introduction

1.1 Critical Incidents in Anesthesia

In the operating room (OR), the anesthesia practitioner is responsible for injecting patients with narcotics that prevent them from experiencing physiological stress, muscle relaxants that serve to prevent movement, and amnesic agents that prevent awareness. Anesthetists are also charged with patient well-being in terms of maintaining hemodynamic (blood circulation) stability, ensuring appropriate breathing or ventilation, and generally monitoring the patient's physiological status. The patient's hemodynamic state is monitored by the anesthetist using computer displays that present continuous data on variables such as blood pressure, electrocardiogram (ECG) output, tidal volume (volume of inspired or expired air per breath), and heart rate (Loeb & Fitch, 2002). Auditory data streams, such as heart and breath sounds (using a stethoscope), team members' verbal communication, and sounds and alarms from machines such as the mechanical ventilator, also serve to indicate patient status (Sowb & Loeb, 2002). Additional information can be obtained through observation of urine output, surgeon activities, patient behavior, etc. or by requesting lab tests (Cook & Woods, 1996). The anesthesia practitioner integrates this data to derive abstract physiological concepts about patient state, such as depth of anesthesia or cardiovascular system performance (Jungk, Thull, Hoeft, & Rau, 2000).

The anesthesia provider must constantly monitor the patient, as well as computer displays and the surgical procedure, in order to anticipate or remedy critical incidents. Cooper, Newbower, and Kitz (1984) have defined critical incidents in the OR as human errors or equipment failures "that could have led (if not discovered or corrected in time) or did lead to an undesirable outcome, ranging from increased length of hospital stay to death" (p. 35). Critical incidents may develop into crisis situations within a matter of seconds. When this happens, swift action is necessary to prevent brain damage, permanent injury, or death. Gaba, Fish, and Howard (1994) estimated that at least 20% of anesthesia cases involve some kind of perioperative (during a surgical procedure) problem, and approximately 5% of cases develop into crisis situations. Given that 40 million anesthetics are administered annually in the United States (American Society of Anesthesiologists (ASA), 2005), this translates to approximately two million patients at risk. However, anesthesia cases are also comprised of long periods with few tasks, which call only for vigilant monitoring of patient state (Watt, Maslana, & Mylrea, 1993). For this reason, the work experience of anesthetists has been described as "hours of boredom, moments of terror" (Gaba et al., 1994, p. 1). During those moments of terror, the anesthetist must perform complex, dynamic tasks involving high workload and information loads, such as hypothesizing what the source of the problem may be, testing different assumptions, monitoring changes in patient state, administering drugs, ventilating the patient, communicating with the surgical staff, etc. This combination of task complexity and OR dynamics can be conducive to error-making when managing critical incidents.

With respect to anesthetist errors, several cognitive factors are thought to limit human performance and increase the likelihood of potential crises (Gaba & Howard, 1995):

- Detection of critical incidents requires attention to multiple data streams, but in dynamic situations, it is difficult to concentrate and monitor every data stream frequently enough. For example, anesthetists only spend about one-third of their time looking at monitors (Loeb & Fitch, 2002) and, therefore, do not detect abnormal values reliably, especially during periods of high workload (Loeb, 1993).

- Dynamic attention allocation (or divided attention) is critical during crisis management, although attention is a limited resource. The anesthetist may need to discern rapid changes embedded in complex data streams while attending to multiple routine tasks (such as bagging (manually ventilating) the patient, intubating (inserting a breathing tube), or administering drugs).
- Experience affects the ability to accommodate unexpected events; however, experienced anesthesia practitioners also appear to be vulnerable to attentional fixation errors, like novices (e.g., Sowb & Loeb, 2002).
- Poor resource use or action planning may lead to inadequate responses to emergency situations. For example, some studies have shown that anesthetists could not reliably identify OR alarms by their distinctive sounds, even when deemed important (Loeb, Jones, Leonard, & Behrman, 1992; Sowb & Loeb, 2002).

In an analysis of 616 critical anesthesia incidents, Cooper et al. (1984) found 70% to be caused primarily by human error in drug administration, anesthesia machine use, and airway management. Factors that commonly contributed to these incidents include failure to check (e.g., equipment or patient vital signs), inadequate caregiver experience, inattention, and haste. A study of the 70 cases that resulted in substantial negative outcomes (such as cardiac arrest or death) found that 33% were caused by judgment errors such as drug overdoses, and an additional 19% were due to monitoring- or vigilance-related issues such as detection failures. The authors suggest several strategies for improving incident detection, based on the causes of these severe events. Additional training is the most important of these strategies; improved supervision (or a second opinion), equipment or human factors improvements, and additional monitoring instrumentation were also cited as strong potential approaches.

Additional studies have attributed between 50% and 82% of anesthesia mishaps to human error (Weinger, 1999; Blike, Surgenor, & Whalen, 1999). Growing awareness to the cause of such mishaps has directed many patient safety initiatives to the practice of anesthesiology (Weinger, 1999; Gaba, 2000). Examples include the introduction of new monitoring technologies, use of patient simulators for training, and application of human factors research methods (Weinger, 1999); these efforts have made anesthesia safer than ever. However, there is still a long way to go until anesthesiology reaches the safety levels of other high workload, high risk professions such as aviation (Gaba et al., 1994). Currently, anesthesia-related deaths occur at a rate of one per 250,000 procedures in the United States (ASA, 2005), while aviation-related fatalities occur at a rate of one per 56 million enplanements (National Transportation Safety Board, 2004).

1.2 Specific Anesthesia Crisis

Anesthetists may be called upon to treat a wide variety of crises in the OR. These can generally be grouped into three types of events: those associated with a certain system in the human body, such as metabolic events; those associated with certain populations or surgical procedures, such as obstetric events; and those caused by equipment failures. Table 1 presents examples of anesthesia-related crises, some of which may be caused, or exacerbated, by anesthetist error (from Gaba et al., 1994).

Table 1. Examples of anesthesia-related crises.

Event Type	**Examples**
Generic events	Acute hemorrhage Hypertension Fire in the OR
Cardiovascular events	Sinus bradycardia Myocardial infarction Venous air/gas embolism
Pulmonary events	Airway rupture Bronchospasm Pneumothorax
Metabolic events	Hypoglycemia Hypothermia Malignant hyperthermia
Neurologic events	Central nervous system injury Local anesthetic toxicity Seizure
Equipment events	Ventilator failure Syringe or ampoule swap Leak in the anesthesia breathing circuit
Cardiac anesthesia events	Cardiac laceration Hypotension during cardiopulmonary bypass
Obstetric events	Emergency cesarean section Obstetric hemorrhage
Pediatric events	Aspiration of a foreign body Laryngospasm

In the event of a crisis, anesthesia providers usually rely on precompiled responses to critical intraoperative events primarily learned through experience. Unfortunately, few crisis treatments have been codified and taught methodically. As a result, most anesthetists are not optimally prepared to respond to complex or critical situations (Gaba, 1994).

Although crisis situations appear to begin suddenly and develop rapidly, a crisis usually begins as a triggering event, becoming a problem that will only evolve into a crisis if not attended to.

Triggering events often stem from underlying conditions, such as latent errors (errors that remain under control until they combine with other factors to cause an acute problem, e.g., lack of usability in design of equipment interfaces), predisposing factors (patient diseases and the nature of the surgery), and pathological precursors (performance-shaping factors such as fatigue and environmental factors such as noise). The event itself is usually triggered by the patient's underlying medical pathology, the surgery itself (e.g., compression of organs), anesthetist actions or errors, or equipment failures. For example, a patient's medical pathology, combined with routine surgical actions, may cause hypertension. If the hypertension is not detected in time (e.g., due to poorly designed monitoring equipment or anesthetist fatigue) and treated, it may develop into a crisis such as a cerebral hemorrhage or a heart attack. Problems in anesthesia will inevitably occur, but prompt detection and correction can prevent them from becoming crisis situations (Gaba et al., 1994).

Myocardial infarction (MI), also known as a heart attack, is one of the most feared critical events an anesthetist may face in the OR (Roberts & Tinker, 1996). MI results from an imbalance between oxygen supply and demand to the myocardium (the heart's muscle layer), which is usually caused by occlusion of a coronary artery. If untreated, these conditions will develop into myocardial ischemia (lack of blood supply to the myocardium) and, eventually, MI – tissue death in parts of the myocardium (Chaney & Slogoff, 1999; Stedman, 2000). Over 50,000 people a year sustain a perioperative MI (Chaney & Slogoff, 1999). The incidence of perioperative MI is 0.13% to 0.66% in healthy patients and 4.3% to 15.9% in patients who suffered MI at least six months prior to surgery; in patients who suffered MI more recently, incidence rates may be as high as 54% (Roberts & Tinker, 1996). Mortality rates associated with perioperative MI are 27% to 70%; these rates are higher than the 20% mortality rate for MI not related to surgery (Roberts & Tinker, 1996).

Roberts and Tinker (1996) specify the various risk factors for MI:

- *Age*. Increasing age is related to an elevated cardiac risk.
- *Gender*. Males are more likely to sustain MI than are females.
- *Family history*. Heart disease, especially in first-degree male relatives, increases the likelihood of suffering MI.
- *Personality type*. Patients with a Type A personality (characterized by hostility, aggressiveness, competitive drive, and impatience) are more susceptible to coronary artery disease (CAD) than are patients with a Type B personality (characterized by a relaxed, uncompetitive temperament).
- *Smoking*. Cigarette smoking increases the risk of CAD.
- *Hyperlipidemia*. High levels of fat and cholesterol in the blood also present an increased risk of CAD.
- *Hypertension*. High blood pressure is associated with a greater risk of cardiovascular death.
- *Diabetes mellitus*. Diabetic patients have an increased incidence and early onset of CAD.
- *Obesity and sedentary lifestyle*. The effect of these factors is less well-established, but in combination with other risk factors, they can also present an increased risk of suffering MI.

- *Previous MI*. Patients who have sustained MI prior to surgery are at greater risk of sustaining perioperative MI (see above).
- *Type of surgery*. Some surgical procedures carry a higher risk of suffering MI than others. Examples include cardiac surgery (Chaney & Slogoff, 1999), upper abdominal surgery, and non-cardiac thoracic surgery.
- *ASA rating*. The American Society of Anesthesiologists developed a five-level patient physical status classification, with ratings varying from 1, indicating a healthy patient, to 5, indicating a moribund patient unlikely to survive 24 hours with or without operation. This classification also proved to be a good predictor of cardiac risk.
- *Physician care*. Finally, MI incidence may be dependent upon the anesthesia provider, with different MI incidence rates associated with different providers.

The manifestations of myocardial ischemia and MI include the following (Chaney & Slogoff, 1999; Gaba et al., 1994; Stedman, 2000; Stoelting & Dierdorf, 1993; Veterans Health Administration, 2003):

- Patients who are awake may experience central chest pain radiating into the arms or throat, dyspnea (shortness of breath), nausea, vomiting, or altered levels of consciousness or cognitive function.
- Abnormalities in the ECG waveform, including ST segment (representing the period from the end of ventricular depolarization to the beginning of ventricular repolarization; see Figure 1) depression or elevation, hyperacute or tall, prominent T waves (representing ventricular repolarization; see Figure 1), Q waves (representing the initial phase of ventricular depolarization; see Figure 1), arrhythmias (heartbeat

irregularities: premature ventricular contractions (PVC), ventricular tachycardia (a rapid heart rate preventing the heart from adequately filling with blood) or fibrillation), and conduction abnormalities (AV block, bundle branch block). MI is distinguished from myocardial ischemia by persistence and progression of the ST segment and T wave changes, development of Q waves, and evidence for myocardial cell necrosis (elevated cardiacisoenzymes).

- Abnormalities in the hemodynamic system, including hypotension, hypertension, elevation of ventricular filling pressures, V wave (representing the filling of the right atrium against the closed tricuspid valve during ventricular contraction) on pulmonary artery (PA) wedge tracing, tachycardia, and bradycardia (slow heart rate).
- Regional wall motion abnormalities or new onset mitral regurgitation on transesophageal echocardiography (TEE; ultrasound heart imaging).
- Increase in body temperature.

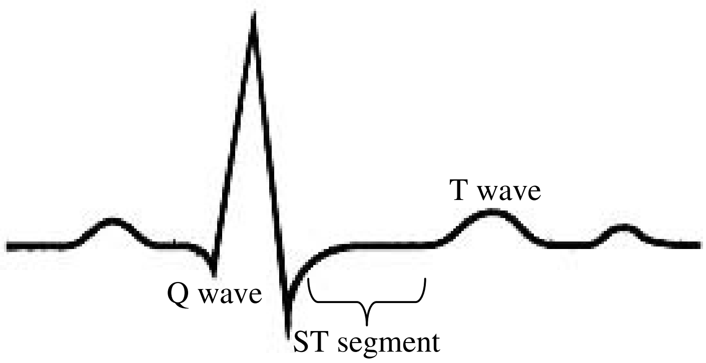


Figure 1. Normal ECG.

Gaba et al. (1994) provide a recommended sequence of detailed steps in treating MI (see Appendix A). If the anesthetist suspects the patient is suffering myocardial ischemia or MI in the OR, the presence of certain clinical signs and symptoms as described above can verify or refute this assumption. If possible, the operation must be terminated and an intensive care unit (ICU) bed requested. Next, ventricular arrhythmias, tachycardia and/or hypertension are treated using drugs, if necessary, and an arterial line to monitor blood pressure is placed. Instructions for treatment of cardiac arrest and hypotension, should they occur, are also provided. MI treatment concludes with secondary management, such as sending blood samples to the clinical laboratory. An additional algorithm for MI treatment is provided by Ludbrook, Webb, Currie, and Watterson (2005), who also describe signs for the detection of MI and a list of precipitating factors (e.g., pre-existing cardiovascular disease).

MI can be further complicated by congestive heart failure, arrhythmias, cardiac arrest, thromboembolic complications, papillary muscle dysfunction or rupture, rupture of the interventricular septum or the ventricular wall, and pericarditis (Gaba et al., 1994; Stoelting & Dierdorf, 1993). Several events are similar to MI: myocardial ischemia, pulmonary embolism, acute dissecting aneurysm of the aorta not involving the coronary arteries, esophageal spasm, costochondritis, acute cholecystitis, acute peptic ulcer/perforation, acute pancreatitis, primary pulmonary pathology, non-ischemic abnormalities of the ST segment or T wave, and ECG artifacts (i.e., if the electrode is improperly placed, or if changes in patient position or surgical manipulation alter the position of the heart relative to the electrodes; Gaba et al, 1994).

1.3 Information Displays for Anesthetist Support

As described on Page 1, anesthetists use many sources of information on the patient's status in order to decide whether a critical incident is developing. In addition to gathering data directly from the environment (e.g., viewing the surgical procedure or listening to the patient), the anesthesia provider monitors several devices to detect and diagnose changes in patient state. Dorsch and Dorsch (1999) list displays that usually make up the anesthesia workstation:

- *Gas monitoring equipment*. Devices such as mass spectrometers are used to display waveforms of inspired and end-tidal oxygen, carbon dioxide (capnometry), volatile anesthetic agents, nitrous oxide, and nitrogen.
- *Airway pressure, volume, and flow measurement devices*. Respirometers measure the volume of and flow of respiratory gases. Monitoring this data can help detect obstructions, disconnections, leaks, ventilator failure, irregular volumes and more in patients breathing spontaneously or through a ventilator. Respirometers can be connected to the breathing system; a dial is often used to display the current volume. Airway pressure monitors are used to warn of abnormal pressure conditions in the breathing system. These monitors, which usually display pressures and alarm limits, may be freestanding or incorporated into a ventilator or anesthesia machine.
- *Pulse oximetry*. This method is used to measure oxygen saturation non-invasively, using a probe that can be attached to different parts of the patient's body, most commonly the fingertip. Pulse oximeters are usually combined with other monitors, and they commonly display percent saturation, pulse rate, and alarm limits.

- *Neuromuscular transmission monitoring equipment*. Neuromuscular block (NMB) is a measure of the degree of patient muscular relaxation. Monitoring NMB involves placing two electrodes along a nerve and passing a current through them. Muscle response can be evaluated visually, tactilely, or through monitoring methods such as accelerography and electromyography.
- *Temperature monitors*. Body temperature must be measured (externally or, more often, internally) continuously during surgery. Most temperature monitors also feature limit alarms. The temperature can be displayed on the probe itself or as a separate display.
- *Blood pressure monitors*. Blood pressure readings from an inflatable cuff can be viewed through a monitor, which usually incorporates alarms for systolic, diastolic, or mean blood pressure, as well as heart rate.
- *Recordkeeping and information management systems*. Anesthesia practitioners are required to maintain an anesthesia record, documenting actions and events that occurred while the patient was in their care. This is usually done manually, but some anesthesia departments make use of automated recordkeeping systems in the OR. Such systems can record patient variables, workstation variables (such as administered drugs), notes by the anesthesia provider, lab results, and other measurements.

Anesthetists also operate and monitor anesthesia machines, gas supply systems, and airway apparatus. All together, as many as 30 physiological variables may be monitored during a surgical procedure (Michels, Gravenstein, & Westenskow, 1997). Watt et al. (1993) provide a list of variables that are commonly monitored using commercial devices (see Table 2).

Table 2. Physiological variables commonly monitored by anesthetists.

Monitored Variable		Description
Electrocardiogram (ECG)		Waveform of heart's electrical activity
Heart rate (HR)		Derived from ECG
ST segment		Specific measure of heart activity, derived from ECG
Arrhythmia		Derived from ECG
Blood analysis		Dissolved gases and electrolytes
Thromboelastogram		Blood clotting factors
Transesophageal Echocardiography (TEE)		Ultrasound heart imaging via esophageal probe
Noninvasive blood pressure		Measured using a cuff and pressure transducer
Invasive blood pressure		Measured using a transducer and catheter
Blood oxygen saturation (SaO_2)		Oxygen carried by hemoglobin
Cardiac output (CO)		Systemic blood flow
End tidal	CO_2 ($ETCO_2$) anesthetic agent nitrogen	Volume percent of exhaled CO_2 Volume percent of exhaled anesthetic agent Volume percent of exhaled nitrogen
Inspired	O_2 anesthetic agent Nitrous oxide (N_2O)	Volume percent of inhaled O_2 Volume percent of inhaled anesthetic agent Volume percent of inhaled N_2O
Respiration rate		Breathing rate
Airway pressure		Pressure in patient ventilation circuit
Tidal volume		Volume of gas delivered to patient on mechanical ventilation at each breath
Minute volume		Volume of gas breathed in one minute
Inspired/expired ratio		Time ratio of inhalation and exhalation
Electroencephalogram		Wave form of brain's electrical activity
Electromyogram		Muscle electrical activity
Evoked potentials		Neurologic response to stimulus
Neuromuscular transmission		Neuromuscular blockade effects
Temperature		Measured internally or externally

Some of the devices used by anesthetists feature single-sensor single-indicator auditory limit alarms, which are activated whenever a physiological variable deviates from a predefined range. Murphy and Vender (2001) maintain that at least five alarms – inspired oxygen, airway pressure, pulse oximetry, blood pressure, and heart rate – should be operational during anesthesia care. However, most practitioners turn off alarms, mostly due to the high frequency of false alarms, but also because they believe they can detect changes without a need for alarms and because it may be difficult to recognize the source of the alarm or what it indicates (Block, Nuutinen, & Ballast, 1999; Seagull & Sanderson, 2004).

There is general agreement that the existing patient monitoring equipment does not provide sufficient support to the anesthetist for prompt and accurate decision-making (e.g., Ahmed, Nevo, & Guez, 1990; de Graaf, van den Eijkel, Vullings, & de Mol, 1997; Krol & Reich, 1999; Mylrea, Orr, & Westenskow, 1993; Weinger, 1999; Zhang et al., 2002). Research aimed at facilitating swift and accurate problem detection through interface design has developed in two main directions. One group of studies has focused on increasing the saliency of deviations of patient physiological variables from normal ranges. For example, since the interpretation of patient states depends on reliable information integration from several data streams, object displays have been developed which depict measured and derived variables (such as depth of anesthesia) as multidimensional graphical objects (Blike et al., 1999; Drews, Wachter, Agutter, Syroid, & Westenskow, 2004; Jungk et al., 2000; Michels et al., 1997; Zhang et al., 2002). Others have applied sonification, the representation of data relations through sound, to physiological variables in an attempt to alert as well as inform anesthetists of patient state (Crawford, Savill, & Sanderson, 2003; Loeb & Fitch, 2002). Crawford et al. (2003), for example, evaluated respiratory sonification, which plays an integrated sound stream depicting the patient's respiration rate, tidal volume and end-tidal CO_2.

In addition to enhancing the anesthetist's monitoring capabilities, human factors techniques have been applied to other tasks in anesthesiology. For example, Syroid et al. (2002)

developed an interface that estimated past, present, and future concentrations and effects of administered intravenous anesthetic drugs, based on pharmacokinetic and pharmacodynamic models. The anesthetist's drug administration task was described, and its requirements generated several iterations of design and usability evaluation. Lin et al. (1998) used human factors design guidelines to redesign the interface of a patient-controlled analgesia pump, based on results of a cognitive task analysis (knowledge elicitation and engineering with actual anesthetists). Zhang, Johnson, Patel, Paige, and Kubose (2003) asked four students who had taken at least one graduate-level human factors or human-computer interaction course to compare the safety of two volumetric infusion pumps using heuristic evaluation. With little training, the evaluators found many usability problems (both in the pumps' physical design and in the behavior of the interface) of varying severity.

Although human factors research methods have been applied in the development and design of the tools described above, typically they only analyze the anesthesia provider's monitoring tasks and result in superficial interface modifications to patient monitoring systems. However, anesthetist actions and problem-solving behaviors are generally concentrated at higher levels of abstraction (e.g., entire physiological systems, rather than single measured variables that only partially map their behavior; Hajdukiewicz, Vicente, Doyle, Milgram, & Burns, 2001). Thus, most of these tools target only the first step to treating critical incidents – recognizing that a problem exists. They do not provide a diagnosis (defining what the problem is), etiology (identifying its causes) or suggestions for treatment.

1.4 Decision Support Tools in Anesthesia

A decision support tool (DST) is a computer-based tool that uses a knowledge base and algorithms to give advice on a particular subject (Sheridan & Thompson, 1994). Many tools have been developed for the medical domain (Rennels & Miller, 1988) to support clinician decision-making in various tasks including chronic pain management (Knab, Wallace, Wagner, Tsoukatos, & Weinger, 2001), antibiotics administration (Evans et al., 1998), laboratory results monitoring, adverse drug event detection, and critiquing orders of blood products (Haug, Gardner, & Evans, 1999). Other uses of such systems include medical education (Lincoln, 1999) and consumer health informatics (i.e., patient decision support; Jimison & Sher, 1999).

Rennels and Miller (1988) discuss the problems faced by developers of artificial intelligence (AI) systems in medicine, and particularly in anesthesiology. The domain is complex and unstructured: to diagnose a medical condition, the clinician must integrate knowledge on links between diseases, their symptoms and causal mechanisms, patient history, clinical literature, and social issues relevant to the disease and its treatment. Some of this knowledge, for example the mechanisms underlying a disease, may not be available or fully understood. For this reason, most systems that have been developed for the anesthesia environment are prototypes (proof-of-principle systems) that are not in clinical use. These systems face several challenges before they can be implemented in the OR, including being able to deal with real-time data and artifacts, dealing with the complexity of medicine (e.g., accounting for concurrent treatment and co-existing disease), and accommodating varied practice approaches.

The problem-solving mechanism behind DSTs is an AI method such as rule-based and probability-based systems, neural networks, fuzzy logic, and genetic algorithms (Krol & Reich, 1998; Spooner, 1999). Unlike other AI techniques, the neural network and genetic algorithm approaches compile a knowledge base by processing example problems (Huang & Endsley, 1997; Spooner, 1999). In this sense, systems based on neural networks or genetic algorithms are easier to create (Krol & Reich, 1998), since existing databases can be used and expert knowledge elicitation processes may not be required. Although these techniques have been applied to some extent in anesthesiology decision support (e.g., Beatty, Pohlmann, & Dimarki, 2000; Linkens & Vefghi, 1997; Mylrea et al., 1993), they have several disadvantages which have limited their applicability. Both neural networks and genetic algorithms require considerable computing power (Spooner, 1999). The main challenge in genetic algorithms is determining criteria by which fitness is defined (i.e., which will provide the best solution; Spooner, 1999). In neural networks, the method by which knowledge outputs are created from raw data is hidden from the user. Thus, a neural network is similar to a "black box" in that its logic is not transparent and explicitly understandable (Huang & Endsley, 1997; Lowe, Harrison, & Jones, 1999; Spooner, 1999). However, to gain user acceptance, a DST should be able to explain the rationale behind its decisions (Huang & Endsley, 1997; Krol & Reich, 1998; Lowe et al., 1999; Sheridan & Thompson, 1994), a function that neural networks cannot fulfill. Such explanations serve both to make the system more intelligible to users and to uncover shortcomings in its knowledge transformation process (Davis & Lenat, 1982). Finally, neural networks often require large amounts of training data (Lowe et al., 1999) and their performance may be unpredictable when presented with rare problems for which they were not specifically trained (Krol & Reich, 1998; Lowe et al., 1999).

The shortcomings of AI techniques that do not utilize an explicit knowledge base make the case for expert systems, which use skilled operators' knowledge to build the knowledge base upon which the DST relies. In particular, rule-based systems have several advantages over other AI methods (Rennels & Miller, 1988):

- The rules used to populate the knowledge base can be easily translated to English using a rule translation program, thus satisfying the need for the system to explain its decisions.
- Use of the rules makes it straightforward for experts to inspect and understand the system's logic. This enables the expert to identify errors in the knowledge base and suggest changes.
- Knowledge can be added incrementally to the system, enhancing its performance.

AI techniques and rule-based systems have been widely used in anesthesiology for drug administration (Mahfouf, Abbod, & Linkens, 2002; Krol & Reich, 1998; Hunt, Haynes, Hanna, & Smith, 1998), fault diagnosis in anesthesia circuits (Uckun, 1994), pre-operative anesthesia planning, mechanical ventilation monitoring, management of congestive heart failure, and more (Rennels & Miller, 1988). Another class of anesthetist DSTs is designed to detect specific conditions in patients (Krol & Reich, 1998). Most common among these are the intelligent alarms (or integrated monitoring), which have been suggested as a solution to the abundance of false alarms in the OR. An intelligent alarm system monitors multiple patient physiological variables in real-time and synthesizes them to produce a status assessment and to warn of possible problems when deviations from a normal status are detected (Mylrea et al., 1993). One such example is described by Becker, Käsmacher, Rau, Kalff, and Zimmermann (1994), who employed a fuzzy inference approach to the design of an intelligent alarm for cardiac anesthesia. Rules for estimation of five state (derived) variables such as depth of anesthesia, based on heart rate and other physiological variables, were constructed by considering expert opinions. An interactive display was used to show deviations of the state variables from normal ranges. More detailed information about each variable could also be accessed by the anesthesia practitioner, if necessary. This system was installed in an OR and evaluated by anesthesia providers during surgical procedures (Becker et al., 1997). Its sensitivity, specificity, and predictability were found to be high.

Some intelligent alarms go beyond diagnosis of abnormal events, suggesting therapeutic actions to correct them. For example, Schecke et al. (1988) developed AES-2 for a specific stage in a surgical procedure (aortocoronary bypass surgery after termination of the extracorporeal circulation). AES-2 is an extension of an advanced anesthesia information system that records patient variables and manual data inputs (e.g., drug administration). To implement the intelligent alarm, anesthetist knowledge was used to create fuzzy rules that determine whether derived variables such as depth of anesthesia deviate from normal ranges. When such a deviation is detected, AES-2 alerts the anesthesia provider and recommends therapeutic action – which symptom to treat first, what drugs to administer (based on side effects) and in what dosage (based on patient data and results of previous dosages). Initial evaluations of this tool have been carried out, but no results were reported.

Two groups describe plans to create similar expert systems. Krol & Reich (1998) put forward a rule-based expert system that would integrate intraoperative physiological data, patient history, and drug effects in real-time to detect critical incidents, rank their etiologies by likelihood, and suggest possible treatments. Ahmed et al. (1990) propose an expert system that would show the anesthesia provider a single index, the Vital Function Status, indicating the patient's real-time level of danger based on deviations from normal ranges of vital signs. Once such a change is detected, the system presents the deviant variable(s), a list of possible diagnoses ranked by urgency, and a list of matching therapeutic actions. The proposed system is adaptive in that its knowledge base is updated based on anesthetist actions and results. Development and validation of these two systems have not been reported.

In summary, none of the research on DSTs for anesthesia administration has involved the use of structured human factors methods to construct knowledge bases or design system interfaces. The majority of systems are not in clinical operation (Rennels & Miller, 1988; Uckun, 1994). Most of the rule-based systems developed for clinical diagnosis support have been designed for narrow application fields, due to the complexity of maintaining systems that include more than a few thousand rules (Miller & Geissbuhler, 1999).

2. Problem Statement and Objective

Crisis management skills are important in several work domains. In aviation, for example, air crews are trained in problem-solving in crisis situations, recognizing that human performance is the critical resource in managing unfamiliar events (Cook & Woods, 1994). In anesthesiology, however, crisis management is not adequately taught; this skill is also not easily learned during clinical practice (Gaba et al., 1994).

Cognitive aids such as checklists and guides are an additional method to aid operators in dealing with complex, dynamic situations (as well as routine events), helping them overcome the tendency to forget facts and skip steps in procedures during crises. In anesthesiology, there has been a historical emphasis on relying on memory to handle both routine and crisis situations (Veterans Health Administration, 2003). As a result, there are few cognitive aids for this domain. Gaba et al.'s (1994) book is one such cognitive aid. It lists various types of anesthesia crises in the OR and describes the procedures for managing them. The Veterans Health Administration (2003) provides a more condensed version of this book in the form of laminated cards for treating a number of more common crises. Both of these sources discuss the treatment of MI (see Appendix A). However, these cognitive aids are intended for use in preparing to recognize and manage crises, during debriefing after a crisis, and for training purposes (Gaba et al., 1994). Usually, they are not referred to during a crisis, unless additional help is available or there is no improvement in the patient's situation after initial treatment (Veterans Health Administration, 2003).

As discussed in Sections 1.3 and 1.4, many research efforts have been focused on helping anesthetists to detect and diagnose critical incidents. Tools to enhance quick and accurate detection of abnormal events are an important first step in crisis management – displays alerting the anesthetist to the existence of a problem. However, these tools only target the task of monitoring the patient for deviations of physiological variables from a predefined range. The anesthetist is still charged with integrating the different sources of information to select between several possible diagnoses, and then decide on a treatment plan.

DSTs have been developed to automate the information integration step and suggest a diagnosis; some also suggest therapeutic actions. Yet the tools described in the body of literature do not, in general, take into account human factors design approaches or principles of interface design. Although experience in aviation and nuclear power plants has shown human factors design techniques to reduce errors (Lin et al., 1998), most of the studies reviewed here do not describe the user interface at all, and those that do make no reference to the application of any structured design principles, such as ecological interface design or usability principles. DST development in this domain rarely uses structured knowledge elicitation techniques or cognitive task analysis as a basis for supporting anesthetist decision-making in real-time crisis management. For this reason, existing prototype DSTs may not provide cognitively plausible explanations as to how their diagnosis was derived.

Since healthcare is an open system, events which the DST does not anticipate are bound to occur (Vicente, 2003). When a tool suggests an uncertain course of action, operators tend to simply accept its imperfect advice, even when the necessary information to make a decision is available (Vicente, 2003). Therefore, it is important that the DST explain its underlying logic so that the anesthetist can evaluate its suggestion before accepting (or rejecting) it. Such a justification will also promote user acceptance of any tool (Huang & Endsley, 1997).

Few studies have systematically examined anesthetist cognitive decision-making processes during crisis management. There is a need to analyze and model anesthetist behavior in crisis detection, diagnosis and treatment. Such a cognitive model could be used as a basis for developing a DST that would provide guidance in diagnosing and treating a crisis, while explaining its suggestions. Such information should be delivered through a cognitively compatible interface to enhance its usability and promote success in resolving crises like MI.

The present study was a methodological investigation that prototyped an anesthesia DST with the capability to accurately recommend actions in a crisis situation with explanatory power, and developed an ecologically-based interface design for delivering decision information. The methodology section outlines this approach and details specific techniques to achieving the research goals. A validation step was carried out to assess the potential applicability of the tool to the OR and the usability of the interface prototype from an expert anesthesiologist perspective.

3. Methods

A novel approach to the development of a decision support tool for decision-making in anesthesiology (ADST) was applied in this research. The steps to the approach included:

1. Performing a hierarchical task analysis of anesthesiologist steps and procedures in managing a crisis situation, specifically myocardial ischemia and MI (myocardial infarction), to identify critical OR environment cues and resources used, as well as the general timing of events. The Gaba et al. (1994) book and Ludbrook et al. (2005) paper were useful references in this step.
2. Carrying out a cognitive task analysis (e.g. Endsley, 1993) to capture the knowledge structure of the anesthetist in detecting, diagnosing, and treating the critical incident (MI).
3. Using information from the hierarchical and cognitive task analyses as a basis for coding a cognitive model in GOMSL (goals, operators, methods, selection rules language), a high-level cognitive modeling language that describes the knowledge a user must have in order to perform tasks on a certain system (Kieras, 1999).
4. Prototyping an interface for presenting output from the computational cognitive model using ecological interface design, a framework for the design of interfaces that is particularly useful for supporting operators during unanticipated events (Vicente & Rasmussen, 1992).
5. Simulating ADST operation using a GOMSL model compiler, EGLEAN (error-extended GOMS language evaluation and analysis tool; Wood, 2000). EGLEAN allows for integrated modeling and execution of GOMSL models with Java-based representations of interface devices (Soar Technology, 2005). EGLEAN was used as

a platform for developing and compiling the GOMSL model and applying it to a simulation of a patient status display for generating decision support tool output.

The steps of this overall method are described in detail in the following sections.

3.1 Hierarchical Task Analysis

A hierarchical task analysis (HTA) is an analytic strategy for developing system or procedural solutions to specific task performance problems (Annett, 2003). In effect, it is a method for analyzing complex tasks in order to better understand the procedures, cues, and information required to accomplish the task. A HTA can be used to design new interfaces or modify existing ones, to compare the complexity of different system designs, and to develop training manuals. The methodology facilitates specification of interfaces that support identified task sequences. It has been applied to a wide range of problems, from printer cartridge replacement to surgery and air traffic control tasks (Annett, 2003).

The process of HTA starts with data acquisition. Information about the task can be gathered using various sources, such as behavior observation, process documents (e.g., standard operating procedures), interviews, and simulations (Annett, 2003). The task is then described as a hierarchy of tasks and sub-tasks using goals, tasks, operations, and plans.

- A *goal* is the desired state of the system.
- A *task* is the method by which the goal may be achieved, where the method depends on user and system characteristics and constraints.

- An *operation* is a unit behavior, specified by a goal to be achieved, the circumstances under which it will be activated (input), the actual activity (action), and the conditions that indicate the goal has been attained (feedback; Annett, 2003).
- A *plan* is a rule or list of rules that specify the order in which operations should be carried out (Annett, 2003).

This information can be represented in either tabular or diagrammatic form, where a hierarchical diagram is more useful for clearly displaying the functional structure of the task (Annett, 2003).

Figure 2 presents an example high-level HTA for the goal of carrying out supermarket checkout operations (Shepherd, 2001). Tasks include setting the till to start a new shift, dealing with customer purchases, etc. An example operation is entering a product price manually. There are plans for deciding which tasks and operations to perform, e.g. if there is a spillage on the conveyor (plan 0), it should be cleaned. This specific analysis identified training needs and indicates where special training might be needed (Shepherd, 2001).

Although the HTA can generate useful outcomes, such as this, for redesigning a task or supporting technology, several general limitations of the methodology have been identified. The HTA may be difficult to learn and apply correctly (Stanton & Young, 1998); it is also considered to be more time-intensive than other human factors research tools such as questionnaires or keystroke level models of user behavior (Stanton & Stevenage, 1998). Beyond this, the HTA does not address many cognitive aspects of performance, such as identification of low-level goals active in working memory during tasks, or identification of critical decisions and information requirements necessary to achieve those goals.

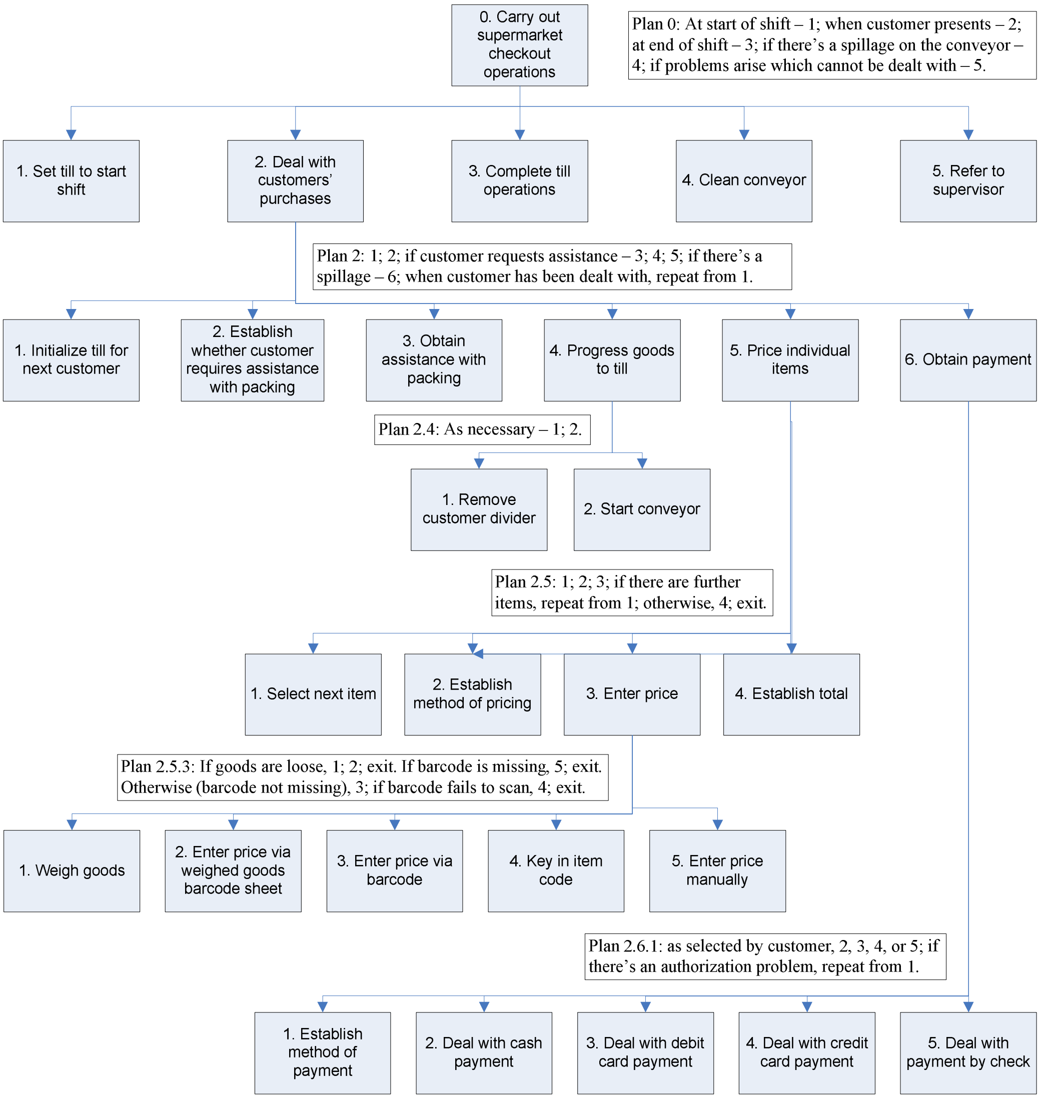


Figure 2. HTA for supermarket checkout task.

A HTA of the task of detecting, diagnosing, and treating myocardial ischemia and MI was developed in this study to identify the sequence of necessary steps, environmental cues, uses of existing technology, etc. In the data acquisition phase, process documents such as Gaba et al.'s (1994) book, the Veterans Health Administration (2003) laminated cards and Ludbrook et al.'s (2005) MI detection and treatment algorithm were reviewed for an overview of the procedures necessary to treat MI. Direct observation, interviews, and simulations were carried out at the Human Simulation and Patient Safety Center (HSPSC) in Duke University's Medical Center as additional sources of information for the analysis.

At the HSPSC, patient simulators in the form of full-sized mannequins (see Figure 3) are used for training and research (Duke University HSPSC, 2005). The mannequins simulate the functioning of the human body (e.g., their heart rate can be measured using a stethoscope) and they respond to stimuli from the environment (e.g., exposure to light causes a reduction in pupil diameter) as well as physical and pharmacologic interventions (i.e., their vital signs will change in response to drug administration). Realistic scenarios such as the occurrence of a perioperative MI can be programmed on the simulators to train students (from Duke University's School of Medicine, School of Nursing, and Department of Anesthesiology) in crisis management. After the students diagnose and treat the critical incident, an expert anesthesiologist discusses the management of the case with the students, to identify any errors or alternate treatments.

Figure 3. Training of anesthesiology residents using the patient simulator.
(Courtesy Duke University Medical Center)

The ability of the HSPSC to artificially simulate critical incidents in order to teach crisis management skills without jeopardizing human lives made it an ideal setting for the data acquisition phase of the HTA. Information about how to detect the onset of MI (e.g. what physiological variables change, which alarms go off, etc.), the consequences of correct and incorrect diagnoses, and the different possible treatments and complications was gathered using pen and paper while observing students manage the crisis and during their follow-up discussions with the expert anesthesiologist. Video recordings of training sessions collected by the HSPSC were also available to support this analysis. The end-product of the observations was essentially a detailed "activity list", a description of the correct sequence of events used to perform the task (Diaper, 1993).

In addition, five semi-structured interviews with three experienced anesthesiologists (1-2.5 hours each) were conducted at Duke University Hospital. The interviews revolved around MI diagnosis and treatment as personally witnessed and treated by the anesthesiologists or as taught to anesthesiology residents (see Appendix B for a list of questions that was used to guide the interviews). Experts were also presented with MI treatment algorithms (Gaba et al., 1994; Ludbrook et al., 2005) and asked to adapt them to their own treatment plans and to provide criteria for quantifying patient states. This step as part of the HTA was utilized to identify plans or strategies the expert anesthetist may use, as well as the task environment and system states that trigger the use of specific strategies.

Tasks identified in the HTA corresponded to methods in the GOMSL cognitive model (see below); HTA operations corresponded to operators in GOMSL; and plans in the HTA corresponded to decisions in GOMSL. The outcomes of this application of the HTA are presented in the Results and Discussion section. They served as a basis for the following cognitive task analysis.

3.2 Cognitive Task Analysis

Cognitive task analysis (CTA) is analysis of the knowledge, thought processes, and goal structures of cognitive tasks (Hollnagel, 2003). Its objective is to identify and describe dynamic goal sets, factual knowledge stores, mental strategies, critical decisions, and situation awareness requirements for performing a particular cognitive task. These structures can be used to design new system interfaces or evaluate existing interfaces; to develop expert systems; for operator selection (based on a defined skill set); and for training purposes (Wei

& Salvendy, 2004). Tasks that stand to benefit most from this type of analysis are generally unstructured and difficult to learn, occur in real-time, complex, dynamic and uncertain environments, and they may involve multitasking (Gordon & Gill, 1997). For this reason, anesthesiology-related tasks are good candidates for the application of CTA methods. CTA has been applied to a wide variety of tasks, including decision support system design (Wei & Salvendy, 2004) and anesthesiology, specifically ventilation management (Sowb & Loeb, 2002), extubation (breathing tube removal) decision-making (Weinger & Slagle, 2002), and preparation for surgery (Xiao, Milgram, & Doyle, 1997).

There are two major phases to CTA. The first involves the analyst becoming conversant in the domain of interest (Hoffman, Shadbolt, Burton, & Klein, 1995). The HTA is useful for this purpose, for example. Observation of expert behavior is also considered part of the initial phase of CTA, when the domain needs to be defined and described (Wei & Salvendy, 2004). Behavior observation may be effective for identifying the tasks involved in a domain as well as information needs and constraints on the tasks (environmental, temporal, resource), discovering basic problem solving strategies that are not consciously accessible, and studying motor skills and automatic procedures (Wei & Salvendy, 2004). Once the analyst develops a thorough understanding of the target domain, they may use structured approaches to behavioral and communications analysis, as well as interrogative methods in one-on-one interaction with operators. This latter step is intended to identify operator goal states, critical decisions, situation awareness requirements, and methods to situation assessment. The analysis may also yield information on the consistency of operator outcomes for one goal state relative to the situation awareness requirements of dependent goals.

Many techniques are available for carrying out a CTA; different methods are appropriate for achieving different objectives (Wei & Salvendy, 2004). Two CTA methods will be discussed here: the critical decision method and goal-directed task analysis. The critical decision method (CDM) is suited for supporting decision-centered design for high time pressure, high information content, dynamic environments (Hutton, Miller, & Thordsen, 2003; Klein, Calderwood, & MacGregor, 1989). CDM is comprised of a series of interviews that are organized around a specific incident, which an expert has experienced. During the first interview, the expert is asked to recall the episode in its entirety. The interviewer then goes over the incident several times with the expert, using probes designed to capture particular aspects of the incident. The probes emphasize perceptual aspects of the event (what was seen, heard, considered, and remembered) rather than rationalizing about decisions that were made at the time. For a particular decision, information may be solicited about factors such as presence or absence of cues and their nature, assessment of the situation and how it might evolve, or goals and options that were considered. CDM has been found to elicit rich information from experts, since this information is specific, reflects the decision maker's approach, and is grounded in actual events (Hutton et al., 2003). It has been used in the construction of a database for an expert system and to identify training requirements for the domain of computer programming (Klein et al., 1989).

Klein et al. (1989) also describe use of the CDM method to create AIQ (artificial intelligence quotient), a method for evaluating expert systems. The AIQ method consists of three steps. First, CDM is used to specify bases for expert performance in the domain of interest. Next, these bases are compared to the expert system, and mismatches (aspects of expertise not covered by the expert system or new expert system capabilities that did not previously exist) are recorded. Finally, expert system performance is compared and contrasted with an expert operator's performance along four scales: system performance, system and operator performance (when working together), interface adequacy, and system impact on the organization. AIQ has been used to compare two existing expert systems that automate air load planning systems, as well as to evaluate expert systems at different stages of development. In this way, the CDM, as a form of CTA, can support the evaluation and further development of expert systems.

In the domain of anesthesiology, Weinger and Slagle (2002) interviewed expert clinicians about the decision whether or not to extubate a patient at the end of a general anesthetic procedure. They asked them to describe a specific notable or difficult extubation decision they had made and then probed them about primary and contributing factors that influenced their decision. Questions about hypothetical situations were used to widen the scope of the interview beyond the specific base case described. Sentences from the interviews were analyzed for concepts and links to other concepts; they were then graphically depicted as concept maps, which were combined into a single map. The concept map provided insight into the four factors that most influence the decision whether to extubate a patient post-anesthesia, such as the patient's current ability to ventilate and the expected ability to mask ventilate or reintubate the patient should extubation fail. Psychosocial issues, including surgeon preferences, were also found to influence this decision. Further CTA interviews with less experienced clinicians were used to determine how knowledge structures, factor prioritization, etc. differ with experience.

Since the CDM approach requires that the expert being interviewed has actual experience in the incident he or she will describe, it can often be difficult to find suitable interviewees for analyzing specific complex tasks or critical situations. As noted, MI occurs on relatively rare occasions, therefore it is unlikely that the majority of the population of anesthesiologists may have personally experienced this event. Consequently, the number of potential interviewees for application of the CDM may be very limited. With this in mind, another CTA method, specifically goal-directed task analysis (GDTA), was explored in this research to investigate the MI treatment task.

GDTA is an information requirements assessment methodology developed by Endsley (1993) for the aviation domain. Anesthesiology, like piloting, is a complex task involving critical decision-making and time pressure, making GDTA an appropriate method for analyzing anesthetist cognitive processes in treating MI. The goal of GDTA is to identify information processing or situation awareness requirements of system users; its outcome is a list of critical decisions and information requirements that can be used as a basis for display design, training program development, development of situation awareness assessment measures, and operator selection.

The general steps to conducting a GDTA include (Usher & Kaber, 2000):

- *Identifying the users' major goals*. In the present study, the major goal is MI treatment.
- *Identifying subgoals to support the overall goal*. High-level subgoals in addressing MI include verifying the manifestations of myocardial ischemia, informing the operating surgeon, etc. These can be further broken down, e.g. verification of myocardial ischemia manifestations includes such subgoals as assessing clinical signs and symptoms. (This information is also revealed through the HTA).
- *Identifying operational tasks to achieve the subgoals*. For example, one of the tasks that should be performed in order to achieve the subgoal of assessing clinical signs and symptoms is to evaluate hemodynamic status. (This information is also revealed through the HTA.)
- *Creating questions to address decision-making in task performance*. Some questions the anesthetist may ask to evaluate hemodynamic status include: Are there unexpected hemodynamic changes? What are the potential causes of hemodynamic changes? (The HTA methodology does not identify critical decisions to operator goal states.)
- *Developing information requirements to answer these questions*. The information necessary to decide whether unexpected hemodynamic changes are occurring include patient heart rate, blood pressure and oxygen saturation, previously administered drugs, and more. (The HTA method identifies information available to the operator through the environment and existing system interfaces. It does not reveal operator information needs for decision-making.)

GDTA elicits task subgoals, key decisions, and information needs from a domain expert using interviews. The expert is typically presented with a task scenario and asked to mentally place themselves in the situation. The analyst then creates a goal tree (or list) describing this information, independent of any technology that may ordinarily be used to achieve tasks or answer operational questions (e.g., a patient's heart rate is shown on standard waveform displays; the use of such displays is not mentioned in the analysis). The analysis is based upon operator goal states in the scenario and not on specific states of the task environment. This is a major difference between the HTA and GDTA. The analysis also does not require that goals be addressed in a specific order. There are two general limitations to GDTA. First, the tool focuses on operator information needs, not on how they should be acquired. Second, GDTAs do not address temporal variations in information requirements (Endsley, 1993). Though some elements may be more important at certain times during task performance and less important at other times, this factor is not addressed in the task representation.

GDTA has been successfully applied to various domains. Endsley and Rodgers (1994) employed the GDTA approach in air traffic control (ATC). The authors utilized existing task analyses, videotapes of simulated ATC tasks, and interviews with air traffic controllers to gather data about this task. An overarching goal of maintaining flight safety was found to depend on the achievement of subgoals such as avoiding conflicts between aircraft. Tasks were assigned to each subgoal, e.g. one of the tasks to be performed in order to avoid conflicts is to ensure aircraft separation. To ensure separation, an air traffic controller must be able to answer questions such as whether the vertical separation of two aircraft meets or exceeds federal limits. The information necessary to answer this question includes the altitude of both aircraft, the altitude rate of change, etc. This analysis was used to develop situation awareness information requirements for air traffic controllers, providing a foundation for future developments of ATC systems.

Usher and Kaber (2000) applied GDTA to control of flexible manufacturing systems (FMSs). A FMS typically consists of a number of CNC (computer-numerical controlled) machines, a material handling system, and robots that are controlled by a supervisory computer. Thus, the operator's overall goal in a FMS is to achieve planned output of products. This goal is achieved by accomplishing such subgoals as avoiding bottlenecks and maintaining normal system functions. The subgoal of avoiding bottlenecks, as an example, can be broken down into objectives (e.g., resolve capacity bottlenecks) that have sub-objectives (e.g., suspend jobs with high stack) which are associated with tasks (e.g., identify jobs ahead of schedule). For the task of identifying jobs that are ahead of schedule, the operator should be able to answer what jobs have a scheduled completion time that is less than their due date. The information necessary to answer this question includes, for example, the slack times (due date – completion time) of all jobs. Usher and Kaber (2000) used their GDTA to develop design guidelines for display content in FMSs. For example, one of the guidelines was to present a list of job order numbers, due dates, slack times, processing times, etc. in order to aid the operator in performing the task of identifying jobs that are ahead of schedule.

Figure 4 presents an example of a GDTA goal tree constructed for the subgoal of choosing an anesthetic technique by nurse anesthetists (Wright, 2004). (This goal is not part of the MI crisis studied in this work, since the patient will normally already be anesthetized when a critical incident occurs. The purpose of presenting this GDTA is merely to illustrate the use of the analytical tool in the target research context.) The overarching goal for the GDTA is to provide safe, effective anesthesia care. One of the subgoals for this goal is to plan anesthesia care, and one of its subgoals is to choose an anesthetic technique. Tasks for achieving this subgoal include analyzing patient history, understanding the surgical procedure, and evaluating existing resources. For each task, a list of questions/decisions the nurse anesthetist needs to address is provided, as well as a list of information requirements. These information requirements were used to develop queries to evaluate nurse anesthetists' situation awareness and could also be used in the design of information displays (Wright, 2004).

In the present study, three expert anesthesiologists were interviewed (1-2 hours each) in order to gather information for the myocardial ischemia and MI treatment GDTA. A partially-completed goal tree including the various goals, subgoals, tasks, decisions, and information requirements was prepared based on the process documents, observations and HTA. The goal tree was presented to the anesthesiologists and they were asked what modifications they would make to it, e.g. what decisions they would add or delete for a certain task. This approach was used to develop a complete GDTA for MI crisis management. Goals identified in the GDTA corresponded to methods in the GOMSL cognitive model (see below) and decisions and information requirements in the GDTA were used to code decisions in GOMSL. With the HTA output, the GDTA results supported the following cognitive modeling work. The outcomes of the analysis are presented in the Results and Discussion section.

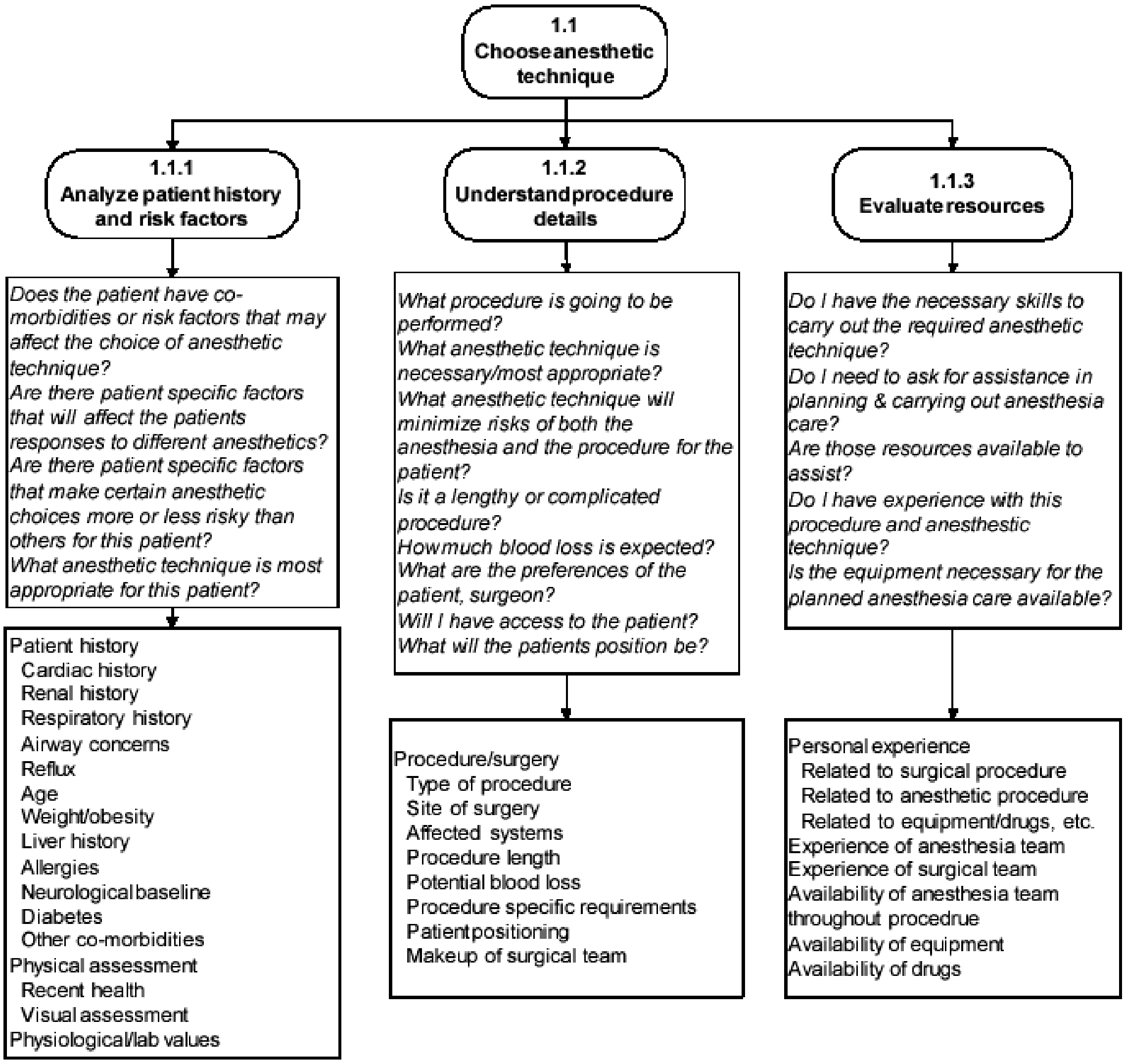


Figure 4. GDTA for goal of choosing anesthetic technique.

3.3 GOMS

Once information about the expert's decision-making processes has been obtained through the CTA step, many methods can be used for cognitive modeling purposes (Gordon & Gill, 1997; Wei & Salvendy, 2004). GOMS (goals, operators, methods, selection rules) is one such formal cognitive modeling tool. The goal of cognitive modeling is to predict how users will interact with a proposed system design (Olson & Olson, 1990) or process. The GOMS model, first proposed by Card, Moran, and Newell (1983), achieves this goal by describing the procedural knowledge that a user needs to have in order to carry out tasks on a certain system or as part of a process, using certain interfaces (Kieras, 1997; Kieras, 1999). Card et al.'s (1983) Model Human Processor, which quantifies human information processing in terms of basic perceptual, cognitive, and motor abilities (Kieras, 1997; Olson & Olson, 1990), is then utilized to predict how long it would take an experienced user to complete the task, based on execution times of plan retrieval from long-term memory, method selection (as a function of task features), working memory access, and motor movement execution (Olson & Olson, 1990). The ability of GOMS models to predict human performance has been found to expedite and reduce costs of user testing during the initial phases of interface design, since the models can serve as surrogates to empirical user data in the comparison and evaluation of different designs (John & Kieras, 1996b; Kieras, 1999). GOMS has been successfully used to model human interaction with many real-world applications, from a television on-screen menu interface to a command and control database system for space operations (John & Kieras, 1996b).

GOMS models can be viewed as programs that the user learns and then executes (Kieras, 1997), and in fact some GOMS variants are structured as parameterized computer programs (John & Kieras, 1996a). GOMS models contain the following information-processing components and data structures (Card et al., 1983; Kieras, 1999):

- *Goals*. A goal is the state of affairs to be achieved. Its dynamic role is to provide a memory point to which the system can return on failure and from which information can be obtained (e.g., about what has already been tried).

- *Operators*. Operators are perceptual, motor, or cognitive actions that the user executes. Depending on the level of abstraction established by the analyst, operators can be primitive or high-level. Lower-level operators reflect basic psychological mechanisms, while high-level operators describe specifics of the task environment.
- *Methods*. A method is a list of steps necessary to accomplish a goal. In a GOMS model, it is a conditional sequence of goals and operators. High-level operators are replaced with methods containing lower-level operators as task analysis increases in depth.
- *Selection rules*. Rules route control to the appropriate method using if-then statements.

GOMS models can be created at different levels of detail. A high-level GOMS model represents tasks and processes, while lower-level analyses will generally include primitive, keystroke-level operators. In a high-level model, goals and operators do not refer to interface-specific aspects of the task. In this case, the lowest level of detail an operator may have is to perform a mental function (think-of, decide) or invoke a system function (e.g., "update database" versus the lower-level "click on UPDATE button"). Methods in a high-level model document what information the user needs to have, where errors may be detected by the user, and how they may be corrected (Kieras, 1997). The purpose of such a high-level analysis is to drive the choice of functionality early in the system design process. By considering tasks at a high level, decisions about which functions the system should ultimately include can be made prior to actual interface design. The analyst can elaborate a high-level GOMS model after making interface-specific design decisions by writing the corresponding lower-level methods, working down to a keystroke level if necessary (Kieras, 1997). The final level of detail is determined by the analyst's needs, environmental constraints, and user experience, where lower-level models are necessary for less experienced users.

There are five variants of GOMS in use today: CMN-GOMS (Card et al., 1983), KLM (keystroke-level model), GOMSL (GOMS language; Kieras, 1999), NGOMSL (natural GOMS language; Kieras, 1996), and CPM-GOMS (cognitive, perceptual and motor operators, or critical path method; John & Kieras, 1996b). Only GOMSL will be discussed here, since it is accommodated by EGLEAN (see Section 3.5 below). GOMSL is based on a simple serial stage human information processing architecture (John & Kieras, 1996a) and as such has auditory, visual, vocal, manual, and cognitive processors, each with its own working memory, as well as shared long-term memory (Kieras, 1999). GOMSL has a structured notation in which methods take on the form of a program and contain both external keystroke-level operators (in low-level models) and internal operators (that can, for example, add or remove content from working memory). There are several outputs from a GOMSL model of a task. By associating execution times (or distributions of execution times) with each operator, the model can predict the total time to carry out the task (Card et al., 1983). Time to learn how to perform the task can be predicted from the length of the methods (Kieras, 1999). Task complexity can also be estimated, from the length and number of methods included in the model. GOMSL has been empirically validated for keystroke-level models (John & Kieras, 1996a). It is useful for applications in which user methods are hierarchical and sequential (Kieras, 1999). This makes GOMSL particularly suitable for modeling anesthetist tasks, which are event-driven (Gaba et al., 1994) and relatively sequential.

Figure 5 presents an example of GOMSL code, modeling the task of copying text in a text editor using menu commands (Kieras, 1999). This is an example that Card et al. (1983) began with some years ago and has been used throughout the GOMS literature for demonstrating variations on the modeling techniques. The hierarchical structure of the code is evident from the higher-level methods (e.g., Copy Text) that call out lower-level methods (e.g., Paste Selection) which, in turn, call out the lowest-level methods (e.g., Select Insertion_point) that contain only primitive operators (e.g., Look_for). Methods are performed step by step. Accomplish_goal statements are used to call out lower-level methods. After they are completed, a Return_with_goal_accomplished statement is used to return control to the higher-level method and the next step is carried out. Selection rules (Select Text) can be used to select between different methods depending on environmental constraints – here, the length of the text to be copied. Card et al. (1983) and Kieras (1999) used CMN-GOMS, NGOMSL and GOMSL to make comparison of different interaction methods for the text copying task and to identify the interaction method representing the lowest level of cognitive complexity.

Figure 5. Portion of GOMSL code for text-editing task.

```
Method_for_goal: Copy Text
      Step 1. Accomplish_goal: Copy Selection.
      Step 2. Accomplish_goal: Paste Selection.
      Step 3. Verify "correct text moved".
      Step 4. Return_with_goal_accomplished.

Method_for_goal: Copy Selection
      Step 1. Accomplish_goal: Select Text.
      Step 2. Accomplish_goal: Issue Command using Copy.
      Step 3. Return_with_goal_accomplished.

Method_for_goal: Paste Selection
      Step 1. Accomplish_goal: Select Insertion_point.
      Step 2. Accomplish_goal: Issue Command using Paste.
      Step 3. Return_with_goal_accomplished.

Selection_rules_for_goal: Select Text
      If Text_size of <current_task> is Word,
        Then Accomplish_goal: Select Word.
      If Text_size of <current_task> is Arbitrary,
        Then Accomplish_goal: Select Arbitrary_text.
      Return_with_goal_accomplished.

Method_for_goal: Select Word
      Step 1. Look_for_object_whose
                Content is Text_selection_start of <current_task>
                and_store_under <target>.
      Step 2. Point_to <target>; Delete <target>.
      Step 3. Double_click mouse_button.
      Step 4. Verify "correct text is selected".
      Step 5. Return_with_goal_accomplished.

Method_for_goal: Select Arbitrary_text
      Step 1. Look_for_object_whose
                Content is Text_selection_start of <current_task>
                and_store_under <target>.
      Step 2. Point_to <target>.
      Step 3. Hold_down mouse_button.
      Step 4. Look_for_object_whose
                Content is Text_selection_end of <current_task>
                and_store_under <target>.
      Step 5. Point_to <target>; Delete <target>.
      Step 6. Release mouse_button.
      Step 7. Verify "correct text is selected".
      Step 8. Return_with_goal_accomplished.
```

```
Method_for_goal: Select Insertion_point
      Step 1. Look_for_object_whose
                Content is Text_insertion_point of <current_task>
                and_store_under <target>.
      Step 2. Point_to <target>; Delete <target>.
      Step 3. Click mouse_button.
      Step 4. Verify "insertion cursor is at correct place".
      Step 5. Return_with_goal_accomplished.

Method_for_goal: Issue Command using <command_name>
      Step 1. Recall_LTM_item_whose
                Name is <command_name>
                and_store_under <command>.
      Step 2. Look_for_object_whose
                Label is Containing_Menu of <command>
                and_store_under <target>.
      Step 3. Point_to <target>.
      Step 4. Hold_down mouse_button.
      Step 5. Verify "correct menu appears".
      Step 6. Look_for_object_whose
                Label is Menu_Item_Label of <command>
                and_store_under <target>.
      Step 7. Point_to <target>.
      Step 8. Verify "correct menu command is highlighted".
      Step 9. Release mouse_button.
      Step 10.Delete <command>; Delete <target>;
                Return_with_goal_accomplished.
```

Figure 5 (continued).

Although GOMS models like this can be very useful for interface evaluation and assessment of interactive task complexity, several limitations of GOMS and GOMSL have been noted in the literature. GOMS only represents expert, error-free performance. Thus it is not applicable for modeling novice behavior which may involve problem-solving rather than expert plan retrieval and execution; nor can the modeling method account for errors, which even skilled users may make. GOMSL was developed for describing serial behavior, while many tasks involve processes that occur in parallel. GOMS also focuses on elementary perceptual and motor components of behavior, with a more limited set of operators for representing complex cognitive operations. Finally, GOMS does not address various issues such as mental workload and operator fatigue (Olson & Olson, 1990).

For the purpose of this research, a cognitive model of anesthetist behavior in myocardial ischemia and MI crisis management was coded in GOMSL. The model made use of the task analyses described above by implementing goals corresponding to the goals and subgoals in the GDTA; methods corresponding to tasks in the HTA; operators corresponding to operations in the HTA; and selection rules and decisions corresponding to plans in the HTA and, more directly, decisions and situation awareness requirements in the GDTA. The outcomes of the GOMSL modeling work are presented in the Results and Discussion section. The cognitive model was ultimately compiled and executed using EGLEAN (see below).

3.4 Ecological Interface Design

Ecological interface design (EID) is a theoretical framework for the design of interfaces for complex human-machine systems (Vicente & Rasmussen, 1992). It originated in the work of Rasmussen and Vicente (1989) and Vicente and Rasmussen (1992) who sought to create an interface design methodology that would support skilled users in coping with unanticipated events. EID draws on two theoretical concepts: the abstraction hierarchy (AH; Rasmussen, 1985) is used to represent constraints on the work domain, and Rasmussen's (1983) skills, rules, knowledge taxonomy provides a context for communicating these constraints to the user. Together, these concepts are used to guide system analysis and interface design using three general principles (Vicente, 2002; Vicente & Rasmussen, 1992):

- To support skill-based behavior (automated behavior), users should be able to directly manipulate the interface.
- To support rule-based behavior (cue-action associations not involving cognitive processing), the interface should provide a one-to-one mapping between work domain constraints and perceptual information. Object displays that integrate several directly measurable variables into a single, more meaningful (i.e., goal-relevant) variable are an example of the application of this principle to interface design (see Section 1.3 for a discussion of object displays in anesthesiology).
- To support knowledge-based behavior (analytical problem-solving), the work domain should be represented in the form of an AH that would serve as an external mental model (see Table 3 for an example – an AH of the anesthesiology work domain).

In general, the interface design should encourage use of the lower levels of cognitive control (skill- and rule-based behavior) since they involve fast, effortless processing that is less error-prone, while supporting knowledge-based behavior that is crucial for novice users and for managing unexpected problems (Vicente & Rasmussen, 1992). Displays designed for the anesthesiology domain can promote these principles by facilitating swift and accurate problem detection and decision-making. We claim that a DST that explains its recommendations can be viewed as an extension to EID concepts, since it may reduce reliance on knowledge-based behavior by interpreting patient status based on multiple data streams, making a diagnosis and deciding on a treatment procedure. However, automating such cognitively complex tasks is a difficult challenge that has historically not been addressed in anesthesiology information displays.

Burns and Hajdukiewicz (2004) describe the steps involved in creating an ecological interface. First, an AH of the work domain is constructed. An AH describes the domain along multiple levels of abstraction (usually five levels) that are connected by a means-end relationship. In anesthesiology, the work domain would be the human body, and the five levels of the hierarchy (from most to least abstract) can be selected as follows (Hajdukiewicz et al., 2001):

- *Purposes* – Physiological purposes governing the interaction between the patient and the medical environment. Examples: homeostasis (maintenance of physiological equilibrium), oxygenation, and circulation.
- *Balances* – Prioritized resource allocation to physiological processes. Examples: oxygen supply/demand, electrolytes, and conservation relationships.
- *Processes* – Coordinated physiological processes. Examples: oxygenation, circulation, diffusion, and osmosis.
- *Physiology* – Physiological functions that maintain the processes. This is the level at which the anesthetist can affect physiological state (e.g. by drug administration). Example: functioning of organs.
- *Anatomy* – Anatomical structures. Example: the location, appearance, form, and material of organs.

Next, a part-whole hierarchy is developed for the work domain. This hierarchy is a decomposition of the work domain into systems, subsystems, etc. In medical practice and medical informatics, the human body is often broken down as follows (Hajdukiewicz et al., 2001):

- *Body* – Structurally and functionally linked organ systems.
- *System* – A group of organs that perform related functions.
- *Organ* – Tissue organized to perform a specific function.
- *Tissue* – Cells sharing a common structure and function. There are four types of primary tissue: muscle, nervous, epithelial, and connective.
- *Cell* – Smallest unit capable of performing processes associated with life.

A complete work domain model is a matrix containing the part-whole decomposition on one axis and the functional (AH) decomposition on the other axis. Table 3 (Hajdukiewicz et al., 2001) is a work domain model of the human body. For a patient cardiovascular system at the system level, for example, purposes include adequate circulation and blood volume; balances include mass inflow, storage and outflow; and processes include circulation, volume, fluid supply and sink. The information necessary to produce this model was elicited from medical sources such as physiology textbooks. This method of analyzing the work domain can be used to extract information requirements, constraining relationships, multivariate relationships, and means-end relationships that can be used as a basis for designing the interface.

Table 3. Part of a work domain model of the human body.

	Body	System	Organ	Tissue	Cell
Purposes	Homeostasis	Adequate circulation, blood volume, oxygenation, ventilation	Adequate organ perfusion, blood flow	Adequate tissue oxygenation and perfusion	Adequate cellular oxygenation and perfusion
Balances (including water, salt, electrolytes, pH, O_2, CO_2)	Balances of mass and energy inflow, storage, and outflow	System balances of mass and energy inflow, storage, outflow, and transfer	Organ balances of mass and energy inflow, storage, outflow, and transfer	Tissue balances of mass and energy inflow, storage, outflow, and transfer	Cellular balances of mass and energy inflow, storage, outflow, and transfer
Processes	Total volume of body fluid, temperature, supply: O_2, fluids, nutrients, sink: CO_2, fluids, waste	Circulation, oxygenation, ventilation, circulating volume	Perfusion pressure, organ blood flow, vascular resistance	Tissue oxygenation, respiration, metabolism	Cell metabolism, chemical reactions, binding, inflow, outflow
Physiology		System function	Organ function	Tissue function	Cellular function
Anatomy			Organ anatomy	Tissue anatomy	Cellular anatomy

Using the AH, the model is converted into sets of variables that describe how each level may be quantified. For example, blood pressure would be a variable associated with the processes level, O_2/CO_2 balance would be associated with the balances level, etc. (Hajdukiewicz et al., 2001). Hajdukiewicz et al. (2001) discuss four types of mapping between these variables and operating room sensors:

- *One-to-one mapping* – one sensor measures one patient variable (e.g. pulse measurements provide information about heart rate).

- *Convergent (redundant) mapping* – many sensors can be used to measure one variable (e.g. heart rate can also be determined from the ECG waveform or arterial blood pressure waveforms).
- *Divergent mapping* – a sensor may measure several variables (e.g. ECG waveforms provide information about heart rate, heart rhythm, myocardial oxygenation, and more).
- *No mapping* – a sensor may measure a variable that is not part of the patient work domain (e.g. oxygen tank pressure).

Not all variables can be physically measured. Sharp and Helmicki (1998) categorize such variables for the process of tissue oxygenation in newborns: an analytical model exists for calculating the value of analytically derived variables (such as balance in the alveolar PO_2); heuristically mapped variables can be subjectively quantified (e.g. the adequacy of ventilation can be assessed using arterial PCO_2 that estimates alveolar PCO_2); finally, some variables cannot be obtained with today's medical knowledge (e.g., the ATP level in each cell of the body).

Once a comprehensive list of variables associated with the AH levels is prepared, it can be used to extract different types of constraints that will guide the interface design process. Single variable constraints are usually desired upper and lower bounds on the variable. These are different from patient to patient and are, thus, difficult to determine in the medical domain (Sharp & Helmicki, 1998). In designing the interface, information on single variable constraints can be used to display ranges of scales, determine alarm limits, define visual coding schemes, etc. Multivariate constraints are relationships, such as equations, between two or more variables. Displaying these constraints in a way that is well-understood by users can enhance performance and reduce mental workload. In anesthesiology, the integration of several patient variables to one meaningful variable, such as depth of anesthesia, constitutes a multivariate constraint. Finally, means-end relationships describe the implication of one variable in the value of another across abstraction levels. These relationships should be explicitly displayed on the interface, even if they are not characterized by equations, since they help the user achieve system goals and diagnose problems. The relationships can be presented through display organization, by grouping related graphics and determining required salience levels.

The EID approach has been used successfully in diverse application domains including process control, aviation, software engineering, and military command and control (Vicente, 2002). In most domains, EID has been found to uncover information requirements that were not captured by the existing systems. When empirical evaluations were conducted, ecological interfaces were also shown to improve user performance over existing system interfaces (Vicente, 2002). Several graphical displays have also been developed for the anesthesia workplace based on the principles of EID. Jungk, Thull, Hoeft, & Rau (1999) compared anesthetist performance on three display types for hemodynamic monitoring – a standard trend display, a profilogram display (Becker et al., 1997), and an ecological display presenting four integrated variables and relationships between measured variables. They found that the ecological display promoted successful task completion and strategic decision-making, but at the cost of slower performance and more control actions. Another study by Jungk et al. (2000) evaluated a more comprehensive ecological monitor that displayed 35 measurable and derived physiological variables and featured fuzzy logic-based intelligent alarms. Here, the ecological display proved superior to a conventional trend display in terms of performance. Effken, Kim, & Shaw (1997) created three ecological displays for the ICU environment: a strip-chart display that shows blood pressure in different parts of the body, an integrated display that portrays the physical relationships between these pressures, and an etiological potentials display that relates etiological factors to symptoms and to the target patient state. Subjects achieved better performance using the etiological potentials display, which emphasizes the hierarchical structure of the hemodynamic system, compared with the two lower-level displays.

Most relevant to the present research, Hajdukiewicz et al. (2001) created a work domain model of a patient (as presented in Table 3) and used it to analyze problem-solving in the OR. Specifically, the actions and verbalizations of an anesthesiologist handling a crisis on a patient simulator were mapped onto the different cells of the work domain matrix. It was found that the problem-solving route was cyclical, moving between the higher and lower levels of abstraction and aggregation, corresponding to the anesthesiologist verifying information and monitoring the effect of interventions (see Figure 6). As the crisis developed, the problem-solving trajectory expanded to include more levels of abstraction; as the patient condition became clear, it contracted again. Since most existing displays only capture variables from the lower levels of abstraction (physiology and processes) and the organ level of aggregation, many trajectory nodes were concentrated in this area of the work domain matrix. However, anesthesia providers need information that is generally found at the higher levels of abstraction and at a broader range of aggregation levels (Hajdukiewicz et al., 2001). This problem-solving strategy should be taken into account when designing an interface for the anesthesiology domain. In this research, Hajdukiewicz et al.'s (2001) work domain model was used as a basis for developing a simple prototype of an EID interface (see Figure 12 below), which presents output from the GOMSL cognitive model for supporting anesthetist decision-making in managing a MI crisis.

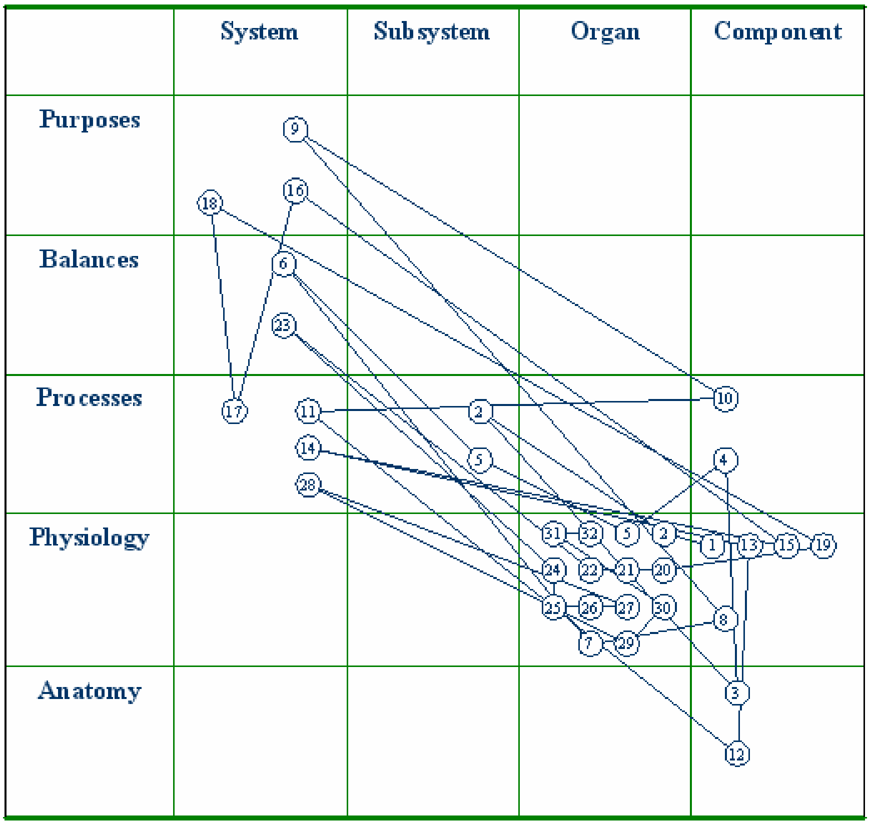


Figure 6. Mapping of anesthesiologist problem-solving to patient work domain model.

3.5 EGLEAN

In this study, EGLEAN (error-extended GOMS language evaluation and analysis; Wood, 2000) was used as a platform for developing and compiling the GOMSL model and applying it to the ADST interface. EGLEAN is an integrated modeling environment developed by Soar Technology, Inc. for simulating GOMSL model interaction with Java-based interfaces (Soar Technology, 2005). The human anesthetist, as modeled in GOMSL, can “see” patient variables displayed in a Java version of the ecological interface and “react” to changes in their values. These reactions are output as ADST advice.

EGLEAN is based on GLEAN (GOMS language evaluation and analysis), a tool developed by Kieras (1999) for compiling and running GOMSL models of human performance. GLEAN can also be used to conduct both run-time analyses (task execution time and working memory load) and static analyses (e.g. method execution profiles) on GOMSL models (Kieras, Wood, Abotel, & Hornof, 1995). EGLEAN provides a GUI for developing GOMSL models, an improvement over the GLEAN command-line interface. Additional EGLEAN features include syntax highlighting, code completion, static error checking, an outline view, interface integration, an advanced run time debugging environment, and access to run time threads, variables and buffers (Soar Technology, 2005). Since EGLEAN was developed based on the GLEAN framework, it compiles GOMSL files with all the psychological constraints and rules used in GLEAN, supporting the cognitive plausibility of models. In GLEAN, the GOMSL-modeled user interacts with a user interface programmed in C++ and populated with time-dependent data using scenario (script) files. However, this requires developing an accurate, text-based model of the abstract behavior of the interface, which is difficult to visualize. In contrast, EGLEAN makes use of a Java graphical user interface (GUI) with which the modeled user interacts. If a Java interface prototype already exists, this can significantly reduce modeling efforts.

EGLEAN is a plug-in to the Eclipse integrated development environment, an open-source platform for Java programming. It makes use of three files to simulate human-computer interaction:

- *.java file*. This is the Java GUI that the modeled user communicates with. The interface includes visual objects the modeled user can see (e.g. text labels) and/or interact with (e.g. buttons). Any underlying functionality (e.g. interface behavior when a button is pressed) is also coded in this file.
- *.gomsl file*. This is the GOMSL model of human behavior in interacting with the interface (see Section 3.3 above). The model receives as input visual objects in the interface and outputs interactions with these objects (e.g. pressing a button).
- *.txt file*. This is a scenario file which can be used to update the Java interface in real time. For example, if the interface includes moving targets, the scenario file could include rows of target coordinates which would serve as input to the .java file.

The relationships between these files within Eclipse are graphically depicted in Figure 7. In this way, the GOMSL model of anesthetist behavior in MI crisis management was applied to the ADST interface. The interface was populated with data from scenario files, including physiological variables for a simulated patient suffering, for example, MI.

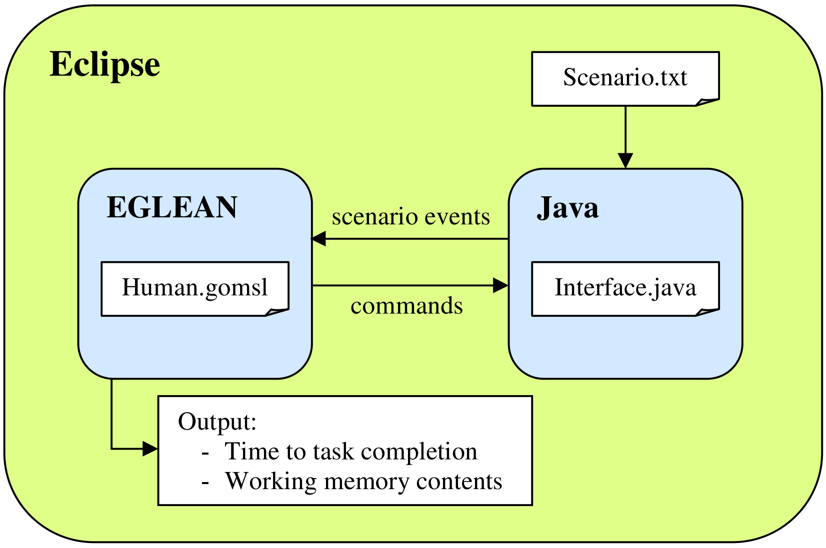


Figure 7. EGLEAN architecture.

3.6 Summary

Figure 8 presents a flow diagram outlining the overall approach taken to the design and development of the ADST to support anesthetists in managing critical incidents. The HTA and GDTA were used to create a GOMS cognitive model. The GOMS model was used to drive the rule-based ADST. Two scenario files were developed to describe two patient state scenarios for evaluation of the cognitive model. An interface for presenting the tool was developed using EID principles. The tool's usability was evaluated using heuristic evaluation and its usefulness was evaluated using an applicability assessment (see next section).

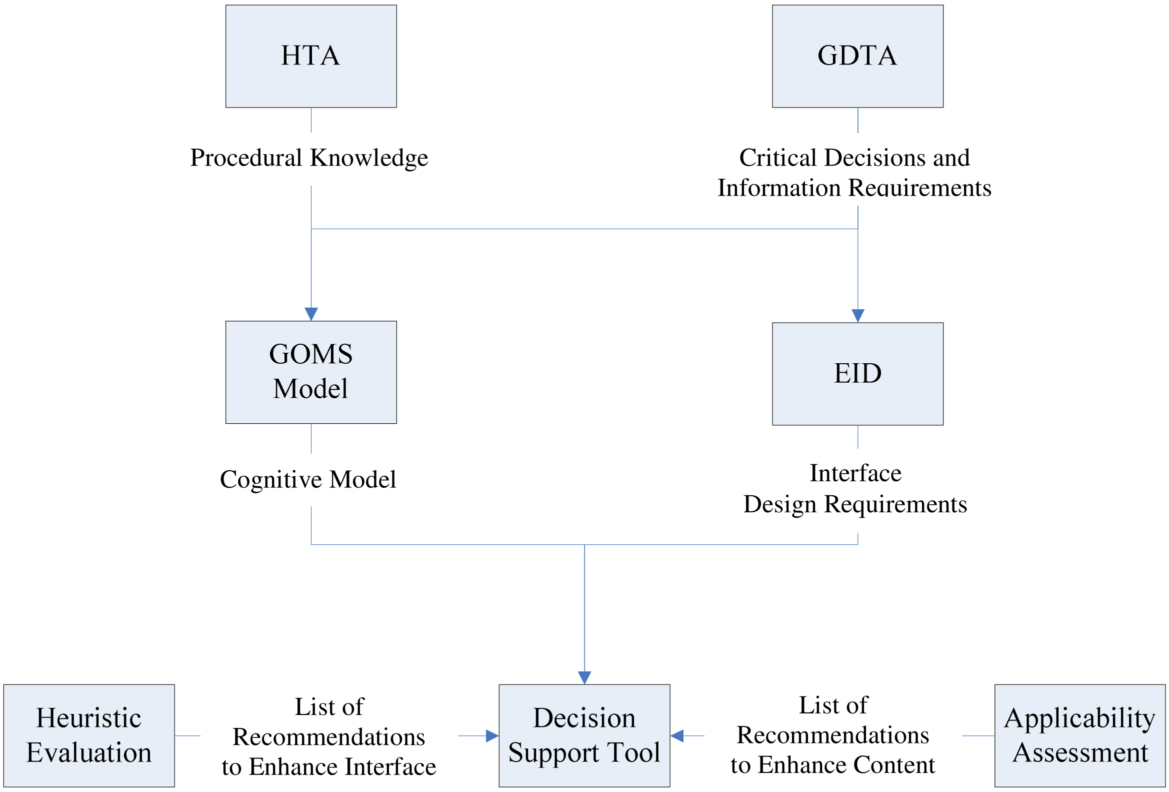


Figure 8. Flow diagram of overall approach to design and development of ADST.

This methodology was designed to provide insight into anesthetist decision-making processes in crisis management. It resulted in a prototype of a cognitive model-based ADST to augment anesthetist decision-making abilities, specifically in treating MI cases, as described in Section 5 – Results and Discussion.

4. Preliminary Validation

The typical approach to validating a cognitive model for describing human behavior in various contexts is to compare action predictions of the model with observations on actual user performance collected during use of the interactive system under investigation. However, this approach is not applicable to validation of the ADST prototyped through the present research, due to the rarity of occurrence of actual cases of MI and virtually no opportunity to collect data on anesthetist performance with the prototype tool in an actual OR environment under crisis conditions. Furthermore, the GOMSL cognitive model is intended to represent expert anesthetist performance in crisis management; consequently, any data that could be collected on intern training at the HSPSC in MI treatment may not represent an appropriate standard by which to evaluate the predictions of the ADST. One of the assumptions of GOMS models is that they represent expert and error-free performance (Card et al., 1983).

In human factors, new approaches to system development and design are often evaluated objectively by having potential users carry out various tasks with and without the proposed system, and comparing their performance along multiple metrics such as number of subtasks completed, time to complete the task, number of errors made, and time spent on correcting errors (Wixon & Wilson, 1997). However, user testing is more appropriate for the later stages of system design (Virzi, 1997); in addition, testing may not be feasible in certain situations, e.g. when resources are limited or when participants representing the user population are rare (such as expert anesthesiologists). Therefore, this validation step involved subjective evaluations of the ADST and interface design through an applicability assessment and a usability inspection.

4.1 Applicability Assessment

The goal of the applicability assessment was to determine the usefulness of the ADST to anesthetists in managing perioperative crises like MI. The assessment was carried out by having three anesthesiologists watch the ADST perform during two hypothetical scenarios, hypertension and MI. The ultimate purpose of the ADST is to run in real-time during surgical procedures in the OR, receiving real-time physiological variable data as input. However, the prototype ADST is not capable of these advanced actions, since it is not directly connected to actual OR sensors. Therefore, two scenario files including values for patient variables were produced, using the HSPSC simulator, to drive the cognitive model simulation for evaluation purposes. In the first evaluation scenario, the simulated patient hemorrhages extensively and develops hypotension, i.e. blood pressure decreases. In the second scenario, the patient suffers MI. The cognitive model diagnoses these problems as they develop and is intended to guide the actual anesthetist through the necessary treatment steps. In the hypothetical scenarios, these conditions do not resolve themselves, i.e. the patient's condition deteriorates continuously, in order to let the treatment protocol play itself out.

Prior to the ADST evaluation, anesthesiologists were requested to sign informed consent forms and complete a short questionnaire. They were presented with a brief user manual describing ADST features and functions and a patient information sheet (provided by the HSPSC) which described the patient's physiological state and the surgical procedure for both scenarios (see Appendix C). A survey was then given to the anesthesiologists, asking them about the tool's performance and perceived usefulness (see survey of applicability in Appendix C). The analyst and anesthesiologists viewed the output display for the ADST on a laptop computer. The anesthesiologists provided perceived ratings of agreement or disagreement with each statement on the applicability survey, such as "Alternative diagnoses were possible that were not suggested by the tool", while the model was running or after it stopped. Scenario-specific statements (items 1- 7) were rated twice, once for each scenario. The form also allowed anesthesiologists to provide comments regarding any of the statements. At the close of each scenario, they were allowed to ask questions about the tool, interface, or scenarios. The analyst recorded the questions and any observations volunteered.

Two response measures resulted from administration of the applicability survey: ratings for the different statements and a summary of comments provided for each statement. These are reported in Section 5.2.1 below. The outcome from this assessment is a concise list of recommendations for improving the content of the ADST prototype.

It was generally expected that utilization of the human factors methodologies, including use of the task analyses and GOMS modeling in the design of the ADST, would lead to a positive evaluation of the tool in terms of applicability. In particular, it was expected that anesthesiologists would find the tool to be useful and would indicate that they would use it in managing real perioperative crises (see survey of applicability in Appendix C).

4.2 Heuristic Evaluation

Usability inspection is an informal means by which to assess interface usability. Inspection methods involve evaluators examining a system interface in early stages of an iterative design process (Virzi, 1997), as compared to end-user testing, which is more suitable for identifying problems in a finished product. A usability inspection technique called heuristic evaluation (Nielsen, 1993) was used to evaluate the ADST interface. Usually, the inspection is done by systematically examining the interface and evaluating its compliance with a set of usability principles, or heuristics. The result is a list of usability problems, each linked to one or more heuristics. Although this method does not directly recommend solutions to the problems identified, it is relatively straightforward to revise an interface design based on any heuristic violations identified (Nielsen, 1993).

Heuristic evaluation has been used to evaluate various applications, such as virtual environment user interfaces (Sutcliffe & Gault, 2004), online documentation (Kantner, Shroyer, & Rosenbaum, 2002), and a voice mail application (Virzi, Sorce, & Herbert, 1993). Kantner and Rosenbaum (1997) describe heuristic evaluation of web sites used for retrieving documents from databases and for looking up industrial product information. They were able to find various usability problems with the help of at least two specialists – usability experts and "double experts" experienced in both usability and the domain of interest. They recommend combining heuristic evaluation with user testing to identify a more comprehensive set of usability problems.

Fu, Salvendy, and Turley (2002) categorized usability problems according to Rasmussen's (1983) skills, rules, knowledge taxonomy (see Section 3.4). For example, they associated consistency problems in the interface with skill-based behavior and learnability with knowledge-based behavior. They evaluated an interface for a web-based training software program through heuristic evaluation and user testing. Six usability experts took part in the heuristic evaluation and six end-users participated in user testing of the software. More usability problems were found through the heuristic evaluation than through user testing. Furthermore, heuristic evaluation was better at identifying problems associated with skill- and rule-based performance and user testing found more problems associated with knowledge-based behavior, in support of Kantner and Rosenbaum's (1997) recommendation to conduct both types of usability assessment.

In general, it is recommended that at least three to five evaluators examine an interface for usability problems (Nielsen, 1993). A smaller number of evaluators will find a smaller number of problems, while a larger number will be less cost-effective. Previous research has found that as few as five evaluators can find up to 75% of known usability problems (Nielsen, 1993). Each evaluator inspects the interface alone several times and notes heuristic violations and comments. The results from all evaluators are then aggregated for a comprehensive list of problems. The evaluators do not need to be usability experts (Virzi, 1997). For example, Zhang et al. (2003) used students with little human factors background to conduct a heuristic evaluation of infusion pumps (see Section 1.3). However, usability experts will find more problems than non-experts, and usability experts who are also familiar with the domain for which the interface was developed will find more problems than those who are not (Nielsen, 1993). In the current study, two usability experts and three domain experts (experienced anesthesiologists) evaluated the ADST interface.

Each evaluator was given the list of heuristics presented in Appendix C. Since the anesthesiologists had no prior usability evaluation experience, they were given oral instructions as to how heuristic evaluation is carried out and what is required of them. Each evaluator watched the ADST prototype step through the two evaluation scenarios on a laptop computer in order to evaluate its compliance with each heuristic. Evaluators then prepared a list of the heuristics that were violated and detailed descriptions of each problem they identified. When all evaluations were complete, the analyst combined them into a list summarizing the heuristics that were violated and the specific problems noted.

Response measures included: (1) the number of unique problems found by the evaluators; and (2) a list of problems that were identified by evaluators. The problems were categorized according to the heuristic that was violated, e.g. problems associated with insufficient feedback. The number of evaluators that identified each problem was also recorded. The outcome from this analysis is a list of recommendations, ranked by severity (i.e., number of evaluators who found each problem), that can be used as a basis for enhancing the ADST interface design. It was expected that use of EID principles to guide interface design would lead to a positive evaluation of the tool in terms of usability.

5. Results and Discussion

5.1 Anesthesia Decision Support Tool

Two task analyses, a HTA and a GDTA, were used to inform a cognitive model describing anesthetist behavior in perioperative myocardial ischemia and MI treatment. This model was coded in GOMSL and interacts with a Java ecological interface: patient variables presented in the interface (imported from scenario files) can be “seen” by the model and resultant actions are output from the model back to the interface, as recommended treatment steps. The rationale behind these recommendations is also presented.

5.1.1 Hierarchical Task Analysis

Figure 9 presents the high-level HTA diagram for the MI treatment task. The overall goal is to treat MI. Tasks for achieving this goal include verifying the manifestations of myocardial ischemia, considering precipitating factors, etc. Operations are unit behaviors such as “Evaluate hemodynamic status” and “Obtain a 12-lead ECG”. Finally, high-level plans (recorded to the right and below the diagram) are used to specify task strategies when certain conditions apply. (Note: Only high-level HTA plans are shown here.)

The complete HTA is presented in Appendix D. It includes 11 high-level tasks, 28 second-level tasks, 45 third-level tasks, and 48 fourth-level tasks. Of these, 103 are operations, i.e. unit behaviors or tasks, which cannot be further broken down. The HTA also includes 18 plans. This analytical tool, as well as the cognitive task analysis results (described below), served as a basis for development of the GOMS model, since HTA is closely related to GOMS (Kieras, 1997). Specifically, the HTA supported the description of task methods as part of the cognitive model (or necessary long-term memory structures) an anesthetist must have for dealing with a MI crisis.

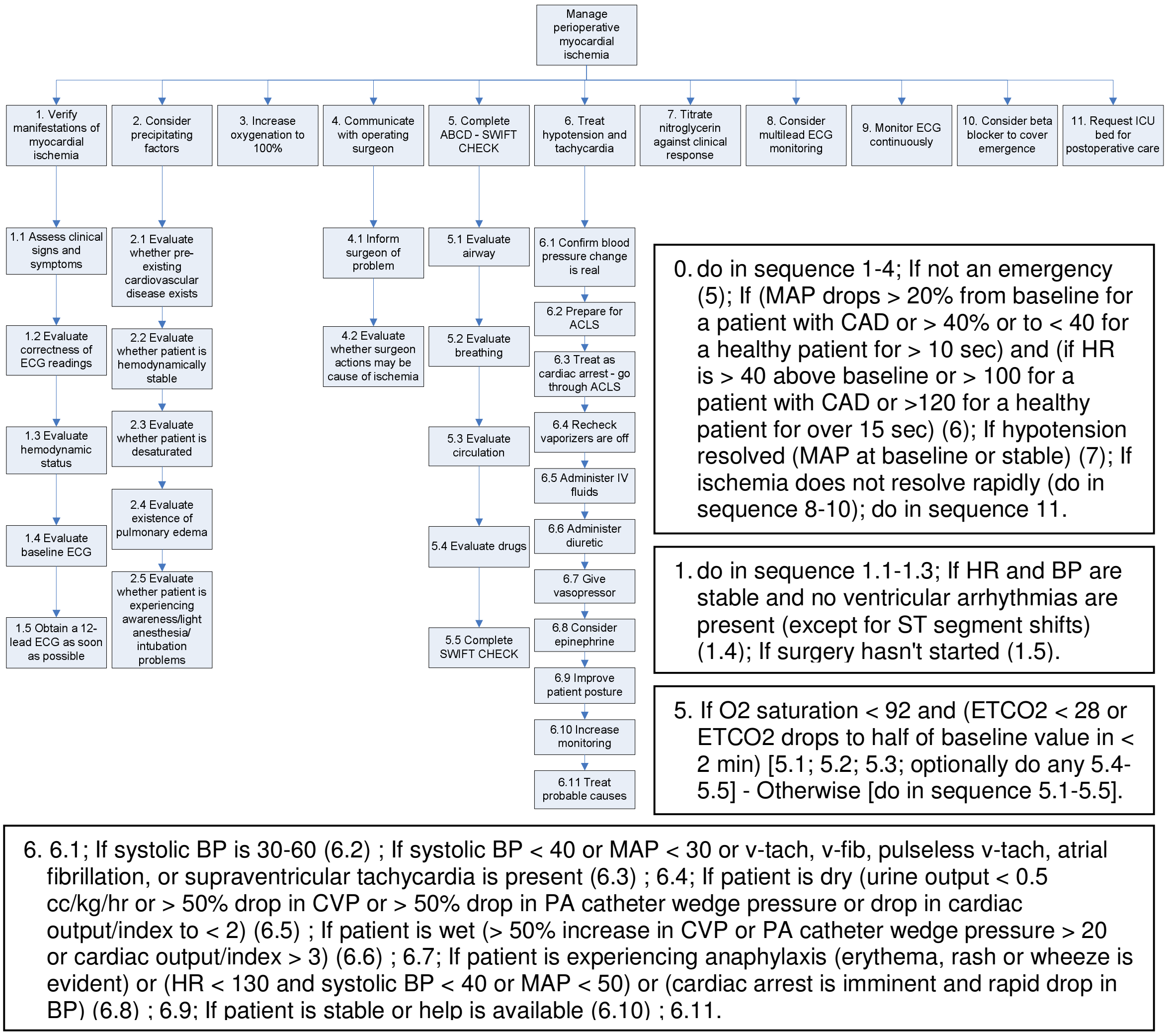


Figure 9. High-level HTA diagram for MI treatment task.

Regarding the high-level limitations of the HTA identified in the Methods section, although this analysis provides a clear sense of the activities as part of anesthesia provider management of the MI crisis and the sequence of tasks, the plans do not provide complete information on the critical decisions on patient states at any time. Furthermore, they do not identify the information requirements the anesthetist may have for addressing each low-level task. The GDTA (see next section) was necessary for providing this information.

5.1.2 Goal-Directed Task Analysis

The high-level goals as part of the GDTA are similar to those of the HTA (see Figure 9). Figure 10 shows the tasks, decisions and information requirements associated only with the subgoal of assessing clinical signs and symptoms, as an example (see Section 3.2).

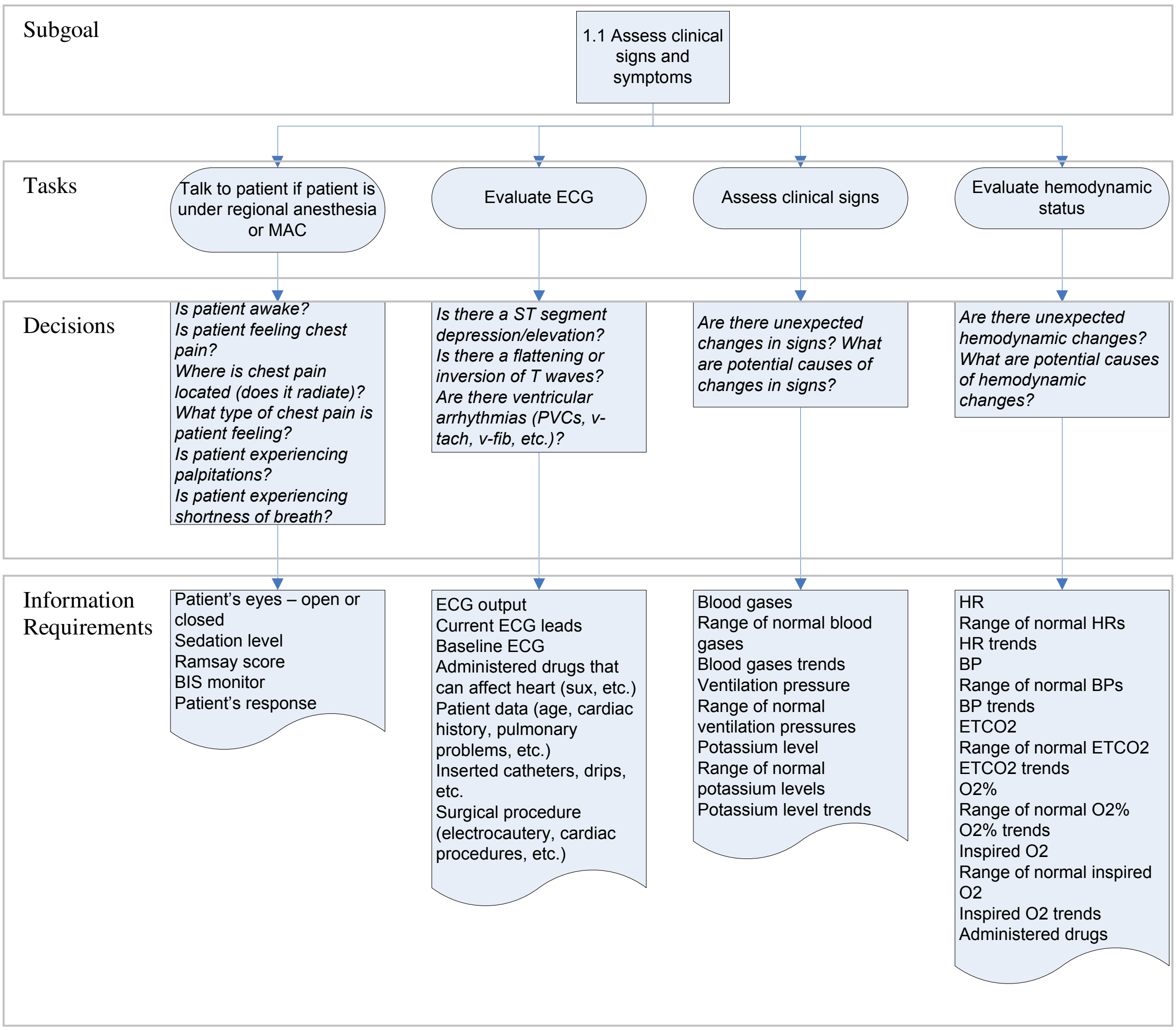


Figure 10. GDTA for subgoal of assessing clinical signs and symptoms.

The outcome from this step was a comprehensive description of the critical decisions and situation awareness requirements of the anesthetist in treating MI. This CTA, along with the HTA which describes the procedures related to this task, served as a basis for developing a cognitive model. The complete GDTA is presented in Appendix E. Again, its high-level goal is perioperative MI management and it includes 11 goals identical to the HTA high-level tasks, two subgoals and 21 tasks. There are 83 decisions, an average of approximately four per task, and 206 information requirements, approximately 2.5 per decision.

With respect to the high-level limitations of the GDTA identified in the Methods section, although the analysis resulted in many information requirements for the anesthetist in MI crisis management, the results of the HTA or a technology inventory (e.g. AH; Segall, Green, & Kaber, 2006) are necessary to provide information on sources that the anesthetist may use to address information needs. Similarly, the HTA results are needed to complement the GDTA findings by giving the analyst a sense of when certain information requirements are critical to situation awareness and performance.

5.1.3 GOMSL Model

The detailed descriptions of the MI treatment task that resulted from the HTA and GDTA were used in this study to guide the development of the GOMSL code. The outcome from this step was a high-level GOMSL model describing the MI treatment task in terms of user goals, methods, decisions, and actions. The GOMSL code is presented in Appendix F. It consists of 13 methods, one selection rule (to route control to the appropriate treatment algorithm based on current patient state) and 136 steps. Each method is between 2 and 19 steps long, for an average of approximately 10.5 steps per method.

Figure 11 shows the GOMSL code for part of the MI treatment task, specifically, deciding whether to begin advanced cardiac life support (ACLS). ACLS is a treatment algorithm endorsed by the American Heart Association involving cardiopulmonary resuscitation (CPR) and defibrillation, among other interventions. This decision is represented in sections 6.2 and 6.3 of the HTA (Appendix D) and section 6 of the GDTA (Appendix E). If systolic blood pressure falls below 60 mm Hg, the anesthetist should prepare for ACLS (e.g., set up the defibrillator). If systolic blood pressure falls below 40 mm Hg, mean arterial pressure (MAP) falls below 30 mm Hg, or severe arrhythmias are present (e.g., atrial fibrillation), the anesthetist should carry out the ACLS algorithm.

The GOMSL code simulates this thought process. The modeled anesthetist searches for variables such as systolic blood pressure (which are displayed in patient monitors, but also in the ADST interface) and, based on their values, decides whether to prepare for or go through ACLS. When the anesthetist decides on a certain action, this action is displayed as a recommended treatment step in the ADST (see Steps 1-3 in example GOMSL code, Figure 11). In general, the ACLS preparation method is performed step by step. An Accomplish_goal statement is used to call out the method, and after it is completed, a Return_with_goal_accomplished statement is used to return control to the higher-level method.

```
Method_for_goal: Prepare for_ACLS
   Step 1. Look_for_object_whose Label is sys_BP and_store_under
      <sys_BP>.
   Step 2. Decide: If Value of <sys_BP> is_less_than "60", Then Type_in
      "24". //Prepare for ACLS - Systolic BP is < 60
   Step 3. Decide: If Value of <sys_BP> is_less_than "40", Then Type_in
      "25". //Treat as cardiac arrest - go through ACLS - Systolic BP
      is < 40
   Step 4. Look_for_object_whose Label is MAP and_store_under <MAP>.
   Step 5. Decide: If Value of <MAP> is_less_than "30", Then Type_in
      "26". //Treat as cardiac arrest - go through ACLS - MAP is < 30
   Step 6. Look_for_object_whose Label is ECG and_store_under <ECG>.
   Step 7. Decide: If Value of <ECG> is_not "V tach",
                and Value of <ECG> is_not "V fib",
                and Value of <ECG> is_not "Pulseless v tach",
                and Value of <ECG> is_not "Atrial fib",
                and Value of <ECG> is_not "Supraventricular tach",
      Then Return_with_goal_accomplished.
   Step 8. Type_in "27".      //Treat as cardiac arrest - go through
      ACLS - Severe arrhythmias present
   Step 9. Delete <sys_BP>; Delete <MAP>; Delete <ECG>.
   Step 10. Return_with_goal_accomplished.
```

Figure 11. GOMSL code for subgoal of preparing for ACLS.

When this model is run in EGLEAN, it outputs the human behaviors based on data input through the Java interface. A time stamp is associated with each behavior. Access is also provided to threads, variables and buffers as they change during run time.

As noted in the Methods section, one of the high-level limitations associated with GOMSL is its assumption of skilled user behavior. However, task representation using linear, error-free actions is not always accurate for describing anesthetist behavior, which may be cyclical in nature. For example, the anesthesia provider typically hypothesizes a reason for an observed problem, decides on a potential solution and tests it, and observes whether the appropriate result was achieved – a “trial and error” approach. In the GOMSL model developed as part of this research, such behavior was simulated by integrating exceptions in the methods, causing the human processor to periodically check the task display to verify that the current diagnosis had not changed (see Section 5.1.5).

5.1.4 Ecological Interface

For the purpose of this research, an ecological ADST interface was prototyped in Java. The interface consists of two main sections: one section displays patient variables that are relevant to the diagnosis and treatment of MI and the other displays a suggested diagnosis and recommended treatment steps (see Figure 12 below). Numerical and textual patient variables, which represent a human patient and change over time, originate from the scenario files described above (see Appendix G). The GOMSL model simulating anesthetist behavior (Appendix F) reads these physiological variables (e.g. "Look_for_object_whose...") from the Java interface and outputs suggested diagnoses and treatment steps based on their values.

The interface addresses the need identified by Hajdukiewicz et al. (2001), i.e. the lack of information provided to anesthetists at high levels of abstraction and a broad range of aggregation levels. This was achieved by designing an interface that integrates only a subset of patient variables, critical to detecting the onset of MI (e.g., heart rate, blood pressure, inspired oxygen, etc.) and suggests and explains a recommended course of action. The source of decision and action recommendations is the rule-based ADST.

When the ADST is first started, the user is prompted for the patient's baseline heart rate and blood pressure. The ecological Java-based interface is then presented. The interface is comprised of four windows:

- *Patient variables window* (top left). Presents patient physiological variables that are relevant to diagnosing and treating MI. During run time, the variables are updated approximately every 5 seconds from the scenario files. As postulated in the EID framework, normal ranges for a healthy patient are displayed for each variable in brackets, representing single variable constraints. Multivariate constraints are illustrated by use of color. The values of the patient variables are black when patient state is normal. When the tool diagnoses a problem, the values of the variables that were considered as a basis for the diagnosis are highlighted in the same color as the diagnosis (displayed in the Diagnosis window on the top right). In addition, if treatment steps (shown in the Treatment steps window on the bottom right) are related to certain variables, the relevant treatment step and variable are highlighted in the same color. If multiple variables are relevant to a certain treatment step, lines are also used to connect the variable displays. Means-end relationships among variables in the ADST interface are reflected in the grouping of data fields by traits of the heart (e.g., ECG), circulation (e.g., mean arterial blood pressure), and respiration (end-tidal CO_2). The purposes of the system are associated with these processes.
- *Diagnosis window* (top right). Presents the most probable diagnosis, as well as how this diagnosis was derived. Red is used to indicate critical patient states, orange indicates a severe problem, and green indicates that patient state is normal.
- *Treatment steps window* (bottom right). Displays recommended treatment steps for the problem diagnosed by the tool, as well as explanations for these recommendations. When these explanations relate to certain patient variables, the text and relevant variables are highlighted in the same color to emphasize this

relationship. The treatment algorithm is updated continuously based on changes in patient variables.

- *ABCD window* (bottom left). Lists ABCD treatment steps, when they are part of the treatment algorithm displayed in the Treatment steps window. ABCD is a mnemonic for memorizing resuscitation steps: airway, breathing, circulation, and drugs. Each of these steps is tailored to the patient's current condition as diagnosed by the tool. Similar to the Diagnosis window, red is used to indicate critical patient states, orange indicates a severe problem, and green indicates that patient state is not severe.

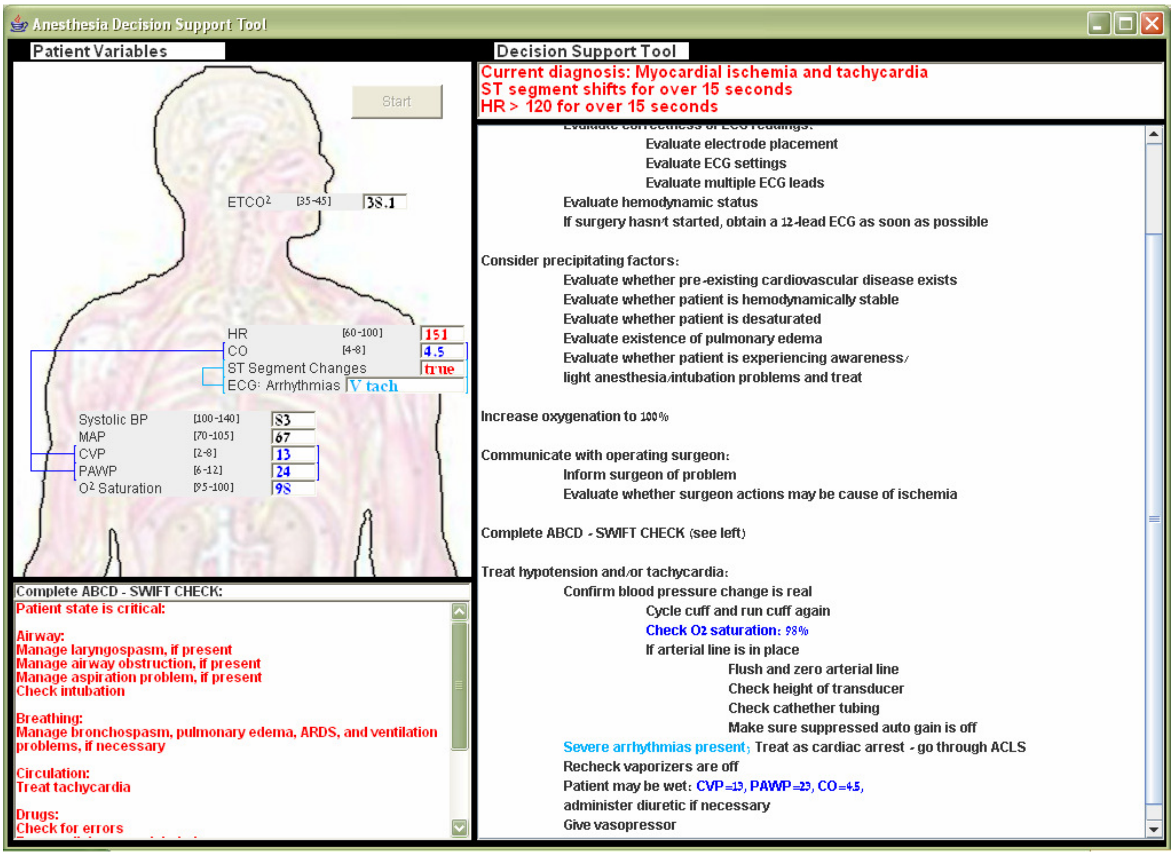


Figure 12. ADST interface.

The interface was designed as a supplement to, and not a replacement for, existing OR monitors (such as waveform displays). There has been a great deal of research on these types of displays (e.g., Blike et al., 1999; Crawford et al., 2003) and it is not an objective of this research to design an improved waveform display. Since the anesthesia workstation is already comprised of a large number of monitors and alarms (Watt et al., 1993), the display will need to compete for the anesthetist's attention. Therefore, when the ADST determines the patient status is acceptable, the interface is passive, only presenting a green "normal" status indicator. When the tool detects that MI may be developing through the integration of data on multiple physiological variables, a diagnosis (e.g., MI) in the form of a salient text message is used to alert the anesthetist. Due to the abundance of OR auditory alarms and the difficulty for practitioners to differentiate them (Loeb et al., 1992), designing a unique auditory alarm to alert anesthetists of a developing crisis is a challenging task and was beyond the scope of this research.

5.1.5 The GOMSL Model and Java Code in Run Time

As described earlier, the GOMSL code (Appendix F) drives the ADST treatment recommendations. The code checks the current diagnosis repeatedly and, when the ADST diagnoses that the simulated patient is hypotensive or suffering MI, it begins stepping through the corresponding treatment algorithm. Some steps are dependent upon certain physiological variables. For example:

```
Step 6. Look_for_object_whose Label is ECG and_store_under <ECG>.
Step 7. Decide: If Value of <ECG> is_not "V tach",
                and Value of <ECG> is_not "V fib",
```

```
and Value of <ECG> is_not "Pulseless v tach",
and Value of <ECG> is_not "Atrial fib",
and Value of <ECG> is_not "Supraventricular tach", Then
                    Return_with_goal_accomplished.
Step 8. Type_in "27".
        //Treat as cardiac arrest - go through ACLS - severe arrhythmias
        present
```

Here, the GOMSL code searches the interface for the ECG waveform (Step 6). If no severe arrhythmias (such as ventricular tachycardia) are detected, control is returned to the higher-level method (Step 7). If any form of severe arrhythmia is present, the next line of code (Step 8) is executed: a treatment step is output to the Treatment steps window of the ADST interface recommending carrying out the ACLS algorithm. The reason for this recommendation is also explained – the presence of arrhythmias. This is done using the Type_in GOMSL operator, which represents text entry to an interface. Rather than entering the entire treatment step (i.e. Type_in "Treat as cardiac arrest – go through ACLS - severe arrhythmias present"), only a number is entered (Type_in "27"). This is done because GOMSL requires 300 msec for each character; that is, printing the text in the example to the ADST would require 21 seconds. This rate of displaying information is too slow for crisis management. Therefore, a short code is entered instead, requiring up to 600 msec, and the ADST interprets this code and displays the full text. This piece of code corresponds to section 6.3 in the HTA and section 6 in the GDTA, specifically the decision whether to treat the situation as cardiac arrest.

However, due to current GOMS code limitations, some types of decision-making cannot be handled by the model, including complex computational operations. In these cases, control is passed to the Java code, which is capable of more complex computations. For example, some anesthetist decisions are made based on values of baseline variables, including heart rate and blood pressure, which are measured before surgery (see patient information sheet in Appendix C). When starting the ADST, the user is prompted for these values. When the HTA and GDTA call for comparing current values to the baseline, these calculations are carried out in Java. Another decision managed in Java involves checking whether a value has changed over the past 2-3 data points, i.e. 10-15 seconds (since patient variables are updated every 5 seconds). For example, in the presence of MI, nitroglycerin should only be administered to the patient if MAP (a function of systolic and diastolic blood pressure) is stable or at baseline. To evaluate MAP stability, the Java program compares the previous two MAP data points to the current data point, to determine whether significant deviations have occurred. To evaluate if MAP is at baseline values, the current MAP data point is compared to the baseline MAP.

For the reasons described above, diagnosing patient state is also handled in the Java code rather than the GOMSL model. In interviews, anesthesiologists indicated that they would like to be notified about extreme changes in blood pressure within 10 seconds, extreme changes in heart rate within 15 seconds, and the presence of ST segment shifts (indicative of myocardial ischemia or MI) also within 15 seconds. Therefore, the diagnosis of MI accompanied by tachycardia, for example, involves the Java code checking for the presence of ST segment shifts and rapid heart rate in the current and previous two data points of a scenario file. The diagnosis is updated every 5 seconds, following the retrieval of a new set of data points (patient variables) from the scenario file.

In interviews, anesthesiologists also stated that they scan patient monitors for changes in patient state approximately every 10 seconds. In GOMSL, checking the current diagnosis repeatedly and updating the recommended treatment procedure accordingly is done using error handling mechanisms. The model begins by checking the diagnosis offered by the Java code. A selection rule is used to route control to the appropriate method (Normal state, Hypotension state or MI state) and the treatment algorithm is output to the ADST, step by step. Every few steps in the GOMSL code, an error exception is raised and the diagnosis is checked again. If it has not changed, the error thread is terminated and control is returned to the point in the code at which the exception was raised. If the diagnosis has changed, the model restarts. The Treatment steps window is cleared and a new treatment algorithm is output to the ADST for the new diagnosis. Error exceptions for rechecking the diagnosis are raised every 4 to 24 steps, representing up to 8.2 seconds between anesthetist scans of patient monitors.

5.2 Preliminary Validation

Three anesthesiologists volunteered to complete both the applicability assessment and the heuristic evaluation of the ADST. The anesthesiologists were faculty members of the Duke University Department of Anesthesiology practicing medicine at Duke University Hospital. They were recruited through the HSPSC. They had an average of 13 years of clinical practice and all had treated a patient for perioperative MI in the past. In addition, two usability experts

– who had attained doctorates in human factors – carried out a heuristic evaluation of the ADST interface. The results of these analyses are presented below.

5.2.1 Applicability Assessment

Figure 13 summarizes results of the survey completed by anesthesiologists. Survey items are shown on the vertical axis and the rating scale is on the horizontal axis. Statements that pertained to a specific scenario – MI or hypotension – were rated twice, and five general statements about the ADST were rated once, after anesthesiologists had viewed both scenarios. In general, anesthesiologists rated the ADST favorably, giving it an average score of 6.4 on a scale from 1 (strongly disagree) to 9 (strongly agree). As hypothesized, they found the tool to be useful (score of 7) and indicated that they would use the tool in the OR (score of 7.7) if further refined.

Essential features anesthesiologists thought should be added to the ADST include:

- *Complete differential diagnosis*. In addition to the most likely diagnosis, a list of all possible diagnoses ranked by likelihood, with supporting and refuting evidence. This could be achieved, for example, by integrating the ADST with a tool such as MedWeaver (Detmer, Barnett, & Hersh, 1997). When a physician enters clinical findings into MedWeaver, it displays a list of possible diagnoses and, upon request, a description of each diagnosis and why it appears on the list. Related medical literature and web sites can also be accessed using this software. For immediate crisis management, as provided by the ADST, the clinical findings could automatically feed into MedWeaver for a complete differential diagnosis in real-time.

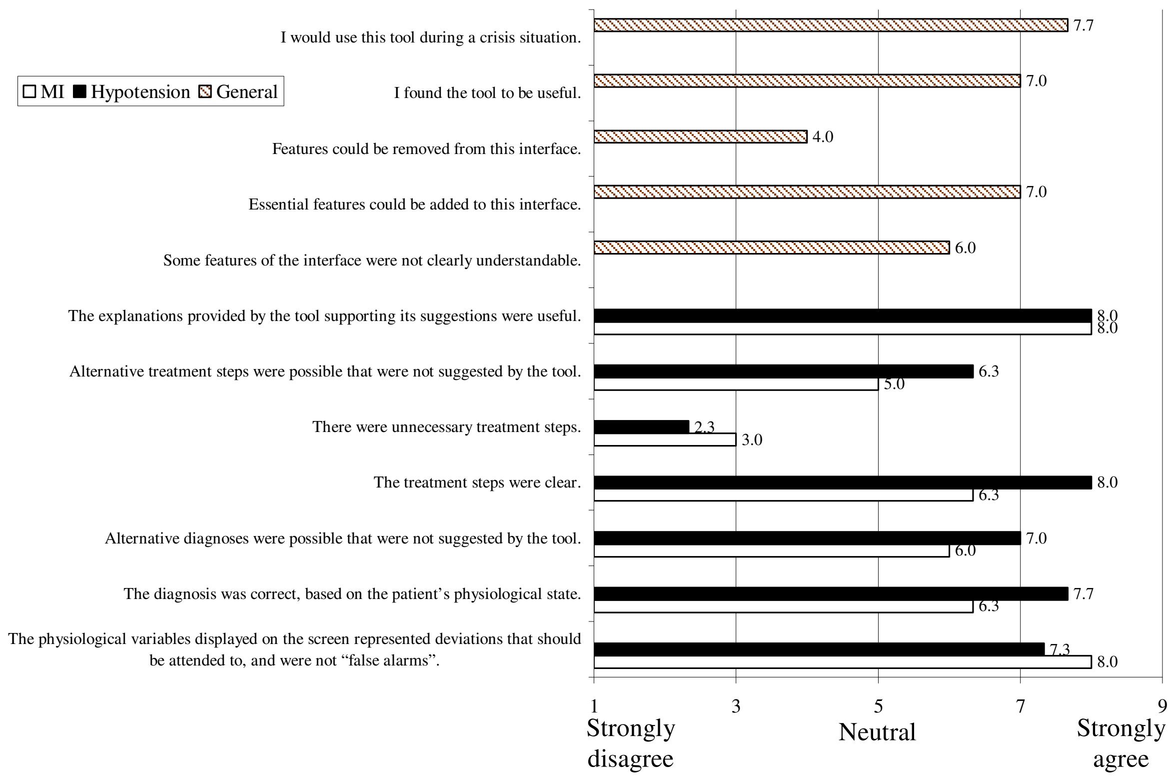


Figure 13. Applicability assessment results.

- *Links to educational material*. When the crisis has passed, junior anesthesia providers could use the ADST to learn more about the problem they had just treated based on a system performance record. Again, a tool such as MedWeaver could be used to this end.
- *Countdown timer for tissue injury*. The timer should start when patient is hypotensive or when treatment steps call for ACLS. This will inform of the possibility of permanent tissue damage. For example, irreparable brain damage can be caused after 6 minutes of low perfusion to the organ.
- *Method for checking off completed treatment steps*. Anesthetists should be able to click on steps they had completed or considered; these should then appear grayed out

or crossed off. In addition, if a new treatment step appears, it should be made more salient than “old” steps.

- *Reorganize information in Treatment steps window*. Suggestions included grouping steps by interventions, manifestations, precipitating factors, etc. or by cognitive (think-about) versus psychomotor (do) steps.

Fixation errors often occur when anesthetists do not treat the critical problem at hand because of attention or actions directed at other efforts (Weinger, 1999). It was noted that the ADST could help anesthesia providers in this respect, by allowing them to think flexibly about alternative diagnoses or treatment options. The anesthesiologists also made many comments about the clinical correctness of the information presented by the ADST. For example, two anesthesiologists stated that rather than having “ST segment changes” in the Patient variables window, they would like to see the magnitude and direction of these changes, which can provide information about the severity of the problem and whether the patient is experiencing myocardial ischemia (ST segment depression) or infarction (elevation). Another anesthesiologist said that the statement “put [patient] in Trendelenburg position” should not be displayed if the patient is wet (hypovolemic). It was also noted that it is not very likely that a healthy 33-year old man undergoing ankle surgery (see patient information sheet in Appendix C) would suffer massive hemorrhaging (leading to hypotension) or MI.

5.2.2 Heuristic Evaluation

The anesthesiologists and usability experts made a total of 22 unique remarks about the ADST interface. For each heuristic, the following comments were made (numbers in parentheses represent severity – the number of evaluators who made the comment, and letters represent the source of the comments – usability experts (U) or anesthesiologists (A)):

- *Simple and natural dialog*. The interface should not contain irrelevant or rarely needed information. All information should appear in a logical order.
  - Layout and grouping of the physiological data is natural and intuitive (1 U).
  - Color coding and lines connecting data and related treatment steps are good (1 U).
  - Rather than grouping physiological variables in the Patient variables window by traits of the heart, circulation and respiration, a more natural arrangement would be to organize them as rhythm-related, hemodynamic and respiratory (1 A).
- *Speak the users' language*. Concepts and terminology should be taken from the anesthesiology domain.
  - Simple terminology in a crisis situation is best. I would advise against using an excessive amount of medical terminology (1 A).
  - Language is from domain of anesthesiology (2 U).
  - Use "hypovolemic" instead of "dry", "fluid overloaded" instead of "wet" (1 A).
  - Tachycardia diagnosis should be relative rather than absolute. An increase of 50% over baseline values should be labeled as tachycardia, rather than a heart rate of 120 beats per minute (2 A).
- *Minimize users' memory load.* Users should not have to remember information from one screen to another.

- There is too much text data in Treatment steps window for use in actual circumstances; the same is true for the ABCD window. Use graphical icons for representing information and/or clickable summary statements integrated with a touchscreen. These types of changes may also allow for re-layout of windows (1 U).
- The display refresh rate is too rapid – can't read all the text (4 U,A).
- In refresh of treatment steps, it isn't easy to tell when/where steps have changed (1 U).

- *Consistency*. Users should not have to wonder whether different words, situations, or actions mean the same thing.
    - I believe that the consistency is quite good (2 A).
    - In the Diagnosis window, red represents critical patient states, orange indicates a severe problem, and green indicates that patient state is normal. In the Patient variables and Treatment steps windows, what do blue, cyan, etc. represent? Either note in the user manual that these colors are only used to associate treatments to variables and have no meaning with relation to patient state or, preferably, add a legend in the Patient variables window (1 U).
    - Use shades of green, rather than blue, in Patient variables window (1 U).
    - The interface is limited (which is a good thing), so users don't have to deal with multiple screens/functions that would make consistency a problem (1 U).
- *Feedback*. The system should keep the users informed about what is going on, within reasonable time.

    - Perhaps it would be better to display all treatment steps at the same time without requiring scrolling (3 U,A).
    - Update rate is good. However, the display should be persistent if a specific condition persists (1 U).
    - Nicely visible as to what information is being used to generate treatment step (1 U).
- *Prevent errors*. The design of the system should prevent errors from occurring.
    - Use a sans serif font throughout (2 U). (This recommendation is supported by Degani (1992), who reports on a body of research finding that sans serif font is more legible than serif font.)
    - Reading small text may be difficult in the OR (2 U,A).
    - Low contrast colored text in ABCD window can be difficult to read. It is recommended that you only color the header (patient state), leaving the remaining text in black (1 U).
    - Information in the ABCD window is not salient enough (1 A).
    - The word vasopressor (a type of drug) may be mistaken for vasopressin (a specific drug) (1 A).

Overall, evaluators provided both positive feedback on the ADST interface, as hypothesized, and negative feedback. Negative usability comments that were made by more than one evaluator pertained to the rapid refresh rate of the Treatment steps window, which prevented reading all the text; the lack of ability to check off steps which had already been completed; and the limited readability of treatment steps due to small fonts and the large amount of text requiring scrolling. However, many positive comments were made although they were not solicited (the evaluation only called for identification of violations of usability principles), indicating evaluators' approval of the ADST interface. Their favorable opinion of the interface was also conveyed verbally to the analyst during the evaluation sessions.

6. Conclusions

The goal of this research was to apply a novel methodology to ADST development based on established human factors techniques – task analyses, cognitive modeling, an interface design framework, and usability principles. This approach was implemented in the domain of anesthesiology, resulting in an ADST with the potential to support anesthetist decision-making in crisis management. It was later validated by both domain and usability experts.

As noted by Kieras (1999), GOMS modeling begins after a task analysis. For this purpose, interviews with experienced anesthesiologists were carried out, along with observations of anesthesiology residents' crisis management training using a patient simulator. The interviews proved to be more useful, eliciting deeper, more insightful information about both cognitive and procedural behavior in crisis detection, diagnosis and treatment. However, this method has its limitations – anesthesiologists were not inclined to articulate "tacit knowledge" (Klein et al., 1989) such as what heart rate constitutes tachycardia and what period of tachycardia would cause them to be concerned. With respect to the observation sessions, it would have been useful to have the help of an experienced anesthetist in explaining, for example, whether residents correctly diagnosed the problem they were facing, whether their interventions were effective, what were the effects of administered drugs, etc., in order to have a clear sense of correct cognitive and procedural behaviors. Additional knowledge elicitation methods, such as asking anesthesiologists to think aloud (Hoffman et al., 1995) as they diagnose and treat MI on the patient simulator, may have provided further information for the task analyses. However, due to resource limitations, this type of knowledge acquisition was not possible.

The task analyses that resulted from the knowledge elicitation served as a basis for the GOMSL cognitive model describing anesthetist behavior. The structured form of the HTA and GDTA proved to be constructive for model development. It was relatively straightforward to translate HTA tasks and operations into GOMSL steps and operators; decisions and information requirements in the GDTA were useful for modeling decision-making using the Decide operator in GOMSL.

However, applying GOMSL to DST development has its limitations. Though GOMSL can model human behavior, often expert systems are called in to provide functionality of which humans are not capable, such as performing complex calculations. When such functionality was required of the ADST, it was not done in GOMSL. Instead, these calculations were carried out in Java, a standard computer programming language. Combining the capabilities of GOMSL with those of a programming language produced a powerful tool for expert systems development. Another avenue worth exploring is using GOMS in conjunction with an expert systems language such as Lisp for this purpose. For example, GOMSL models can be translated to the ACT-R computational cognitive language (Anderson et al., 2004), which is implemented in Lisp, using a compiler tool called G2A (St. Amant & Ritter, 2004).

The design of the interface for the ADST was grounded in EID principles. The EID framework provided guidance on data content, grouping and organization, but it is not intended to support design for usability, i.e. it does not address issues such as context sensitivity, visual momentum and dialog (Vicente, 2002; Vicente & Rasmussen, 1992). In addition, the intent of the EID approach is only to provide operators with the necessary information to diagnose an unanticipated problem; responsibility for detection, diagnosis and treatment of the problem is left in the hands of the operator (Vicente & Rasmussen, 1992). For these reasons, ecological design efforts were focused mainly on the Patient variables window. Other parts of the interface were text-based and were directed at supporting decision-making aspects of the crisis management task not addressed by the EID framework. In general, the applicability of EID to the medical domain is slightly restricted by limitations on understanding of the human body in terms of physical laws and by the limited number of available sensors (Sharp & Helmicki, 1998).

The preliminary validation carried out in order to evaluate the usefulness and usability of the ADST elicited many insightful comments from the domain and usability experts. In the applicability assessment, anesthesiologists commented on issues related to ADST content, e.g. "...may want to have a 'prepare for' prompt" for treatment steps such as administering IV fluids; they also commented on formatting-related issues, e.g. "scrolling text is hard to follow". Similarly, they noted both format and content issues in the heuristic evaluation. For example, one anesthesiologist cautioned to "be careful of possible mistake of medicines – vasopressor versus vasopressin" with respect to the error prevention heuristic. Comments pertaining to ADST content were an unexpected but welcome outcome of the heuristic evaluation, which was developed with the goal of finding interface usability problems.

The ADST developed as part of this research was expected to improve on other tools in the anesthesiology domain. Typically, interfaces whose design is based on human factors methods and principles, such as object displays, are aimed at enhancing anesthetists' ability to detect changes in patient state. They do not directly support higher-level information processing functions (information analysis and decision-making), i.e. diagnosing and recommending a treatment procedure, as the ADST does. This is also true of intelligent alarms, which warn of problems when deviations from a normal patient status are detected. As for AI-based expert systems, the ADST has explanatory powers (e.g. when a treatment step is dependent upon the value of some physiological variable, the variable and treatment step are highlighted in the same color), which methods such as neural networks cannot provide. Although research has shown decision support systems to improve the quality of clinical decision-making, their acceptance has been limited in the medical community, in part due to a lack of understanding of their underlying logic (Lai, Macmillan, Daudelin & Kent, 2006). Thus, decision explanations can make the system more comprehensible to users and promote acceptance of the tool (Huang & Endsley, 1997). Expert systems developed using techniques that have the capability to explain their decisions, such as rule-based systems, have not typically made use of structured human factors methods for constructing knowledge bases or for interface design. It should be noted that some researchers are calling to limit the use of decision aids in safety-critical systems such as health care, since they will never be able to account for all unanticipated events that can occur, and may therefore give imperfect advice (Vicente, 2003).

There are several social factors that may promote (or hinder) operator acceptance of automation in general and DSTs in particular. One of these is trust. Sheridan (1992) lists seven attributes of trust in technology, several of which are relevant to anesthesia provider acceptance of the ADST:

- *Reliability*. Operators should observe repeated, consistent system behavior under similar circumstances. With respect to the ADST, this means that recommendations should remain the same when the same patient behavior, as documented in changes in vital signs, is witnessed.
- *Robustness*. The system should be able to perform in a variety of circumstances. This is one of the ADST's properties: it can tailor its diagnosis and recommended treatment procedure to changing patient states.
- *Familiarity*. The system should use procedures and terms which are familiar and natural to the user. This concept parallels one of the usability heuristics by which the ADST was evaluated, that of speaking the users' language. However, familiarity can also engender irrational trust in the system.
- *Understandability*. Operators should be able to form a mental model to predict future system behavior. A lack of understanding may account for irrational distrust in automation. Understandability of the ADST is promoted by its recommendation explanations.
- *Dependence*. Operators should depend on the system, but dependence should only be placed upon systems that warrant trust. Dependence can also lead to obedience to the decision aid, causing operators to abandon responsibility for their actions (Sheridan, 2002). This effect is undesired in anesthesiology (and other domains). Anesthesia providers should calibrate their trust in decision aids relative to historical performance and known types of errors. The anesthetist needs to consider all options which they believe may apply to the patient's condition, especially in light of information that may not be available to the ADST.

Other factors that can affect acceptance of medical DSTs include social barriers, such as the perception that it is “bad form” to use these tools, and the role they may play in justifying decisions if legal issues arise (Lai et al., 2006). All these factors need to be taken into account in any implementation of the ADST for actual use in an OR.

6.1 Caveats

There are several limitations to the DST development approach described here. First, it is very labor-intensive: generating task analyses for a single critical incident, MI, was a time-consuming process. It involved many hours of interviews and observations, and many additional hours of coding the information gleaned from these sources. Hoffman et al. (1995) referred to this step in expert systems development as the “knowledge acquisition bottleneck”. Therefore, designing a ADST that would account for every crisis listed in Gaba et al.’s (1994) book was not feasible as part of this dissertation.

Writing the GOMSL model and Java code to provide the ADST with the decision-making functionality captured in the task analyses was also time-consuming. In addition to entering the treatment steps, decisions had to be made about how to allocate functions between the two tools, primarily because of the limitations of GOMSL. For novices, there is also a learning curve associated with Java, GOMSL and linking Java devices to GOMSL models using EGLEAN.

The task analysis methods used in this research have several limitations. HTA, though considered to be a useful tool, requires an extensive amount of training and practice to master

(Stanton & Young, 1998). It was also found to be more time-intensive than other research tools (Stanton & Stevenage, 1998). Learning how to create a GDTA, as well as eliciting decisions and information requirements from subject matter experts, can be challenging processes. Furthermore, this tool focuses on operator information needs, not on how they should be acquired (Endsley, 1993). For example, some information requirements gathered through the MI treatment GDTA – such as heart rate and ECG trends – are available through existing OR displays, while others, e.g. patient responses and surgical actions, can only be obtained from a direct visual scan of the anesthetist's environment. GDTAs also do not address temporal variations in information requirements (Endsley, 1993), that is, some elements are more important at certain times during the task, while at other times, they may hold less significance for operators. In the GDTA for MI crisis management, the state of the arterial line, for example, is not important if it has not been inserted. For this reason, GDTA was used alongside HTA, which structures tasks and operations sequentially and can address changes in environmental cues.

There are also drawbacks to using GOMSL (rather than an expert systems language) for DST prototyping. As mentioned above, since GOMSL simulates human cognition, there are some complex calculations it cannot carry out and these must be managed externally. Since it was originally developed with the purpose of modeling human-computer interaction, GOMSL is not well-equipped to model anesthetist actions (motor control behaviors) such as drug administration. For this reason, the time estimates produced by the GOMSL model are not accurate and cannot be used to predict time to task completion. Kieras (personal communication) is currently adding new operators to GOMSL to represent a broader range of human physical behaviors (e.g. transporting, walking) and to promote applicability of the modeling technique beyond human-computer interaction tasks. Unlike other GOMS variants (e.g. CPM-GOMS), GOMSL also cannot handle parallel processing; therefore, the simulated anesthetist can't re-assess its diagnosis continuously while treating the patient, but must stop the treatment every few steps to check for changes in patient state. Therefore, it may be worthwhile to consider other advanced computational cognitive modeling techniques, such as ACT-R, for future enhancement of the ADST engine. ACT-R is capable of modeling parallel processing and, being a lower-level language than GOMSL, it can model behavior in much more detail, allowing for more accurate task time estimates (Anderson et al., 2004).

In general, GOMSL may not be particularly well suited for modeling complex anesthetist decision-making. Two patients, exhibiting similar trends in vital signs, may experience entirely different problems, or react differently to the same treatment. Iterative problem-solving cycles of hypothesizing the source of the problem and treating the theorized diagnosis may be necessary in order to bring the patient to a safe and stable state. This type of stochastic, non-linear behavior is difficult to model in GOMSL, which is better suited for structured, sequential actions (Kieras, 1999).

Finally, there is a long way to go before the prototype ADST can be used in actual OR settings. The treatment algorithm needs to be updated to reflect comments made by anesthesiologists during the applicability assessment. It should also be expanded to deal with other complications of MI in addition to tachycardia and hypotension. Extending the tool to other OR crises would increase its usefulness. This would require much larger HTA and

GDTA structures. A comprehensive crisis management HTA would have many more tasks and operations for addressing every possible deviation from the normal patient state. The plans as part of the analysis would become more detailed, synthesizing multiple environmental conditions and patient variables for selecting the appropriate tasks. Likewise, a comprehensive GDTA for all OR crises would include new subgoals and tasks that are not part of the current MI treatment GDTA. Decisions and information requirements would be associated with each new task. Decision mechanisms in the ADST, currently handled by the GOMSL and Java code, would become more detailed as a function of the number and complexity of HTA plans, as well as the type of GDTA information requirements. In particular, the code for determining patient state would need to incorporate additional variable trends and factors to arrive at the most likely diagnosis. Finally, the ADST interface would need to be expanded to include additional patient variables that were deemed less important for treating MI but would be crucial for diagnosing and treating other problems.

To efficiently expand the ADST as described, the data acquisition and modeling processes would need to be streamlined. One method for reducing the time required to produce task analyses for each crisis would be to train experienced anesthetists in HTA and GDTA and ask them to create such documents for specific crises. These analyses would then be evaluated by their peers for completeness and correctness, as well as a human factors professional.

The interface prototype also needs to be changed to address the heuristic violations noted by evaluators. The text presentation method, in particular, should be modified. The text refresh rate should be reduced and, if possible, scrolling should be eliminated to allow hands-free interaction with the tool. These steps should be carried out in an iterative design cycle, in which anesthesiologists would evaluate the ADST after changes are made to it. On a more functional level, drivers would need to be included in the Java code for accepting data from external OR sensors on patient states, versus using a simulated scenario file. Otherwise, the scenario file could be written by sensors in real-time and read by the Java code in near real-time for GOMSL model processing. When the tool is ready for use in the OR, it would need to be approved by such organizations as the American Heart Association and the Food and Drug Administration.

6.2 Future Research

Several research directions would be interesting to pursue with regard to the prototype ADST. First, many features could be added based on the suggestions made by the evaluators, including implementing a countdown timer for permanent tissue damage, displaying a complete differential diagnosis, etc. Other features could be added as well, such as:

- *Information about specific drugs and doses*. In the current ADST, reference is made to generic drug groups, such as vasopressors. It would be possible to provide additional information about these drugs, e.g. hyperlinks to the varied vasopressors, their recommended doses, side effects, etc. However, this also increases the interactivity of the tool, which is likely not to be exploited under crisis situations.
- *Incorporating the ACLS algorithm into treatment steps*. The American Heart Association has established treatment algorithms for a variety of cardiac emergencies in the form of flow charts. These could be adapted to the perioperative environment

and incorporated into the MI treatment procedure when it calls for going through ACLS.

- *Patient information*. In diagnosing and recommending treatment steps, it would be useful if the ADST had access to information such as the patient's age, gender, history of CAD, and surgical procedure. This would enable more accurate recommendations and explanations.
- *Information presentation*. Several evaluators commented on the heavy reliance on text for conveying diagnosis and treatment information, which made following the treatment algorithm difficult. Other presentation methods may be perceived more easily, such as graphical icons, flow charts or abbreviations. Some evaluators suggested segmenting the text, for example into interventions, manifestations, precipitating factors, etc., which may also enhance readability. However, such grouping of information might make display scrolling or interface options selection requisite in the use of the ADST.
- *Alarms*. Some anesthesiologists expressed an interest in being notified when the patient becomes tachycardic, for example. Visual alarms are already provided in the interface, thus it may be interesting to explore the use of additional alarms. For example, an auditory display could be designed to sound varying pitches, indicating severity of patient state. Alternatively, a wireless tactile alarm could be donned by anesthesia providers in the OR.
- *PDAs*. Several evaluators noted that they would like to have a means for checking off recommended treatment steps that had already been completed or considered. To accommodate this request and allow anesthetists to freely move about, the ADST

could be implemented in an arm-mounted PDA. If ORs have wireless internet access, it would also be possible to integrate the ADST with tools such as MedWeaver (Detmer et al., 1997) that display medical literature and web sites related to specific diagnosed problems.

After further refining the treatment algorithm and interface, it would also be interesting to carry out a summative evaluation of the ADST. For example, two groups of experienced anesthetists could be asked to treat perioperative MI on the patient simulator while describing their thoughts and actions. One group would have access to the ADST and the other would not. Variables such as time to detect the existence of the problem, time to diagnose the problem correctly, the appropriateness of interventions (number of incorrect and missing actions), time to treat the patient, and outcomes (patient state following treatment, e.g. whether permanent damage has been caused to the myocardium) could be measured to statistically determine whether use of the decision aid promotes the effectiveness and efficiency of anesthetists in crisis management. Similar user testing could also serve to assess the usefulness of the ADST for training purposes.

Another approach to validating the ADST would involve comparing its output to actual anesthesia provider behavior in a real MI crisis in the OR. Many hospitals have recordkeeping systems which gather information including patient variables, administered drugs, lab results, anesthetist notes, etc. during surgical procedures. Patient variables for a procedure in which the patient suffered MI (and recovered due to correct interventions) could be fed to the ADST via a scenario file, and its treatment steps could be compared to the anesthesia provider's actions as documented in OR notes and administered drugs. If a similar treatment procedure was carried out by the anesthetist and recommended by the ADST, i.e. intervention steps were identical and carried out in the same order, this would provide evidence of the correctness of the MI treatment algorithm, thus validating its development methodology.

7. References

Ahmed, F., Nevo, I., & Guez, A. (1990). Anesthesiologist's adaptive associate (ANAA). In *Proceedings of the 12th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 12*(3) (pp. 949-950). Piscataway, New Jersey: IEEE.

American Society of Anesthesiologists (2005). *Patient Education*. Retrieved May 3, 2005 from http://www.asahq.org/patientEducation.htm.

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060.

Annett, J. (2003). Hierarchical task analysis. In E. Hollnagel (Ed.), *Handbook of Cognitive Task Design* (pp. 17-35). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Beatty, P.C.W., Pohlmann, A., & Dimarki, T. (2000). Shape-only identification of breathing system failure. In *Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2*(2) (pp. 982-984). Piscataway, New Jersey: IEEE.

Becker, K., Käsmacher, H., Rau, G., Kalff, G., & Zimmermann, H.-J. (1994). A fuzzy logic approach to intelligent alarms in cardioanesthesia. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems, 3* (pp. 2072-2076). Piscataway, New Jersey: IEEE.

Becker, K., Thull, B., Käsmacher-Leidinger, H., Stemmer, J., Rau, G., Kalff, G., & Zimmermann, H.-J. (1997). Design and validation of an intelligent patient monitoring and alarm system based on a fuzzy logic process model. *Artificial Intelligence in Medicine, 11*, 33-53.

Blike, G.T., Surgenor, S.D., & Whalen, K. (1999). A graphical object display improves anesthetists' performance on a simulated diagnostic task. *Journal of Clinical Monitoring and Computing, 15*, 37-44.

Block, F.E., Nuutinen, L., & Ballast, B. (1999). Optimization of alarms: A study on alarm limits, alarm sounds, and false alarms, intended to reduce annoyance. *Journal of Clinical Monitoring and Computing, 15*, 75-83.

Burns, C.M. & Hajdukiewicz, J.R. (2004). *Ecological Interface Design*. Boca Raton, Florida: CRC Press.

Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, New Jersey: Erlbaum.

Chaney, M.A. & Slogoff, S. (1999). Perioperative myocardial ischemia and infarction. In J.L. Atlee (Ed.), *Complications in Anesthesia* (pp. 348-350). Philadelphia, Pennsylvania: W.B. Saunders Company.

Cook, R.I. & Woods, D.D. (1994). Operating at the sharp end: The complexity of human error. In M.S. Bogner (Ed.), *Human Error in Medicine* (pp.255-310). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Cook, R.I. & Woods, D.D. (1996). Adapting to new technology in the operating room. *Human Factors, 38*(4), 593-613.

Cooper, J.B., Newbower, R.S., & Kitz, R.J. (1984). An analysis of major errors and equipment failures in anesthesia management: Considerations for prevention and detection. *Anesthesiology, 60*, 34-42.

Crawford, J., Savill, A., & Sanderson, P. (2003). Monitoring the anesthetized patient: An analysis of confusions in vital sign reports. In *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting* (pp. 1574-1578). Santa Monica, California: Human Factors and Ergonomics Society.

Davis, R. & Lenat, D.B. (1982). *Knowledge-Based Systems in Artificial Intelligence*. New York, New York: McGraw-Hill.

Degani, A. (1992). On the typography of flight-deck documentation. (NASA Contractor Report 177605). Moffett Field, California: NASA Ames Research Center.

de Graaf, P.M.A., van den Eijkel, G.C., Vullings, H.J.L.M., & de Mol, B.A.J.M. (1997). A decision-driven design of a decision support system in anesthesia. *Artificial Intelligence in Medicine, 11*, 141-153.

Detmer, W.M., Barnett, G.O., & Hersh, W.R. (1997). MedWeaver: Integrating decision support, literature searching, and Web exploration using the UMLS Metathesaurus. In *Proceedings of the 1997 AMIA Annual Symposium* (pp. 490-494). Philadelphia, Pennsylvania: Hanley & Belfus, Inc.

Diaper, D. (1993). Task observation for human-computer interaction. In D. Diaper (Ed.), *Task Analysis for Human-Computer Interaction* (pp. 210-237). New York, New York: John Wiley & Sons.

Dorsch, J.A. & Dorsch, S.E. (1999). *Understanding Anesthesia Equipment*. Baltimore, Maryland: Lippincott Williams & Wilkins.

Drews, F.A., Wachter, S.B., Agutter, J., Syroid, N., & Westenskow, D. (2004). Design and evaluation of a graphical pulmonary display for anesthesia. In *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting* (pp. 1648-1650). Santa Monica, California: Human Factors and Ergonomics Society.

Duke University Human Simulation and Patient Safety Center (2005). *The Human Simulation and Patient Safety Center*. Retrieved May 10, 2005 from http://simcenter.duhs.duke.edu/about.html.

Effken, J.A., Kim, N.G., & Shaw, R.E. (1997). Making the constraints visible: Testing the ecological approach to interface design. *Ergonomics, 40*(1), 1-27.

Endsley, M.R. (1993). A survey of situation awareness in air-to-air combat fighters. *The International Journal of Aviation Psychology, 3*(2), 157-168.

Endsley, M.R. & Rodgers, M.D. (1994). *Situation awareness information requirements for en route air traffic control*. (Tech. Report DOT/FAA/AM-94/27). Washington, DC: Office of Aviation Medicine, United States Department of Transportation, Federal Aviation Administration.

Evans, R.S., Pestotnik, S.L., Classen, D.C., Clemmer, T.P., Weaver, L.K., Orme, J.F., Lloyd, J.F., & Burke, J.P. (1998). A computer-assisted management program for antibiotics and other antiinfective agents. *The New England Journal of Medicine, 338*(4), 232-238.

Fu, L., Salvendy, G., & Turley, L. (2002). Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology, 21*(2), 137-143.

Gaba, D.M. (1994). Human error in dynamic medical domains. In M.S. Bogner (Ed.), *Human Error in Medicine* (pp. 197-224). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Gaba, D.M. (2000). Anaesthesiology as a model for patient safety in health care. *British Medical Journal, 320*, 785-788.

Gaba, D.M., Fish, K.J., & Howard, S.K. (1994). *Crisis Management in Anesthesiology*. New York, New York: Churchill Livingstone.

Gaba, D.M. & Howard, S.K. (1995). Situation awareness in anesthesiology. *Human Factors, 37*(1), 20-31.

Gordon, S.E. & Gill, R.T. (1997). Cognitive task analysis. In C.E. Zsambok and G. Klein (Eds.), *Naturalistic Decision Making* (pp. 131-140). Mahwah, New Jersey: Laurence Erlbaum Associates.

Hajdukiewicz, J.R., Vicente, K.J., Doyle, D.J., Milgram, P., & Burns, C.M. (2001). Modeling a medical environment: An ontology for integrated medical informatics design. *International Journal of Medical Informatics, 62*, 79-99.

Haug, P.J., Gardner, R.M., & Evans, R.S. (1999). Hospital-based decision support. In E.S. Berner (Ed.), *Clinical Decision Support Systems: Theory and Practice* (pp. 77-104). New York, New York: Springer.

Hoffman, R.R., Shadbolt, N.R., Burton, A.M., & Klein, G. (1995). Eliciting knowledge from experts: A methodological analysis. *Organizational Behavior and Human Decision Processes, 62*(2), 129-158.

Hollnagel, E. (2003). Prolegomenon to cognitive task design. In E. Hollnagel (Ed.), *Handbook of Cognitive Task Design* (pp. 3-16). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Huang, S.H. & Endsley, M.R. (1997). Providing understanding of the behavior of feedforward neural networks. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, 27*(3), 465-474.

Hunt, D.L., Haynes, R.B., Hanna, S.E., & Smith, K. (1998). Effects of computer-based clinical decision support systems on physician performance and patient outcomes: A systematic review. *Journal of the American Medical Association, 280*(15), 1339-1346.

Hutton, R.J.B., Miller, T.E., & Thordsen, M.L. (2003). Decision-centered design: Leveraging cognitive task analysis in design. In E. Hollnagel (Ed.), *Handbook of Cognitive Task Design* (pp. 383-416). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Jimison, H.B. & Sher, P.P. (1999). Decision support for patients. In E.S. Berner (Ed.), *Clinical Decision Support Systems: Theory and Practice* (pp. 139-166). New York, New York: Springer.

John, B.E. & Kieras, D.E. (1996a). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction, 3*(4), 320-351.

John, B.E. & Kieras, D.E. (1996b). Using GOMS for user interface design and evaluation: Which technique? *ACM Transactions on Computer-Human Interaction, 3*(4), 287-319.

Jungk, A., Thull, B., Hoeft, A., & Rau, G. (1999). Ergonomic evaluation of an ecological interface and a profilogram display for hemodynamic monitoring. *Journal of Clinical Monitoring and Computing, 15*, 469-479.

Jungk, A., Thull, B., Hoeft, A., & Rau, G. (2000). Evaluation of two new ecological interface approaches for the anesthesia workplace. *Journal of Clinical Monitoring and Computing, 16*, 243-258.

Kantner, L. & Rosenbaum, S. (1997). Usability studies of WWW sites: Heuristic evaluation vs. laboratory testing. In *Proceedings of the 15th Annual International Conference on Computer Documentation* (pp. 153-160). New York, New York: ACM Press.

Kantner, L., Shroyer, R., & Rosenbaum, S. (2002). Structured heuristic evaluation of online documentation. In *Proceedings of the IEEE International Professional Communication Conference* (pp. 331-342). Piscataway, New Jersey: IEEE.

Kieras, D. (1996). *A guide to GOMS model usability evaluation using NGOMSL*. University of Michigan, Ann Arbor, Michigan.

Kieras, D. (1997). Task analysis and the design of functionality. In A. Tucker (Ed.), *The Computer Science and Engineering Handbook* (pp. 1401-1423). Boca Raton, Florida: CRC Inc.

Kieras, D. (1999). *A guide to GOMS model usability evaluation using GOMSL and GLEAN3*. University of Michigan, Ann Arbor, Michigan.

Kieras D.E., Wood, S.D., Abotel, K., & Hornof, A. (1995). GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface designs. In *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology* (pp. 91-100). New York, New York: ACM.

Klein, G.A., Calderwood, R., & MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man, and Cybernetics, 19*(3), 462-472.

Knab, J.H., Wallace, M.S., Wagner, R.L., Tsoukatos, J., & Weinger, M.B. (2001). The use of a computer-based decision support system facilitates primary care physicians' management of chronic pain. *Anesthesia and Analgesia, 93*, 712-720.

Krol, M. & Reich, D.L. (1998). Expert systems in anesthesiology. *Drugs of Today, 34*(7), 593-601.

Krol, M. & Reich, D.L. (1999). The algorithm for detecting critical conditions during anesthesia. In *Proceedings of the 12th IEEE Symposium on Computer-Based Medical Systems* (pp. 208-213). Piscataway, New Jersey: IEEE.

Lai, F., Macmillan, J., Daudelin, D.H., & Kent, D.M. (2006). The potential of training to increase acceptance and use of computerized decision support systems for medical diagnosis. *Human Factors, 48*(1), 95-108.

Lin, L., Isla, R., Doniz, K., Harkness, H., Vicente, K.J., & Doyle, D.J. (1998). Applying human factors to the design of medical equipment: Patient-controlled analgesia. *Journal of Clinical Monitoring and Computing, 14*, 253-263.

Lincoln, M.J. (1999). Medical education applications. In E.S. Berner (Ed.), *Clinical Decision Support Systems: Theory and Practice* (pp. 105-138). New York, New York: Springer.

Linkens, D.A. & Vefghi, L. (1997). Recognition of patient anaesthetic levels: neural network systems, principal components analysis, and canonical discriminant variates. *Artificial Intelligence in Medicine, 11*, 155-173.

Loeb, R.G. (1993). A measure of intraoperative attention to monitor displays. *Anesthesia and Analgesia, 76*, 337-341.

Loeb, R.G. & Fitch, W.T. (2002). A laboratory evaluation of an auditory display designed to enhance intraoperative monitoring. *Anesthesia and Analgesia, 94*, 362-368.

Loeb, R.G., Jones, B.R., Leonard, R.A., & Behrman, K. (1992). Recognition accuracy of current operating room alarms. *Anesthesia and Analgesia, 75*, 499-505.

Lowe, A., Harrison, M.J., & Jones, R.W. (1999). Diagnostic monitoring in anaesthesia using fuzzy trend templates for matching temporal patterns. *Artificial Intelligence in Medicine, 16*, 183-199.

Ludbrook, G.L., Webb, R.K., Currie, M., & Watterson, L.M. (2005). Crisis management during anaesthesia: Myocardial ischaemia and infarction. *Quality and Safety in Healthcare, 14*(3), e13.

Mahfouf, M., Abbod, M.F., & Linkens, D.A. (2002). The design of supervisory rule-based control in the operating theatre via an anaesthesia simulator. *Expert Systems, 19*(1), 11-20.

Michels, P., Gravenstein, D., & Westenskow, D.R. (1997). An integrated graphic data display improves detection and identification of critical events during anesthesia. *Journal of Clinical Monitoring, 13*, 249-259.

Miller, R.A. & Geissbuhler, A. (1999). Clinical diagnostic decision support systems – An overview. In E.S. Berner (Ed.), *Clinical Decision Support Systems: Theory and Practice* (pp. 3-34). New York, New York: Springer.

Murphy, G.S. & Vender, J.S. (2001). Monitoring the anesthetized patient. In P.G. Barash, B.F. Cullen, and R.K. Stoelting (Eds.), *Clinical Anesthesia* (pp. 667-686). Philadelphia, Pennsylvania: Lippincott Williams & Wilkins.

Mylrea, K.C., Orr, J.A., & Westenskow, D.R. (1993). Integration of monitoring for intelligent alarms in anesthesia: Neural networks – can they help? *Journal of Clinical Monitoring, 9*, 31-37.

National Transportation Safety Board (2004). *Aviation Accident Statistics*. Retrieved April 27, 2005 from http://www.ntsb.gov/aviation/Table3.htm.

Nielsen, J. (1993). *Usability Engineering*. San Diego, California: Academic Press.

Olson, J.R. & Olson, G.M. (1990). The growth of cognitive modeling in human-computer interaction since GOMS. *Human-Computer Interaction, 5*, 221-265.

Rasmussen, J. (1983). Skills, rules, knowledge: Signals, signs, and symbols and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics, 13*, 257-267.

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man, and Cybernetics, 15*, 234-243.

Rasmussen, J. & Vicente, K.J. (1989). Coping with human error through system design: Implications for ecological interface design. *International Journal of Man-Machine Studies, 31*, 517-534.

Rennels, G.D. & Miller, P.L. (1988). Artificial Intelligence research in anesthesia and intensive care. *Journal of Clinical Monitoring, 4*, 274-289.

Roberts, S.L. & Tinker, J.H. (1996). Perioperative myocardial infarction. In N. Gravenstein and R.R. Kirby (Eds.), *Complications in Anesthesiology* (pp. 335-349). Philadelphia, Pennsylvania: Lippincott-Raven.

Schecke, T., Rau, G., Klocke, H., Kaesmacher, H., Hatzky, U., Kalff, G., & Zimmermann, H.-J. (1988). Knowledge-based decision support in anesthesia: A case study. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, 2* (pp. 962-965). Piscataway, New Jersey: IEEE.

Seagull, F.J. & Sanderson, P.M. (2004). The Trojan horse of the operating room: Alarms and the noise of anesthesia. In M.S. Bogner (Ed.), *Misadventures in Health Care: Inside Stories* (pp. 105-125). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Segall, N., Green, R.S., & Kaber, D.B. (2006). User, robot and automation evaluations in high-throughput biological screening processes. In *Proceedings of the 2006 ACM Conference on Human-robot Interaction* (pp. 274-281). New York: ACM Press.

Sharp, T.D. & Helmicki, A.J. (1998). The application of the ecological interface design approach to neonatal intensive care medicine. In *Proceedings of the 42nd Annual Meeting of the Human Factors and Ergonomics Society* (pp. 350-354). Santa Monica, California: Human Factors and Ergonomics Society.

Shepherd, A. (2001). *Hierarchical Task Analysis*. London: Taylor & Francis.

Sheridan, T.B. (1992). *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, Massachusetts: The MIT Press.

Sheridan, T.B. (2002). *Humans and Automation: System Design and Research Issues*. Indianapolis, Indiana: John Wiley & Sons, Inc.

Sheridan, T.B. & Thompson, J.M. (1994). People versus computers in medicine. In M.S. Bogner (Ed.), *Human Error in Medicine* (pp. 141-158). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Soar Technology (2005). *EGLEAN science and technology report: The first six months*. Ann Arbor, Michigan: Author.

Sowb, Y.A. & Loeb, R.G. (2002). Cognitive analysis of intraoperative critical events: A problem-driven approach to aiding clinicians' performance. *Cognition, Technology & Work, 4*, 107-119.

Spooner, S.A. (1999). Mathematical foundations of decision support systems. In E.S. Berner (Ed.), *Clinical Decision Support Systems: Theory and Practice* (pp. 35-60). New York, New York: Springer.

St. Amant, R. & Ritter, F.E. (2004). Specifying ACT-R models of user interaction with a GOMS language. *Cognitive Systems Research, 6*(1), 71-88.

Stanton, N. & Stevenage, S. (1998). Learning to predict human error: Issues of reliability, validity and acceptability. *Ergonomics, 41*(11), 1737-1756.

Stanton, N. & Young, M. (1998). Is utility in the eye of the beholder? A study of ergonomics methods. *Applied Ergonomics, 29*(1), 41-54.

Stedman, T.L. (2000). Myocardial infarction. *Stedman's Medical Dictionary Online.* Retrieved March 27, 2005 from Stedman's Medical Dictionary Online on the World Wide Web: http://www.stedmans.com/section.cfm/45.

Stoelting, R.K. & Dierdorf, S.F. (1993). *Anesthesia and Co-Existing Disease*. New York, New York: Churchill Livingstone.

Sutcliffe, A. & Gault, B. (2004). Heuristic evaluation of virtual reality applications. *Interacting with Computers, 16*(4), 831-849.

Syroid, N.D., Agutter, J., Drews, F.A., Westenskow, D.R., Albert, R.W., Bermudez, J.C., Strayer, D.L., Prenzel, H., Loeb, R.G., & Weinger, M.B. (2002). Development and evaluation of a graphical anesthesia drug display. *Anesthesiology, 96*, 565-574.

Uckun, S. (1994). Intelligent systems in patient monitoring and therapy management: A survey of research projects. *International Journal of Clinical Monitoring and Computing, 11*, 241-253.

Usher, J.M. and Kaber, D.B. (2000). Establishing information requirements for supervisory controllers in a flexible manufacturing system using GTA. *Human Factors and Ergonomics in Manufacturing*, *10*(4), 431-452.

Veterans Health Administration (2003). *Cognitive Aid for Anesthesiology*. Ann Arbor, Michigan: VA National Center for Patient Safety.

Vicente, K.J. (2002). Ecological interface design: Progress and challenges. *Human Factors, 44*(1), 62-78.

Vicente, K.J. (2003). Less is (sometimes) more in cognitive engineering: The role of automation technology in improving patient safety. *Quality and Safety in Health Care, 12*, 291-294.

Vicente, K.J. & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics, 22*(4), 589-606.

Virzi, R.A. (1997). Usability inspection methods. In M.G. Helander, T.K. Landauer, and P.V. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 705-715). Amsterdam: Elsevier.

Virzi, R.A., Sorce, J.F., & Herbert, L.B. (1993). Comparison of three usability evaluation methods: Heuristic, think aloud, and performance testing. In *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting, 1* (pp. 309-313). Santa Monica, California: Human Factors and Ergonomics Society.

Watt, R.C., Maslana, E.S., & Mylrea, K.C. (1993). Alarms and anesthesia: Challenges in the design of intelligent systems for patient monitoring. In *IEEE Engineering in Medicine and Biology, 12*(4) (34-41). Piscataway, New Jersey: IEEE.

Wei, J. & Salvendy, G. (2004). The cognitive task analysis methods for job and task design: Review and reappraisal. *Behaviour & Information Technology, 23*(4), 273-299.

Weinger, M.B. (1999). Anesthesia equipment and human error. *Journal of Clinical Monitoring and Computing, 15*, 319-323.

Weinger, M.B. & Slagle, J. (2002). Human factors research in anesthesia patient safety: Techniques to elucidate factors affecting clinical task performance and decision making. *Journal of the American Medical Informatics Association, 9*(6 Suppl 1), S58-S63.

Wixon, D. & Wilson, C. (1997). The usability engineering framework for product design and evaluation. In M.G. Helander, T.K. Landauer, and P.V. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 653-688). Amsterdam: Elsevier.

Wood, S.D. (2000). *Extending GOMS to human error and applying it to error-tolerant design*. Doctoral dissertation, University of Michigan, Department of Electrical Engineering and Computer Science.

Wright, M. C. (2004). *Information management in the perioperative environment*. K02 Independent Scientist Award proposal prepared for the National Institutes of Health. Durham, North Carolina: Duke Human Simulation and Patient Safety Center.

Xiao, Y., Milgram, P., & Doyle, D.J. (1997). Capturing and modeling planning expertise in anesthesiology: Results of a field study. In C.E. Zsambok and G. Klein (Eds.), *Naturalistic Decision Making* (pp. 197-205). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Zhang, J., Johnson, T.R., Patel, V.L., Paige, D.L., & Kubose, T. (2003). Using usability heuristics to evaluate patient safety of medical devices. *Journal of Biomedical Informatics, 36*, 23-30.

Zhang, Y., Drews, F.A., Westenskow, D.R., Foresti, S., Agutter, J., Bermudez, J.C., Blike, G., & Loeb, R. (2002). Effects of integrated graphical displays on situation awareness in anaesthesiology. *Cognition, Technology & Work, 4*, 82-90.

Appendices

Appendix A: Steps to Treating Myocardial Infarction

Gaba et al. (1994) recommend the following steps in treating myocardial infarction:

1. Verify manifestations of myocardial ischemia:
    - Assess clinical signs and symptoms.
    - Evaluate electrode placement and ECG settings.
    - Evaluate multiple ECG leads.
    - Obtain a 12-lead ECG as soon as possible and review previous ECGs.
    - Evaluate hemodynamic status.
2. Inform the operating surgeon:
    - Terminate surgery.
    - Request ICU bed for postoperative care.
3. Treat ventricular arrhythmias:
    - Administer lidocaine IV, 1.0-1.5 mg/kg bolus, then infusion of 1-4 mg/min.
    - Administer procainamide IV, 500 mg loading dose over 10-20 minutes, then infusion of 2-6 mg/min.
4. Place an arterial line to monitor blood pressure.
5. Treat tachycardia and/or hypertension:
    - Increase depth of anesthesia if appropriate.
    - Administer a β blockade:
        - Administer esmolol IV, 0.25-0.5 mg/kg bolus, 50-300 μg/kg/min infusion.
        - Administer labetolol IV, 5-10 mg bolus, repeat as necessary.
        - Administer propranolol IV, 0.25-1.0 mg bolus, repeat as necessary.
        - Administer drugs with caution if patient is hypotensive or has severe chronic obstructive pulmonary disease (COPD) or asthma.
    - Administer nitroglycerin (NTG) using one of the following methods:
        - Sublingual (absorption uncertain, can cause hypotension).
        - Transdermal paste, 1-2 in. applied to the chest wall (slow onset).
        - IV infusion, 0.25-2.0 μg/kg/min (titrated to effect).
    - Provide calcium channel blockade:
        - Sublingual administration of nifedipine, 5-10 mg (absorption uncertain, can cause hypotension).
        - Administer verapamil IV, 2.5 mg, repeat as necessary (avoid if β blockade present).
        - Administer diltiazem IV, 2.5 mg increments, repeat as necessary.
6. If hypotension develops:
    - Coronary perfusion pressure should take precedence over attempts at afterload reduction.
    - Maintain blood pressure with phenylephrine IV, 0.25-1.0 μg/kg/min by infusion.
    - Optimize circulating fluid volume. Use pulmonary artery (PA) pressures as a guide, consider placement of a PA catheter if not already in place.
    - Support myocardial contractility as needed using inotropic agents:

- Use inotropes with caution since they may increase myocardial O_2 demand and worsen ischemia.
- Administer dobutamine IV infusion, 5-10 μg/kg/min.
- Administer dopamine IV infusion, 5-10 μg/kg/min.
- Administer epinephrine IV infusion, 10-100 ng/kg/min.

- Avoid NTG or calcium channel blockade until hypotension or bradycardia are resolved. Consider combined use of phenylephrine and NTG infusions.

7. If cardiac arrest occurs, begin advanced cardiac life support (ACLS).
8. Conduct secondary management:
    - Ensure adequate oxygenation and ventilation – monitor with pulse oximeter and capnograph.
    - Treat pain and anxiety in the awake patient by titrating sedatives and narcotics.
    - Send blood samples to clinical laboratory for:
        - Arterial blood gases;
        - Hemoglobin/hematocrit;
        - Electrolytes; and
        - Creatine kinase (CK), CK-MB isoenzyme (for comparison with subsequent measurements).
    - Obtain cardiology consultation to determine postoperative management of patient:
        - Assessment for cardiac catheterization;
        - Circulatory support with a circulatory assist device (intra-aortic balloon pump);
        - Percutaneous transluminal coronary angioplasty or coronary artery bypass surgery; or
        - Thrombolytic therapy.

Appendix B: Expert Anesthesiologist Interview Questions

Below is Nielsen's (1993) proposed list of questions to be used as part of the "think aloud" methodology of task observational analysis. The questions are typically asked during an interview in which the interviewee describes how a human-computer interaction task is carried out:

- Why do you do this?
- How do you do it?
- Why do you not do this in such and such a manner?
- Do errors ever occur when doing this?
- How do you discover and correct these errors?
- Describe an exception from a normal work flow.
- Describe a notable success or failure in carrying out this task.
- Describe problems with this task.
- Describe what features of this task you like best and least.
- What changes would you like to make to this task?

In applying the "think aloud" methodology, the task under study is typically decomposed in a hierarchical fashion, starting with the analysis of higher-level tasks and working down to analysis of lower-level tasks.

Below is a straightforward adaptation of Nielsen's (1993) list of questions to the anesthesiology domain. These revised questions were asked during interviews in which anesthesiologists described how steps and procedures with OR systems are carried out during the handling of a crisis situation:

- Why do you do this?
- How do you do it?
- Why do you not do this in such and such a manner?
- Do unintended results ever occur when doing this?
- How do you discover and correct these results?
- Describe an exception from the routine treatment of MI.
- Describe a notable success or failure in treating MI.
- Describe problems with the MI treatment procedure.
- Describe what features of the existing tools and interfaces you like best and least.
- What changes would you like to make to these tools and interfaces?

Appendix C: Evaluation Packet

1. Subject Instructions

[Prepare forms, user manual and decision support tool. Fill out subject number in all forms.]

Thank you for participating in this study. As part of my dissertation work, I have developed a decision support tool for anesthesia crisis situations, specifically, myocardial infarction (MI). You will be asked to evaluate this tool. The evaluation will include an applicability assessment (to evaluate the usefulness of the tool) and a heuristic evaluation (to evaluate the usability of its interface). The total expected duration of the experiment is approximately one hour.

Before we start, please fill out these forms. **[Have subject complete general information page and sign informed consent forms. Sign informed consent forms.]** Thank you.

[Give subject copy of user manual.] The decision support tool is comprised of four windows. On the top left, it displays various patient physiological variables that are relevant to diagnosing and treating MI. During run time, these variables are updated approximately every 5 seconds. The top right window presents the most probable diagnosis, as well as how this diagnosis was derived. The large window on the bottom right presents recommended treatment steps for the problem diagnosed by the tool, as well as explanations for these recommendations. When these explanations relate to certain specific patient variables, the text and relevant variables are highlighted to emphasize this relationship. Finally, the window on the bottom left details ABCD treatment steps, which are tailored to the patient's current condition (as diagnosed by the tool). You can refer to the user manual, which describes the features of the decision support tool, during the evaluation process for more information. Do you have any questions about the tool?

I will now run the tool under two scenarios. In one, the patient will hemorrhage continuously and develop hypotension; in the other, the patient will experience MI. Each scenario will last approximately 10-12 minutes. The data for the patient variables was obtained from the patient simulator. This is the patient information. **[Give subject patient information sheet.]** During (or after) each scenario, you are asked to complete an applicability assessment survey. Following both scenarios, you are also asked to complete a heuristic evaluation of the decision support tool interface.

[Provide applicability assessment forms.] In the applicability assessment, you will be asked to rate your level of agreement or disagreement with several statements (such as "The diagnosis by the decision support tool was correct, based on the patient's physiological state"). Please provide comments for these statements. You will complete the applicability assessment survey twice, once for each scenario. In the second survey, there are additional statements addressing general issues related to the decision support tool interface design. Please review these surveys before we begin. Do you have any questions about any of the statements or the evaluation process?

[Demonstrate first scenario, either MI or hypotension, on the decision support tool.] This completes the first scenario. Do you have any questions about the decision support tool or scenario? If you haven't done so already, please complete the applicability assessment survey. **[Give subject time to complete form. Allow for a break.]**

[Demonstrate second scenario, either MI or hypotension, on the decision support tool.] This completes the second scenario. Do you have any questions about the decision support tool or scenario? If you haven't done so already, please complete the applicability assessment survey.

[Provide heuristic evaluation forms.] Usability is defined as "the effectiveness, efficiency, and satisfaction with which users can achieve specified goals in a particular environment". Interface usability has several components, such as providing prompt feedback and preventing users from making mistakes. In heuristic evaluation, subject matter experts are asked to inspect the interface in question and find where these components are lacking. Now you are asked to systematically examine the decision support tool interface and evaluate its compliance with the six usability principles listed in these forms (such as "Speak the users' language"). Write down interface issues that you think constitute violations of each principle. Please review these forms before we begin. Do you have any questions about any of the principles or how heuristic evaluation is carried out? Do you have any other questions?

Please fill out the heuristic evaluation forms. [Leave decision support tool on, to allow subject to perform heuristic evaluation. Give subject time to complete forms.]

Before we adjourn, I'd like to ask you a few questions:
What was your general impression of the tool? Would you recommend its use by junior anesthesia providers?
Do you have any suggestions for improving the tool, in terms of usefulness and usability?
[Write down responses in Notes sheet.]

Thank you again for agreeing to take part in this evaluation. Do you have any questions about this study?

2. Patient Information

Name, Age, and Gender:

Stan D. Ardman ("Standard Man", "Stan"), 33-year old, male

History of Present Illness:

Otherwise healthy adult with compound ankle fracture requiring ORIF.

Past Medical History:

None
NKDA
Denies tobacco, alcohol, and IV drug use
Runs 2 miles several times a week

Past Surgical/Anesthetic History:

Tonsillectomy at age 6, general anesthesia without complications
No family history of anesthetic problems

Review of Systems:

CNS:	Negative for stroke
Cardiovascular:	Negative for hypertension, angina, DOE
Pulmonary:	Negative for COPD, asthma, recent URI
Renal/Hepatic:	Negative for renal failure, jaundice
Endocrine:	Negative for diabetes, thyroid disease
Heme/Coag:	Negative for anemia, bruising

Current Medications:

None

Physical Examination:

General:	Healthy adult male, average build, in no distress
Weight, Height:	70 kg, 6'0"
Vital Signs:	HR 73 bpm, BP 113/52 mmHg, RR 13 br/min, SpO2 97%
Airway:	Full dentition, no loose teeth FROM neck & TMJ, wide oral opening, 4 fb mandible, MC 1
Lungs:	Relaxed respiration, with clear bilateral breath sounds
Heart:	RRR. Normal S1, S2; no S3, S4, murmur, or rub

Laboratory, Radiology, and other relevant studies:

HCT:	42.3%

Narrative:

A healthy adult male who runs two miles several times a week suffers a compound ankle fracture and requires ORIF. Patient has no systemic illness or other health problems. He received general anesthesia uneventfully as a child and there is no family history of anesthesia problems. Physical examination reveals no anesthetic concerns. Patient refuses regional anesthesia and requests general anesthesia.

3. General Questions

Gender (circle one): Male Female

Age: _________

Current position/title: **____**

Years of clinical practice: _________

I have treated a patient for perioperative myocardial infarction (circle one): Yes No

Do not write below this line. Experimenter use only.

Subject _________

4. Survey of Applicability of Decision Support Tool – Scenario 1

Please indicate your level of agreement or disagreement with the following statements. Where appropriate, please provide comments.

1. The physiological variables displayed on the screen represented deviations that should be attended to, and were not "false alarms".

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

____

2. The diagnosis was correct, based on the patient's physiological state.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

____

Do not write below this line. Experimenter use only.

Subject _________ Scenario _________

3. Alternative diagnoses were possible that were not suggested by the tool.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

4. The treatment steps were clear.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

5. There were unnecessary treatment steps.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

6. Alternative treatment steps were possible that were not suggested by the tool.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

7. The explanations provided by the tool supporting its suggestions were useful.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

5. Survey of Applicability of Decision Support Tool – Scenario 2

Please indicate your level of agreement or disagreement with the following statements. Where appropriate, please provide comments.

1. The physiological variables displayed on the screen represented deviations that should be attended to, and were not "false alarms".

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

2. The diagnosis was correct, based on the patient's physiological state.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Do not write below this line. Experimenter use only.

Subject _________ Scenario _________

3. Alternative diagnoses were possible that were not suggested by the tool.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

4. The treatment steps were clear.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

5. There were unnecessary treatment steps.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

6. Alternative treatment steps were possible that were not suggested by the tool.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

7. The explanations provided by the tool supporting its suggestions were useful.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

8. Some features of the interface were not clearly understandable.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Please list these features and how they were unclear.

9. Essential features could be added to this interface.

Strongly Agree		Agree		Neutral		Disagree		Strongly Disagree

Please list and describe these features.

10. Features could be removed from this interface.

Strongly Agree		Agree		Neutral		Disagree		Strongly Disagree

Please list these features and describe why they are unnecessary.

11. I found the tool to be useful.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

12. I would use this tool during a crisis situation.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

If not, please explain why.

6. Heuristic Analysis Evaluation Form

Please evaluate the decision support tool interface using the following heuristics. Write down issues that constitute violations of each heuristic.

1. *Simple and natural dialog.* The interface should not contain irrelevant or rarely needed information. All information should appear in a logical order.

2. *Speak the users' language.* Concepts and terminology should be taken from the anesthesiology domain.

3. *Minimize users' memory load.* Users should not have to remember information from one screen to another.

4. *Consistency*. Users should not have to wonder whether different words, situations, or actions mean the same thing.

____

____

____

____

____

5. *Feedback*. The system should keep the users informed about what is going on, within reasonable time.

____

____

____

____

____

6. *Prevent errors*. The design of the system should prevent errors from occurring.

____

____

____

____

____

Do not write below this line. Experimenter use only.

Subject _________

Appendix D: Hierarchical Task Analysis for Treating Myocardial Infarction

	Task	Plan
	Manage perioperative myocardial ischemia	do in sequence 1-4; If not an emergency (5) ; If (MAP drops > 20% from baseline for a patient with CAD or > 40% or to < 40 for a healthy patient for > 10 sec) and (if HR is > 40 above baseline or > 100 for a patient with CAD or >120 for a healthy patient for over 15 sec) (6) ; If hypotension resolved (MAP at baseline or stable) (7) ; If ischemia does not resolve rapidly (do in sequence 8-10) ; do in sequence 11-12
1	Verify manifestations of myocardial ischemia	do in sequence 1-3; If HR and BP are stable and no ventricular arrhythmias are present (except for ST segment shifts) (4) ; If surgery hasn't started (5)
1.1	Assess clinical signs and symptoms	
1.1.1	Talk to patient if patient is under regional anesthesia or MAC	
1.1.2	Look at ECG	
1.1.2.1	Look for ST segment elevation or depression	
1.1.2.2	Look for T wave flattening or inversion	
1.1.2.3	Look for ventricular arrhythmias	
1.2	Evaluate correctness of ECG readings	
1.2.1	Evaluate electrode placement	
1.2.2	Evaluate ECG settings	
1.2.3	Evaluate multiple ECG leads	
1.3	Evaluate hemodynamic status	
1.4	Evaluate baseline ECG	
1.5	Obtain a 12-lead ECG as soon as possible	
2	Consider precipitating factors	
2.1	Evaluate whether pre-existing cardiovascular disease exists	
2.2	Evaluate whether patient is hemodynamically stable	

2.3	Evaluate whether patient is desaturated	
2.4	Evaluate existence of pulmonary edema	
2.5	Evaluate whether patient is experiencing awareness/light anesthesia/intubation problems and treat	
3	Increase oxygenation to 100%	
4	Communicate with operating surgeon	
4.1	Inform surgeon of problem	
4.2	Evaluate whether surgeon actions may be cause of ischemia	
5	Complete ABCD - SWIFT CHECK	If O2 saturation < 92 and (ETCO2 < 28 or ETCO2 drops to half of baseline value in < 2 min) [1; 2; 3; optionally do any 4-5] - Otherwise [do in sequence 1-5]
5.1	Evaluate airway	If patient state is not severe (1) ; If patient state is severe (2) ; If patient state is critical (3)
5.1.1	Check patient status	1; If suspicious of airway obstruction (2)
5.1.1.1	Observe, palpate and auscultate neck	
5.1.1.2	Plan direct laryngoscopy	
5.1.2	Clear airway	do in sequence 1-2; If suspicious (3)
5.1.2.1	Adjust head and neck, attempt gentle chin lift	
5.1.2.2	Prepare for laryngoscopy	
5.1.2.3	Manage airway obstruction	
5.1.3	Prepare for emergency	optionally do any 1-3; 4
5.1.3.1	Manage laryngospasm	
5.1.3.2	Manage airway obstruction	
5.1.3.3	Manage aspiration problems	
5.1.3.4	Check intubation	
5.2	Evaluate breathing	If patient state is not severe (1) ; If patient state is severe (2) ; If patient state is critical (3)
5.2.1	Check patient status	1; If a capnograph is in use (2)
5.2.1.1	Palpate and auscultate chest	
5.2.1.2	Review ETCO2	
5.2.2	Examine patient chest	
5.2.2.1	Expose chest and abdomen	
5.2.2.2	Compare left and right sides	

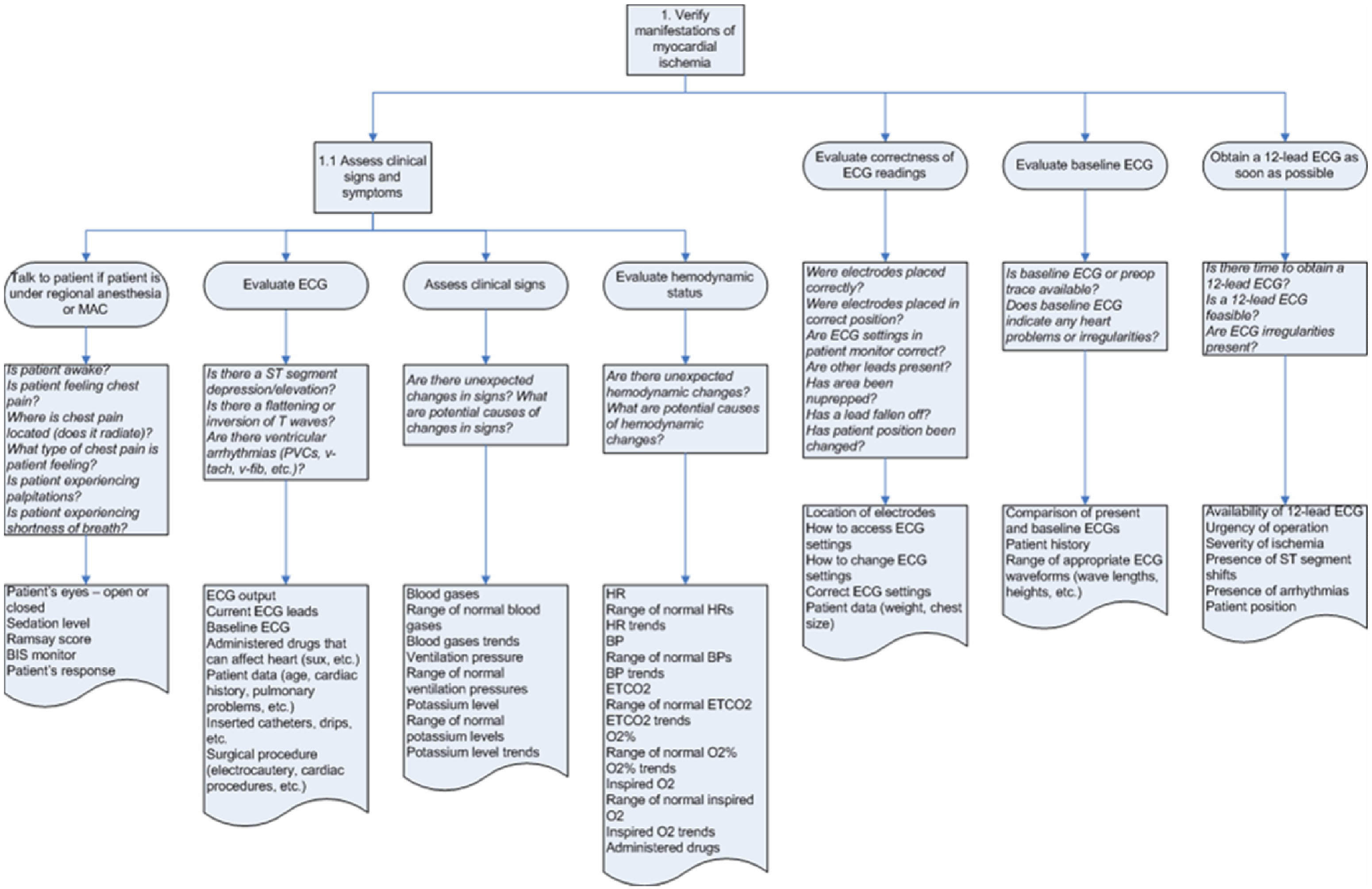
5.2.2.3	Look for causes of breathing problems	
5.2.3	Prepare for emergency	optionally do any 1-4; If patient is dry (urine output < 0.5 cc/kg/hr or > 50% drop in CVP or > 50% drop in PA catheter wedge pressure or drop in cardiac output/index to < 2) (5)
5.2.3.1	Manage bronchospasm	
5.2.3.2	Manage pulmonary edema	
5.2.3.3	Manage acute respiratory distress syndrome	
5.2.3.4	Manage ventilation problems	
5.2.3.5	Administer IV fluids	
5.3	Evaluate circulation	If patient state is not severe (1) ; If patient state is severe (2) ; If patient state is critical (3)
5.3.1	Check correctness of blood pressure readings	1; If arterial line is in place (do in sequence 2-5)
5.3.1.1	Cycle cuff and run cuff again	
5.3.1.2	Flush and zero arterial line	
5.3.1.3	Check height of transducer	
5.3.1.4	Check catheter tubing	
5.3.1.5	Make sure suppressed auto gain is off	
5.3.2	Evaluate access	1; optionally do any 2-3
5.3.2.1	Check IV access	
5.3.2.2	Secure additional access (venous and arterial)	
5.3.2.3	Prepare to transfuse	
5.3.3	Prepare for emergency	optionally do any 1-4
5.3.3.1	Treat hypotension (step 6)	
5.3.3.2	Treat tachycardia (step 6)	
5.3.3.3	Treat bradycardia	
5.3.3.4	Treat hypertension	
5.4	Evaluate drugs	If patient state is not severe (1) ; If patient state is severe (2) ; If patient state is critical (3)
5.4.1	Check equipment	
5.4.1.1	Check ampoules	
5.4.1.2	Check syringes	
5.4.1.3	Check labels	
5.4.1.4	Check infusion apparatus	
5.4.1.5	Check connections from fluid source to vein	
5.4.1.6	Check cannulae from fluid source to vein	
5.4.2	Check critical equipment and drugs	

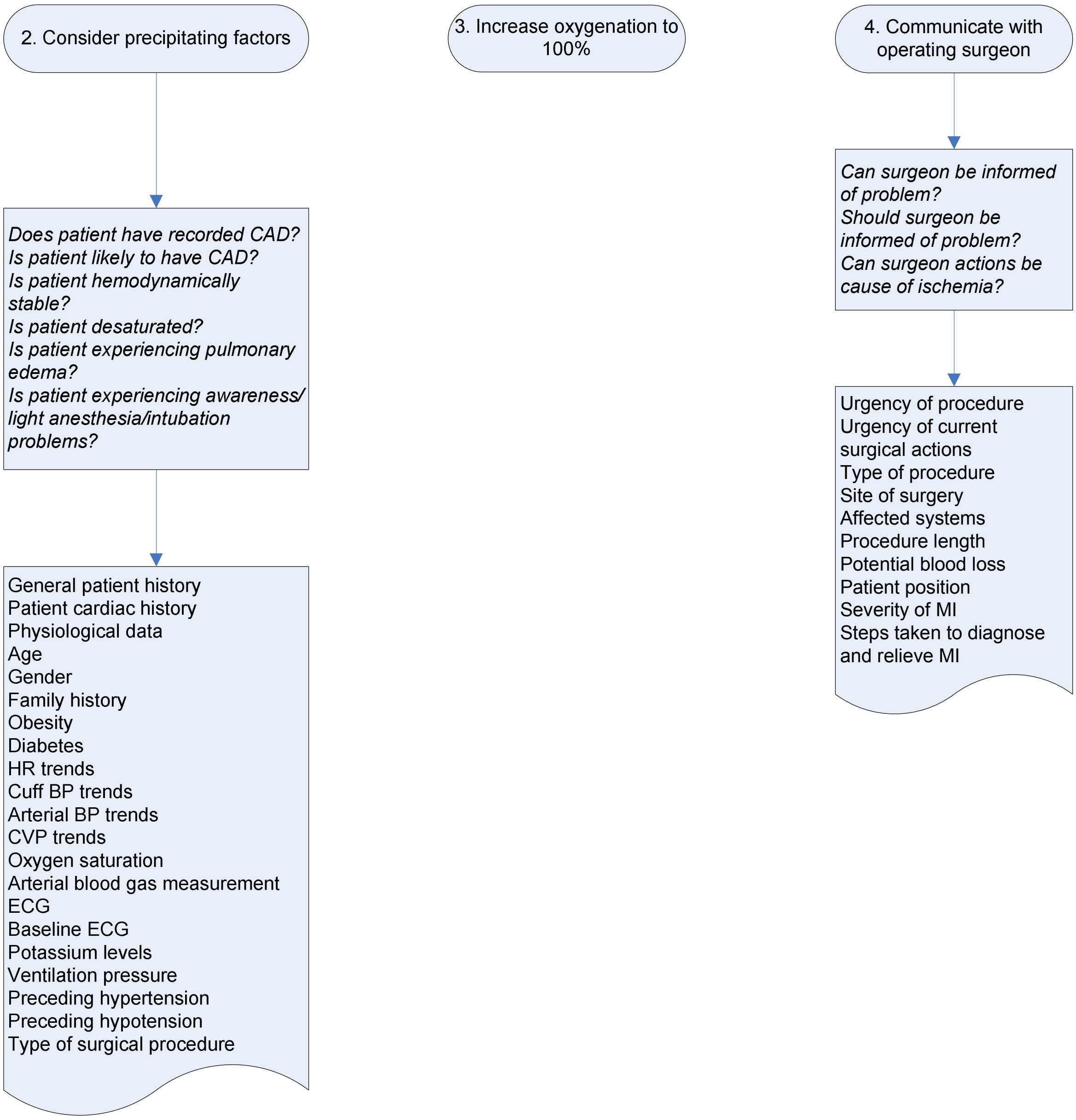
5.4.2.1	Allocate "Drugs" task	
5.4.2.2	Check drugs	
5.4.2.3	Check infusions	
5.4.2.4	Check entire IV apparatus	
5.4.2.5	Draw up, check and label drugs that may be needed	
5.4.3	Prepare for emergency	
5.4.3.1	Check for errors	
5.4.3.2	Ensure all drugs are labeled	
5.4.3.3	Keep record of drug doses and administration times	
5.5	Complete SWIFT CHECK	
5.5.1	Correlate monitored parameters with clinical situation and risk factors	
5.5.2	Check pre-operative assessment	
5.5.3	Check medical record	
5.5.4	Check ward drug chart	
6	Treat hypotension and tachycardia	1; If systolic BP is 30-60 (2) ; If systolic BP < 40 or MAP < 30 or v-tach, v-fib, pulseless v-tach, atrial fibrillation, or supraventricular tachycardia is present (3) ; 4; If patient is dry (urine output < 0.5 cc/kg/hr or > 50% drop in CVP or > 50% drop in PA catheter wedge pressure or drop in cardiac output/index to < 2) (5) ; If patient is wet (> 50% increase in CVP or PA catheter wedge pressure > 20 or cardiac output/index > 3) (6) ; 7; If patient is experiencing anaphylaxis (erythema, rash or wheeze is evident) or (HR < 130 and systolic BP < 40 or MAP < 50) or (cardiac arrest is imminent and rapid drop in BP) (8) ; 9; If patient is stable or help is available (10) ; 11
6.1	Confirm blood pressure change is real	do in sequence 1-2; If arterial line is in place (do in sequence 3-6)
6.1.1	Cycle cuff and run cuff again	
6.1.2	Check O2 saturation	
6.1.3	Flush and zero arterial line	
6.1.4	Check height of transducer	
6.1.5	Check catheter tubing	
6.1.6	Make sure suppressed auto gain is off	
6.2	Prepare for ACLS	
6.3	Treat as cardiac arrest - go through ACLS	
6.4	Recheck vaporizers are off	

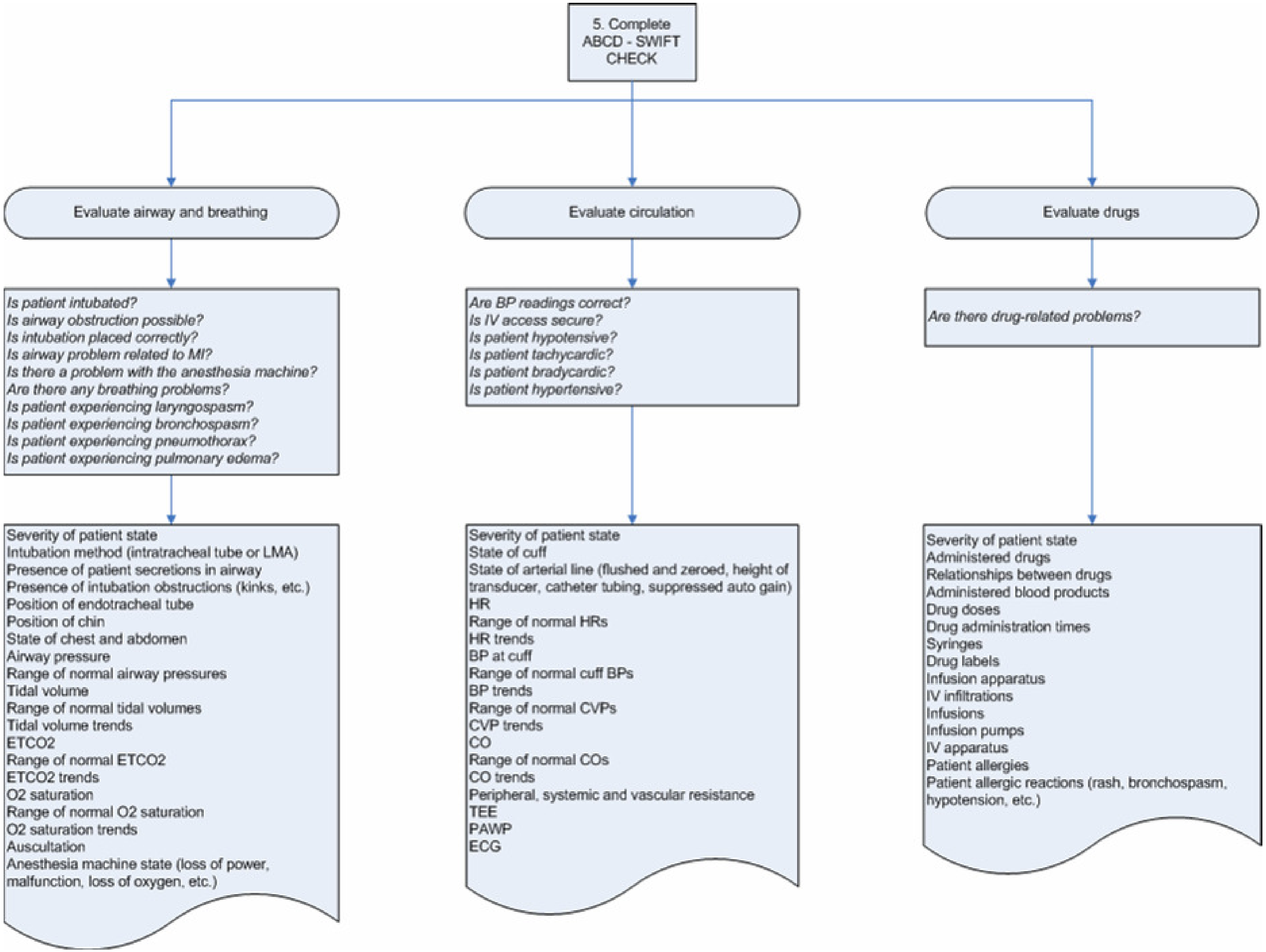
6.5	Administer IV fluids	
6.6	Administer diuretic	
6.7	Give vasopressor	
6.8	Consider epinephrine	
6.9	Improve patient posture	
6.9.1	Put in Trendelburg position	
6.9.2	Elevate legs	
6.10	Increase monitoring	
6.10.1	Increase ECG monitoring	
6.10.2	Insert arterial line	
6.10.3	Insert CVP line	
6.10.4	Insert PA catheter	
6.11	Treat probable causes	If patient is hypovolemic (consider blood loss, dehydration, diuresis, sepsis) (do in sequence 1-4) ; If a drug problem is suspected (consider induction and inhalation agents, atropine, local anesthetic toxicity, adrenaline, cocaine, vasopressor/vasodilator infusion problem, opioids, suxamethonium, anticholinesterases, vancomycin, protamine, surgeon drugs) (do in sequence 5-6) ; If a regional anesthesia problem is suspected (Consider vasodilation, bradycardia, respiratory failure) (do in sequence 7-9) ; If a surgical event is suspected (Consider vagal reflexes, obstructed venous return, pneumoperitonium, retractors and position) (10) ; If airway problem is suspected (Consider laryngoscopy, central venous catheter insertion, surgical manipulation, awareness) (11) ; If cardiopulmonary problem is suspected (Consider tension pneumothorax, hemothorax, tamponade, embolism (gas, amniotic or thrombus), sepsis, myocardial irritability (from drugs, ischemia, electrolytes, trauma), pulmonary edema, anaphylaxis) (12)
6.11.1	Ensure adequate IV access	
6.11.2	Ensure fluid replacement	
6.11.3	Cross match blood	
6.11.4	Check hematocrit	
6.11.5	Ensure agent ceased	
6.11.6	Support circulation	
6.11.7	Ensure volume loading	
6.11.8	Ensure airway support	
6.11.9	Consider left lateral displacement during pregnancy	

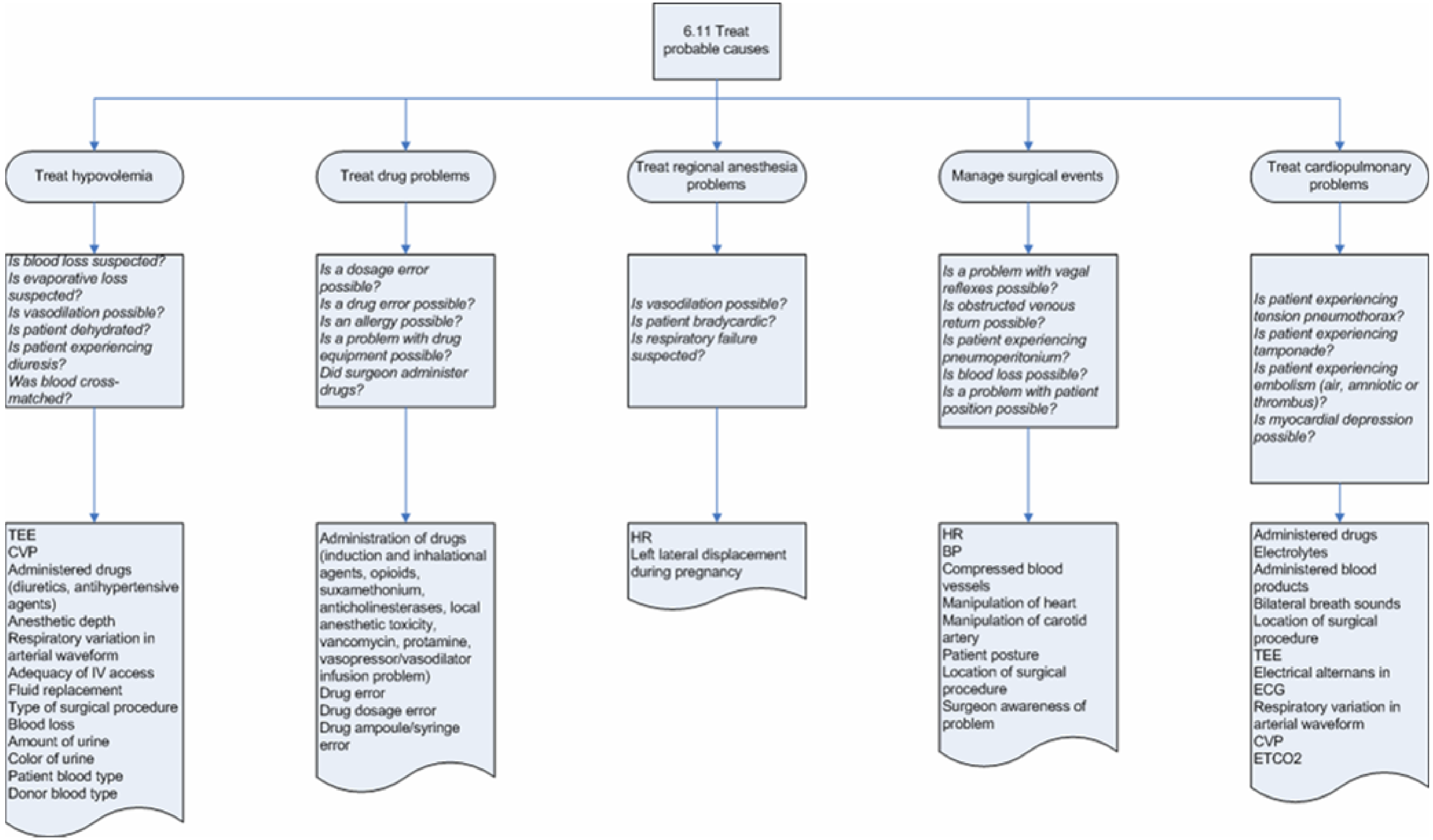
6.11.10	Ensure surgeon aware
6.11.11	Treat airway problems
6.11.12	Treat cardiopulmonary problems
7	Titrate nitroglycerin against clinical response
8	Consider multilead ECG monitoring
9	Monitor ECG continuously
10	Consider beta blocker to cover emergence
11	Request ICU bed for postoperative care

Appendix E: Goal-directed Task Analysis for Treating Myocardial Infarction

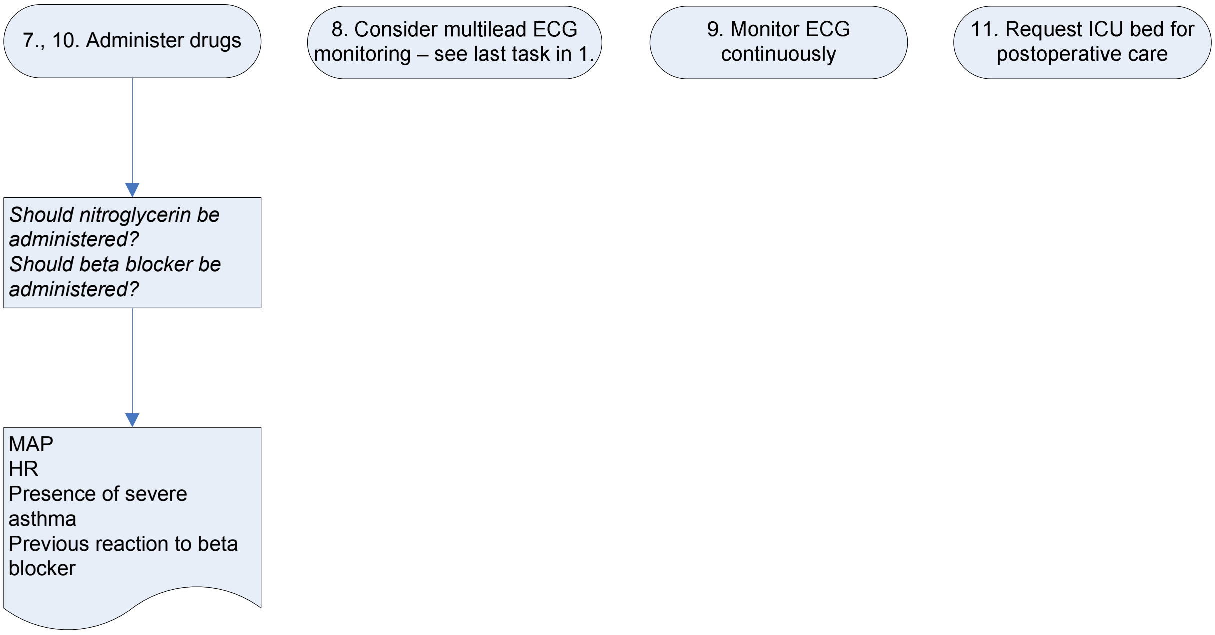



2. Consider precipitating factors
Does patient have recorded CAD?
Is patient likely to have CAD?
Is patient hemodynamically stable?
Is patient desaturated?
Is patient experiencing pulmonary edema?
Is patient experiencing awareness/ light anesthesia/intubation problems?
General patient history
Patient cardiac history
Physiological data
Age
Gender
Family history
Obesity
Diabetes
HR trends
Cuff BP trends
Arterial BP trends
CVP trends
Oxygen saturation
Arterial blood gas measurement
ECG
Baseline ECG
Potassium levels
Ventilation pressure
Preceding hypertension
Preceding hypotension
Type of surgical procedure
3. Increase oxygenation to 100%
4. Communicate with operating surgeon
Can surgeon be informed of problem?
Should surgeon be informed of problem?
Can surgeon actions be cause of ischemia?
Urgency of procedure
Urgency of current surgical actions
Type of procedure
Site of surgery
Affected systems
Procedure length
Potential blood loss
Patient position
Severity of MI
Steps taken to diagnose and relieve MI


5. Complete ABCD - SWIFT CHECK
Evaluate airway and breathing
Evaluate circulation
Evaluate drugs
Is patient intubated?
Is airway obstruction possible?
Is intubation placed correctly?
Is airway problem related to MI?
Is there a problem with the anesthesia machine?
Are there any breathing problems?
Is patient experiencing laryngospasm?
Is patient experiencing bronchospasm?
Is patient experiencing pneumothorax?
Is patient experiencing pulmonary edema?
Are BP readings correct?
Is IV access secure?
Is patient hypotensive?
Is patient tachycardic?
Is patient bradycardic?
Is patient hypertensive?
Are there drug-related problems?
Severity of patient state
Intubation method (intratracheal tube or LMA)
Presence of patient secretions in airway
Presence of intubation obstructions (kinks, etc.)
Position of endotracheal tube
Position of chin
State of chest and abdomen
Airway pressure
Range of normal airway pressures
Tidal volume
Range of normal tidal volumes
Tidal volume trends
ETCO2
Range of normal ETCO2
ETCO2 trends
O2 saturation
Range of normal O2 saturation
O2 saturation trends
Auscultation
Anesthesia machine state (loss of power, malfunction, loss of oxygen, etc.)
Severity of patient state
State of cuff
State of arterial line (flushed and zeroed, height of transducer, catheter tubing, suppressed auto gain)
HR
Range of normal HRs
HR trends
BP at cuff
Range of normal cuff BPs
BP trends
Range of normal CVPs
CVP trends
CO
Range of normal COs
CO trends
Peripheral, systemic and vascular resistance
TEE
PAWP
ECG
Severity of patient state
Administered drugs
Relationships between drugs
Administered blood products
Drug doses
Drug administration times
Syringes
Drug labels
Infusion apparatus
IV infiltrations
Infusions
Infusion pumps
IV apparatus
Patient allergies
Patient allergic reactions (rash, bronchospasm, hypotension, etc.)



6.11 Treat probable causes
Treat hypovolemia
Treat drug problems
Treat regional anesthesia problems
Manage surgical events
Treat cardiopulmonary problems
Is blood loss suspected?
Is evaporative loss suspected?
Is vasodilation possible?
Is patient dehydrated?
Is patient experiencing diuresis?
Was blood cross-matched?
Is a dosage error possible?
Is a drug error possible?
Is an allergy possible?
Is a problem with drug equipment possible?
Did surgeon administer drugs?
Is vasodilation possible?
Is patient bradycardic?
Is respiratory failure suspected?
Is a problem with vagal reflexes possible?
Is obstructed venous return possible?
Is patient experiencing pneumoperitonium?
Is blood loss possible?
Is a problem with patient position possible?
Is patient experiencing tension pneumothorax?
Is patient experiencing tamponade?
Is patient experiencing embolism (air, amniotic or thrombus)?
Is myocardial depression possible?
TEE
CVP
Administered drugs (diuretics, antihypertensive agents)
Anesthetic depth
Respiratory variation in arterial waveform
Adequacy of IV access
Fluid replacement
Type of surgical procedure
Blood loss
Amount of urine
Color of urine
Patient blood type
Donor blood type
Administration of drugs (induction and inhalational agents, opioids, suxamethonium, anticholinesterases, local anesthetic toxicity, vancomycin, protamine, vasopressor/vasodilator infusion problem)
Drug error
Drug dosage error
Drug ampoule/syringe error
HR
Left lateral displacement during pregnancy
HR
BP
Compressed blood vessels
Manipulation of heart
Manipulation of carotid artery
Patient posture
Location of surgical procedure
Surgeon awareness of problem
Administered drugs
Electrolytes
Administered blood products
Bilateral breath sounds
Location of surgical procedure
TEE
Electrical alternans in ECG
Respiratory variation in arterial waveform
CVP
ETCO2



Note: Numbers in tasks or subgoals correspond to tasks in the HTA (Appendix D).

Appendix F: GOMSL Code for Anesthesia Decision Support Tool

Note: Comments (denoted by //) represent text displayed in the ADST. Numbers in comments correspond to tasks in the HTA (Appendix D).

```
Define_model: "Manage MI"
Starting_goal is Begin diagnosis.

Method_for_goal: Begin diagnosis
    Step 1. Type_in "c".   //clear decision support tool window
    Step 2. Look_for_object_whose Label is hidden and_store_under <state>.  //diagnosis
    Step 3. Accomplish_goal: Select state using Value of <state>.
    Step 4. Delete <state>.
    Step 5. Goto 1.
    Step 6. Return_with_goal_accomplished.

Selection_rules_for_goal: Select state using <patient_state>
    If <patient_state> is "Hypo", Then Accomplish_goal: Hypotension state.
    If <patient_state> is "Normal", Then Accomplish_goal: Normal state.
    If <patient_state> is_not "Hypo", and <patient_state> is_not "Normal",
        Then Accomplish_goal: MI state.
    Return_with_goal_accomplished.

Method_for_goal: Determine state
    Step 1. Look_for_object_whose Label is hidden and_store_under <state>.
    Step 2. Decide: If Value of <state> is "Normal", and <Exception_name> is_not
        "Normal_exception", Then Abort_and_restart.
    Step 3. Decide: If Value of <state> is "Hypo", and <Exception_name> is_not
        "Hypo_exception", Then Abort_and_restart.
    Step 4. Decide: If Value of <state> is "MI", and <Exception_name> is_not
        "MI_exception", Then Abort_and_restart.
    Step 5. Decide: If Value of <state> is "MI Hypo", and <Exception_name> is_not
        "MI_exception", Then Abort_and_restart.
    Step 6. Decide: If Value of <state> is "MI Tachy", and <Exception_name> is_not
        "MI_exception", Then Abort_and_restart.
    Step 7. Decide: If Value of <state> is "MI Tachy Hypo", and <Exception_name> is_not
            "MI_exception", Then Abort_and_restart.
    Step 8. Delete <state>.
    Step 9. Return_with_goal_accomplished.

Method_for_goal: Normal state
    On_error: Determine state
    Step 1. Raise "Normal_exception".
    Step 2. Return_with_goal_accomplished.

Method_for_goal: Hypotension state
```

On_error: Determine state
Step 1. Accomplish_goal: Complete ABCD_SWIFT_CHECK.
Step 2. Raise "Hypo_exception".
Step 3. Accomplish_goal: Treat hypotension.
Step 4. Raise "Hypo_exception".
Step 5. Return_with_goal_accomplished.

Method_for_goal: MI state
On_error: Determine state
Step 1. Accomplish_goal: Confirm myocardial_ischemia_manifestations.
Step 2. Type_in "16". //2. Consider precipitating factors:
//Evaluate whether pre-existing cardiovascular disease exists
//Evaluate whether patient is hemodynamically stable
//Evaluate whether patient is desaturated
//Evaluate existence of pulmonary edema
//Evaluate whether patient is experiencing awareness/light
//anesthesia/intubation problems and treat
Step 3. Type_in "17". //3. Increase oxygenation to 100%
Step 4. Type_in "18". //4. Communicate with operating surgeon:
//Inform surgeon of problem
//Evaluate whether surgeon actions may be cause of ischemia
Step 5. Raise "MI_exception".
Step 6. Accomplish_goal: Complete ABCD_SWIFT_CHECK.
Step 7. Accomplish_goal: Treat hypotension_and_tachycardia.
Step 8. Type_in "38". //7. Titrate nitroglycerin against clinical response
Step 9. Type_in "39". //If ischemia does not resolve rapidly, do 8-9:
//8. Consider multilead ECG monitoring
//9. Monitor ECG continuously
Step 10. Type_in "40". //10.Consider beta blocker to cover emergence
Step 11. Type_in "41". //11.Request ICU bed for postoperative care
Step 12. Raise "MI_exception".
Step 13. Return_with_goal_accomplished.

//1
Method_for_goal: Confirm myocardial_ischemia_manifestations
On_error: Determine state
Step 1. Type_in "1". //Verify manifestations of myocardial ischemia:
Step 2. Type_in "2". //Talk to patient if patient is under regional anesthesia or MAC
Step 3. Accomplish_goal: Look at_ECG.
Step 4. Raise "MI_exception".
Step 5. Type_in "12". //Evaluate correctness of ECG readings:
//Evaluate electrode placement
//Evaluate ECG settings
//Evaluate multiple ECG leads
Step 6. Type_in "13". //Evaluate hemodynamic status
Step 7. Accomplish_goal: Evaluate baseline_ECG.

Step 8. Raise "MI_exception".
Step 9. Type_in "15". //If surgery hasn't started, obtain a 12-lead ECG as soon as possible
Step 10. Return_with_goal_accomplished.

//1.1.2
Method_for_goal: Look at_ECG
Step 1. Type_in "3". //Look for irregularities in ECG:
Step 2. Look_for_object_whose Label is ST_segment and_store_under <ST_segment>.
Step 3. Decide: If Value of <ST_segment> is "true", Then Type_in "4".
//ST segment changes present
Step 4. Look_for_object_whose Label is ECG and_store_under <ECG>.
Step 5. Decide: If Value of <ECG> is "10% PVCs", Then Type_in "5".
//10% PVCs present
Step 6. Decide: If Value of <ECG> is "25% PVCs", Then Type_in "6".
//25% PVCs present
Step 7. Decide: If Value of <ECG> is "V tach", Then Type_in "7".
//Ventricular tachycardia present
Step 8. Decide: If Value of <ECG> is "Pulseless v tach", Then Type_in "8".
//Pulseless ventricular tachycardia present
Step 9. Decide: If Value of <ECG> is "V fib", Then Type_in "9".
//Ventricular fibrillation present
Step 10. Decide: If Value of <ECG> is "Sinus tach", Then Type_in "10".
//Sinus tachycardia present
Step 11. Decide: If Value of <ECG> is "Asystole", Then Type_in "11".
//Asystole present
Step 12. Delete <ST_segment>; Delete <ECG>.
Step 13. Return_with_goal_accomplished.

//1.4
Method_for_goal: Evaluate baseline_ECG
Step 1. Look_for_object_whose Label is ECG and_store_under <ECG>.
Step 2. Decide: If Value of <ECG> is "none", Then Type_in "14".
//Evaluate baseline ECG if available
Step 3. Delete <ECG>.
Step 4. Return_with_goal_accomplished.

//5
Method_for_goal: Complete ABCD_SWIFT_CHECK
Step 1. Type_in "19". //5. Complete ABCD - SWIFT CHECK:
Step 2. Look_for_object_whose Label is hidden and_store_under <state>.
Step 3. Decide: If Value of <state> is "MI", Then Type_in "22". //Critical
Step 4. Decide: If Value of <state> is "MI Hypo", Then Type_in "22". //Critical
Step 5. Decide: If Value of <state> is "MI Tachy", Then Type_in "22". //Critical
Step 6. Decide: If Value of <state> is "MI Tachy Hypo", Then Type_in "22". //Critical
Step 7. Decide: If Value of <state> is "Hypo", Then Type_in "21". //Severe
Step 8. Decide: If Value of <state> is "Normal", Then Type_in "20". //Not severe

```
    Step 9. Delete <state>.
    Step 10. Return_with_goal_accomplished.

//6
Method_for_goal: Treat hypotension_and_tachycardia
    On_error: Determine state
    Step 1. Look_for_object_whose Label is hidden and_store_under <state>.
    Step 2. Decide: If Value of <state> is "MI", Then Return_with_goal_accomplished.
    Step 3. Type_in "23".     //6. Treat hypotension and/or tachycardia:
                              //Confirm blood pressure change is real
                              //Cycle cuff and run cuff again
                              //Check O2 saturation
                              //If arterial line is in place
                              //Flush and zero arterial line
                              //Check height of transducer
                              //Check catheter tubing
                              //Make sure suppressed auto gain is off
    Step 4. Accomplish_goal: Prepare for_ACLS.
    Step 5. Type_in "28".     //Recheck vaporizers are off
    Step 6. Raise "MI_exception".
    Step 7. Accomplish_goal: Manage hydration.
    Step 8. Raise "MI_exception".
    Step 9. Type_in "31".     //Give vasopressor
    Step 10. Type_in "32".    //If patient is experiencing anaphylaxis (erythema, rash or
                              //wheeze is evident) or (HR < 130 and systolic BP < 40 or
                              //MAP < 50) or (cardiac arrest is imminent and rapid drop in
                              //BP), consider epinephrine
    Step 11. Type_in "33".    //Improve patient posture:
                              //Put in Trendelenburg position
                              //Elevate legs
    Step 12. Type_in "34".    //If patient is stable or help is available, increase monitoring:
                              //Increase ECG monitoring
                              //Insert arterial line
    Step 13. Look_for_object_whose Label is CVP and_store_under <CVP>.
    Step 14. Decide: If Value of <CVP> is Absent, Then Type_in "35".    //Insert CVP line
    Step 15. Look_for_object_whose Label is PA_cath_WP and_store_under
        <PA_cath_WP>.
    Step 16. Decide: If Value of <PA_cath_WP> is Absent, Then Type_in "36".
        //Insert PA catheter
    Step 17. Type_in "37".    //Treat probable causes (see 6.11 in HTA)
    Step 18. Delete <state>; Delete <ECG>; Delete <CVP>; Delete <PA_cath_WP>; Delete
        <PA_cath_CO>.
    Step 19. Return_with_goal_accomplished.

Method_for_goal: Prepare for_ACLS
    Step 1. Look_for_object_whose Label is sys_BP and_store_under <sys_BP>.
```

```
    Step 2. Decide: If Value of <sys_BP> is_less_than "60", Then Type_in "24".
        //Prepare for ACLS - Systolic BP is < 60
    Step 3. Decide: If Value of <sys_BP> is_less_than "40", Then Type_in "25".
        //Treat as cardiac arrest - go through ACLS - Systolic BP is < 40
    Step 4. Look_for_object_whose Label is MAP and_store_under <MAP>.
    Step 5. Decide: If Value of <MAP> is_less_than "30", Then Type_in "26".
        //Treat as cardiac arrest - go through ACLS - MAP is < 30
    Step 6. Look_for_object_whose Label is ECG and_store_under <ECG>.
    Step 7. Decide: If Value of <ECG> is_not "V tach",
                and Value of <ECG> is_not "V fib",
                and Value of <ECG> is_not "Pulseless v tach",
                and Value of <ECG> is_not "Atrial fib",
                and Value of <ECG> is_not "Supraventricular tach", Then
            Return_with_goal_accomplished.
    Step 8. Type_in "27".
        //Treat as cardiac arrest - go through ACLS - Severe arrhythmias present
    Step 9. Delete <sys_BP>; Delete <MAP>; Delete <ECG>.
    Step 10. Return_with_goal_accomplished.

Method_for_goal: Manage hydration
    Step 1. Look_for_object_whose Label is CVP and_store_under <CVP>.
    Step 2. Decide: If Value of <CVP> is_less_than_or_equal_to "4", Then Goto 8.
    Step 3. Look_for_object_whose Label is PA_cath_WP and_store_under <PA_cath_WP>.
    Step 4. Decide: If Value of <PA_cath_WP> is_less_than_or_equal_to "14", Then Goto 8.
    Step 5. Look_for_object_whose Label is PA_cath_CO and_store_under <PA_cath_CO>.
    Step 6. Decide: If Value of <PA_cath_CO> is_less_than "2", Then Goto 8.
    Step 7. Goto 9.
    Step 8. Type_in "29". //Patient may be dry; administer IV fluids as necessary
    Step 9. Look_for_object_whose Label is CVP and_store_under <CVP>.
    Step 10. Decide: If Value of <CVP> is_greater_than_or_equal_to "12", Then Goto 16.
    Step 11. Look_for_object_whose Label is PA_cath_WP and_store_under
        <PA_cath_WP>.
    Step 12. Decide: If Value of <PA_cath_WP> is_greater_than_or_equal_to "20", Then
        Goto 16.
    Step 13. Look_for_object_whose Label is PA_cath_CO and_store_under
        <PA_cath_CO>.
    Step 14. Decide: If Value of <PA_cath_CO> is_greater_than "3", Then Goto 16.
    Step 15. Goto 17.
    Step 16. Type_in "30".     //Patient may be wet; administer diuretic as necessary
    Step 17. Delete <CVP>; Delete <PA_cath_WP>; Delete <PA_cath_CO>.
    Step 18. Return_with_goal_accomplished.

Method_for_goal: Treat hypotension
    On_error: Determine state
    Step 1. Type_in "23".      //6. Treat hypotension and/or tachycardia:
                               //Confirm blood pressure change is real
```

//Cycle cuff and run cuff again
//Check O2 saturation
//If arterial line is in place
//Flush and zero arterial line
//Check height of transducer
//Check catheter tubing
//Make sure suppressed auto gain is off

Step 2. Accomplish_goal: Prepare for_ACLS.

Step 3. Type_in "28". //Recheck vaporizers are off

Step 4. Raise "Hypo_exception".

Step 5. Accomplish_goal: Manage hydration.

Step 6. Raise "Hypo_exception".

Step 7. Type_in "31". //Give vasopressor

Step 8. Type_in "32". //If patient is experiencing anaphylaxis (erythema, rash or
//wheeze is evident) or (HR < 130 and systolic BP < 40 or
//MAP < 50) or (cardiac arrest is imminent and rapid drop in
//BP), Consider epinephrine

Step 9. Type_in "33". //Improve patient posture:
//Put in Trendelburg position
//Elevate legs

Step 10. Type_in "34". //If patient is stable or help is available, increase monitoring:
//Increase ECG monitoring
//Insert arterial line

Step 11. Look_for_object_whose Label is CVP and_store_under <CVP>.

Step 12. Decide: If Value of <CVP> is Absent, Then Type_in "35". //Insert CVP line

Step 13. Look_for_object_whose Label is PA_cath_WP and_store_under <PA_cath_WP>.

Step 14. Decide: If Value of <PA_cath_WP> is Absent, Then Type_in "36". //Insert PA catheter

Step 15. Type_in "37". //Treat probable causes (see 6.11 in HTA)

Step 16. Delete <state>; Delete <ECG>; Delete <CVP>; Delete <PA_cath_WP>; Delete <PA_cath_CO>.

Step 17. Return_with_goal_accomplished.

Appendix G: Scenario Files for Anesthesia Decision Support Tool

1. Myocardial Infarction Scenario

HR	Systolic BP	MAP	CVP	PAWP	CO	ST Segment Shifts	ECG	SPO_2	$ETCO_2$
73	114	72	7	28	5.9	FALSE	0	98	39.1
73	115	73	10	27	5.9	FALSE	0	98	39
72	115	73	10	27	5.9	FALSE	0	98	39.2
72	116	73	8	30	5.9	FALSE	0	98	39.9
72	116	73	4	29	5.9	TRUE	0	98	39.2
73	114	72	10	27	5.9	TRUE	0	98	39.6
73	116	73	10	28	5.9	TRUE	0	98	39.6
73	117	74	10	30	5.9	TRUE	0	98	39.6
74	116	73	5	30	5.9	TRUE	0	98	40.4
74	116	73	9	28	5.9	TRUE	0	98	39.2
74	117	75	11	28	5.9	TRUE	0	98	39.3
74	117	75	11	29	5.9	TRUE	0	98	39.3
74	118	75	7	30	5.9	TRUE	1	98	40.4
77	115	77	6	28	5.9	TRUE	1	98	39.4
74	117	74	10	27	6	TRUE	1	98	39.5
75	115	74	10	28	5.9	TRUE	1	98	38.7
73	118	75	8	29	6	TRUE	1	98	40
73	118	75	4	31	5.9	TRUE	1	98	39.2
74	116	74	10	29	5.9	TRUE	1	98	39.2
73	117	78	11	28	5.8	TRUE	1	98	39.3
73	116	74	10	27	5.8	TRUE	1	98	39.3
84	95	58	10	28	5.8	TRUE	1	98	39.6
71	119	74	4	30	5.9	TRUE	1	98	39.6
73	114	72	9	28	5.9	TRUE	1	98	39.2
72	117	74	11	29	5.8	TRUE	1	98	39.1
71	118	75	10	30	5.8	TRUE	1	98	38.8
81	113	63	10	28	5.7	TRUE	1	98	40.2
72	115	72	5	28	5.7	TRUE	1	98	39.3
83	111	62	12	26	5.7	TRUE	1	98	38.4
83	112	63	11	25	5.8	TRUE	1	98	38.5
78	118	77	11	28	5.7	TRUE	1	98	39
72	99	72	9	28	5.7	TRUE	1	98	39.9
71	117	74	5	29	5.7	TRUE	1	98	38.9
74	115	72	9	27	5.8	TRUE	1	98	39.2
74	117	74	11	28	5.8	TRUE	1	98	39.6
74	117	74	11	28	5.8	TRUE	1	98	39.3
74	117	75	10	30	5.8	TRUE	1	98	39.3
74	118	75	4	30	5.9	TRUE	2	98	39.7
75	116	74	9	29	6	TRUE	2	98	39
74	117	75	11	28	6	TRUE	2	98	39
74	117	75	10	28	6	TRUE	2	98	39
65	121	77	5	31	5.9	TRUE	2	98	40
77	115	68	8	26	5.8	TRUE	2	98	38.4
74	116	73	10	27	6	TRUE	2	98	38.8

73	116	74	10	27	6	TRUE	2	98	39.3
69	117	74	10	28	5.9	TRUE	2	98	39.3
74	120	81	9	29	5.8	TRUE	2	98	40.7
70	116	73	4	28	5.9	TRUE	2	98	39.5
81	116	63	10	27	5.8	TRUE	2	98	38.3
74	116	73	10	25	5.9	TRUE	2	98	38.5
82	96	62	12	25	5.8	TRUE	2	98	38.9
71	118	73	7	30	5.8	TRUE	2	98	40.5
82	111	63	7	27	5.7	TRUE	2	98	38.1
73	117	76	11	28	5.7	TRUE	2	98	38.9
86	109	62	13	22	5.7	TRUE	2	98	38.8
70	120	75	10	31	5.6	TRUE	2	98	38.7
80	115	76	8	28	5.6	TRUE	2	98	40.7
151	71	57	9	18	5.6	TRUE	3	98	38.1
151	82	67	14	24	5.3	TRUE	3	98	37.9
151	83	67	13	24	5.1	TRUE	3	98	37.5
151	83	67	13	24	4.9	TRUE	3	98	37.4
151	82	67	9	23	4.8	TRUE	3	98	38.9
151	81	66	8	22	4.7	TRUE	3	98	38.2
151	80	65	11	20	4.6	TRUE	3	98	38.3
151	82	66	13	20	4.5	TRUE	3	98	38.4
151	82	67	14	21	4.5	TRUE	3	98	38.4
151	83	67	13	23	4.5	TRUE	3	98	38.4
151	83	67	13	24	4.5	TRUE	3	98	38.1
151	83	67	13	24	4.4	TRUE	3	98	38.2
151	83	67	12	24	4.4	TRUE	3	98	38.7
151	82	67	9	23	4.4	TRUE	4	98	39.1
151	20	20	18	18	3.5	TRUE	4	98	36.8
151	19	19	18	18	2.7	TRUE	4	98	36.8
151	19	19	18	18	2.1	TRUE	4	98	36.8
151	18	18	18	18	1.6	TRUE	4	98	36.8
151	18	18	18	18	1.3	TRUE	4	98	36.8
151	18	18	18	18	1	TRUE	4	98	36.8
151	18	18	18	18	0.8	TRUE	4	98	36.8
151	18	18	18	18	0.6	TRUE	4	98	36.8
151	18	18	18	18	0.5	TRUE	4	98	36.8
151	18	18	18	18	0.4	TRUE	4	98	36.8
151	18	18	18	18	0.3	TRUE	4	98	36.8
151	18	18	18	18	0.2	TRUE	4	98	36.8
151	18	18	18	18	0.2	TRUE	4	98	36.8
151	18	18	18	18	0.1	TRUE	4	98	36.8
151	18	18	18	18	0.1	TRUE	4	98	36.8
151	18	18	18	18	0.1	TRUE	4	98	36.8
151	18	18	18	18	0.1	TRUE	4	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8

151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
151	18	18	18	18	0	TRUE	5	98	36.8
120	108	79	10	41	0.6	FALSE	6	98	39.9
120	127	92	5	17	2.5	FALSE	6	98	38.7
120	108	77	9	26	3.6	FALSE	6	97	40.4
120	112	81	7	29	4.4	FALSE	6	97	40.8
120	108	77	5	26	5	FALSE	6	97	41.1
120	112	81	8	28	5.5	FALSE	6	97	40.9
120	111	79	4	28	5.9	FALSE	6	97	40.7
120	113	82	8	29	6.2	FALSE	6	97	41.4
120	110	79	8	26	6.5	FALSE	6	97	41.6
120	113	81	7	29	6.7	FALSE	6	97	41.2
120	108	77	9	24	6.8	FALSE	6	97	41.1
120	112	81	5	29	7	FALSE	6	97	43
120	109	78	9	24	7	FALSE	6	97	41.4
120	112	81	6	29	7.1	FALSE	6	97	41.9
120	108	77	8	24	7.2	FALSE	6	97	41.5
120	112	81	8	29	7.2	FALSE	6	97	41.4
120	109	78	5	26	7.3	FALSE	6	97	41.7
120	111	80	8	26	7.3	FALSE	6	97	41.4
120	111	80	2	28	7.3	FALSE	6	97	42.7
120	110	79	8	24	7.3	FALSE	6	98	41.5
120	112	81	8	29	7.3	FALSE	6	98	41.1
120	110	78	1	27	7.3	FALSE	7	98	41.1
115	22	22	18	17	6	FALSE	7	98	41.7
115	20	20	18	18	4.6	FALSE	7	98	41.7
115	19	19	18	18	3.6	FALSE	7	98	41.7
115	19	19	18	18	2.8	FALSE	7	98	41.7
115	18	18	18	18	2.2	FALSE	7	98	41.7
115	18	18	18	18	1.7	FALSE	7	98	41.7
115	18	18	18	18	1.3	FALSE	7	98	41.7
115	18	18	18	18	1	FALSE	7	98	41.7

2. Hypotension Scenario

HR	Systolic BP	MAP	CVP	PAWP	CO	ST Segment Shifts	ECG	SPO_2	$ETCO_2$
109	81	56	0	9	4.3	FALSE	0	98	37.4
108	83	58	0	10	4.3	FALSE	0	98	37.4
108	84	59	0	13	4.3	FALSE	0	98	37.4
107	85	59	0	13	4.3	FALSE	0	98	37.5
107	84	59	-2	13	4.3	FALSE	0	98	38.2
107	82	58	-4	12	4.3	FALSE	0	98	38.2
109	80	57	-5	10	4.3	FALSE	0	98	37.1
110	83	58	0	10	4.3	FALSE	0	98	37.4
110	83	58	0	11	4.3	FALSE	0	98	37.2
111	79	58	0	11	4.2	FALSE	0	98	37.5
110	79	57	0	13	4.2	FALSE	0	98	37.4
110	79	57	-3	12	4.2	FALSE	0	98	38.4
117	68	50	-6	8	4.2	FALSE	0	98	36.9
117	55	41	-3	3	4	FALSE	0	98	35.8
117	55	40	-1	4	3.7	FALSE	0	98	35.7
117	55	40	-1	5	3.6	FALSE	0	98	35.6
116	56	41	-1	6	3.5	FALSE	0	98	35.8
116	56	41	-1	7	3.3	FALSE	0	98	36.3
116	50	37	-1	7	3.2	FALSE	0	98	36.4
116	50	37	-1	7	3.1	FALSE	0	98	36.1
116	51	38	-1	7	3	FALSE	0	98	36.2
117	51	38	-1	7	2.9	FALSE	0	98	36
117	51	37	-1	7	2.9	FALSE	0	98	35.9
117	51	37	-1	7	2.8	FALSE	0	98	35.8
117	51	37	-1	8	2.8	FALSE	0	98	35.8
117	51	37	-1	8	2.8	FALSE	0	98	35.7
117	51	37	-1	8	2.7	FALSE	0	98	35.8
117	51	37	-1	8	2.7	FALSE	0	98	35.6
117	51	37	-1	8	2.7	FALSE	0	98	35.6
118	51	37	-1	7	2.7	FALSE	0	98	35.6
118	51	37	-3	7	2.7	FALSE	0	98	36.3
118	49	36	-5	7	2.7	FALSE	0	98	36.4
118	49	36	-6	6	2.7	FALSE	0	98	35.7
118	47	34	-6	5	2.7	FALSE	0	98	35

ECG values:		
	0	none
	1	10% PVCs
	2	25% PVCs
	3	ventricular tachycardia
	4	pulseless ventricular tachycardia
	5	ventricular fibrillation
	6	sinus tachycardia
	7	asystole