

Requirements Definitions of Real-Time System Using the Behavioral Patterns Analysis (BPA) Approach: The Missile Control System

Assem El-Ansary

Abstract—This paper illustrates the event-oriented behavioral pattern analysis (BPA) modeling approach. The Event defined in BPA is a real-life conceptual entity that is unrelated to any implementation. The BPA Behavioral Patterns are temporally ordered according to the sequence of the real world events. The major contributions of this research are:

The behavioral pattern analysis (BPA) modeling methodology.

Validation of the hypothesis that the behavioral pattern analysis (BPA) modeling methodology is a more effective alternative to use case analysis (UCA) in modeling the functional requirements of Human-Machine Safety-Critical Real-time Systems.

The development of an interactive software tool (DECISION), which is based on a combination of the analytic hierarchy process (AHP) and the ELECTRE multi-criteria decision making (MCDM) methods. The DECISION software tool was used to process the assessment results of the case studies.

Index Terms—Analysis, modeling methodology, software modeling, event-oriented, behavioral pattern, use cases.

I. INTRODUCTION

Experience reports problems with Use Cases such as [1]:

- 1) The lack of a Use Case formal specification.
- 2) Lack of atomicity has been the reason for generating hundreds of use cases for some simple applications.
- 3) The lack of a notion of atomicity has made the measurement of a project's task complexity, by counting the use cases, unreal.
- 4) The absence of the notion of triggering events.
- 5) There a problem with the phrase use cases itself.

A major problem in the use case approach is its tendency to focus on the solution rather than the problem. Jacobson defined use case as “a behaviorally related sequence of transactions in a dialogue with the system” [2]. The processing of transactions, or operations, or use cases is what the machine does. It is part of the solution, not part of the problem [3].

The concluding statement of the “Question Time! About Use Cases” Panel of the OOPSLA'98 Conference by Ian

Graham [4] was “There is a need for another modeling methodology with a sound theoretical basis and a precise definition.” This need is what this research problem area is about.

In addition to the problems with the use cases [3], [4] that were described briefly above, several additional problems were identified during this research [5], [6]. The following is a discussion of these problems:

- The types of interactions are: interactions among users, interactions between users and the system, and interactions among the different components of the system. Yet, use cases describe only the users' interaction with the system. This is just one type of interaction. Hence, the resulting object (class) model, and the sequence diagram may be incomplete.
- Using natural language in use cases description, with the absence of any semantic structure such as alternation or repetition, increases the risks of ambiguity, incompleteness, and inconsistency.
- One may argue that any missing interaction description may be captured via the state diagrams. However a missing object or interaction is unlikely to be captured and explicitly represented in these diagrams. Also, a state diagram describes an individual object's response to specific events rather than objects interaction. Hence, objects interaction must be reconstructed from the analysis of groups of diagrams. Such a task is complex and error-prone.

In conclusion, if the analyst misinterpreted or neglected some structural or behavioral aspects, the resulting conceptual model will not be a good representation or understanding of the real world. The resulting software solution system built from the model may not demonstrate the correct behavior or may ungracefully terminate. The end result might be the loss of opportunities in using business systems, serious damages in embedded systems, or the loss of lives in using a safety-critical system.

In the BPA modeling methodology, the BPA Behavioral Pattern, which is the template that one uses to model and describe an event, takes the place of the Use Case in the UML Use Case View. The BPA Behavioral Patterns are temporally ordered according to the sequence of the real world events.

II. ILLUSTRATING BPA THROUGH THE MISSILE CONTROL SYSTEM (MCS)

The main function of the MCS [7] is to control missiles in

a naval vessel control system. The following describe the operation:

- The operator types information into the system defining the missile target, in terms of location and time.
 - The missile trajectory will be calculated, and at the correct time will be launched from a big gun, which has been positioned in accordance with the calculated trajectory.
 - Although the gun launches the missile in the in the correct direction, it is possible that in certain weather conditions, the missile may be blown off course.
 - Therefore, it is necessary to track the missile throughout its flight to ensure that it sticks to the calculated trajectory. As the missile is being tracked, a display of the current missile location will be given to the operator. Also, at anytime, the operator may change the target coordinates, either because the target has moved, or because a mistake has been made.
 - Therefore, once the missile has been launched, the system will continue to provide it with the directional information to steer it towards the target.
 - The missile will detonate, either on command from the control system or on impact with the target.
 - This system will be spread across three processors:
 - a) Base station processor, responsible for the operator interface and any remote control of the gun and missile
 - b) Missile processor that allow it to communicate with the base station. It is envisaged that some intelligence will be placed within the missile to allow it to 'steer' itself
 - c) Gun processor that may enable the gun to aim itself and fire the missile in a stand alone mode.
- 3) Analyze the Originating Requirements to identify the Critical Constraints (e.g. Safety) and/or the Utility Requirements.
 - 4) Decompose the scoped problem (from Step 2) into Main Events based on the Mission and Operating Requirements (Step 1).
 - 5) Using the identified Main Events, draw the High Level Event Hierarchy Diagram which is constructed in several levels whose top level includes the highest main event (Fig. 3).
 - 6) Decompose these identified Main Events into smaller and simpler events represented as Episodes (Composite Events) with clear boundaries¹.
 - An Episode Boundary at this stage may be marked with Location / Loci of Control and Effect.
 - 7) Add additional levels to the Event Hierarchy Diagram (Event Hierarchy Sub-Diagrams). For complex problems, it is often helpful to extract these sub-diagrams and analyze them. Detailed level event hierarchy diagrams are drawn as necessary.
 - Decomposition Heuristics at this stage is 'One Agent and One Location'
 - 8) For each identified main event (from Step 4) draw an Event Thread Diagram which represents the events' sequence (Fig. 4)
 - Starting with the Main Events, as initial composite events, recursively decompose the composite events into Basic Events
 - The Event Decomposition Heuristics at this stage is 'One Agent, One Location, One Motion Direction, and One Time Interval'.
 - Group Basic Events by their Location / Loci of Control and Effect. Draw a frame box around these Basic Events.
 - 9) Refine and transform the above Basic Events into their corresponding BPA Behavioral Patterns which describes the which, who, when, and where of each of the basic events (Fig. 5)
 - 10) Using the Event Thread Diagrams from Step 8, draw the Temporal/Causal Constraint Diagrams by adding the temporal constraints (time order as illustrated in Fig. 6 and Fig 7) alongside the associations and identifying the enable/causal relationships (Enable is what makes it ready, and Causal means making something happen) in each corresponding Event Thread Diagram (Fig. 8).
 - 11) Using the Critical Constraints (e.g. Safety), identify the critical events, identify all possible ways of each critical event's failure, and draw the Critical Event Analysis Diagram (Fig. 9).
 - 12) Using the BPA Event Patterns and the Critical Event Analysis Diagrams, identify any missing requirements that are necessary to satisfy the critical constraints. One develops a Derived Requirements document and get users approval on this document.
 - 13) Using the Missing Requirements (from Step 12), refine the Event Hierarchy Diagram (from Step 6), the

III. RESEARCH THESIS

The specific thesis is that the proposed Behavioral Pattern Analysis (BPA) approach is more effective than the Use Case Analysis (UCA) approach at modeling the functional requirements of Interactive Safety-Critical Real-time Systems. To validate that the BPA approach is more effective than the Use Case approach, sixteen Subject Matter Experts were given two case studies that are modeled using the two approaches and were asked to evaluate the models using the Safety, Repeatability, Unambiguity, Completeness, Consistency, Modifiability, and Traceability as the effectiveness criteria.

The following subsection presents a summary of the research approach.

IV. THE BPA REQUIREMENTS DEVELOPMENT PROCEDURE

The following is an outline of the BPA functional requirements development procedure (see Fig. 1 and Fig. 2):

- 1) Identify the problem at the highest level of abstraction (e.g. The Mission Statement and Operating Requirements).
- 2) Identify the scope of the requirements (problem) from the Originating Requirements.

Thread Diagrams (from Step 8), and the Temporal Constraint Diagram (from Step 10) as necessary. Draw additional Event Thread Diagrams for identified critical events as necessary.

with the Originating Requirement and Steps 1 to 3, then Steps 4 to 7, then Step 8, then Refine and come-up with the Derived Requirements which covers any Missing Requirements as explained in Steps 9 and 10. After that we re-iterate as explained in Steps 11 and 12.

The figure below illustrates the BPA iterative and incremental development process. The figure shows the start

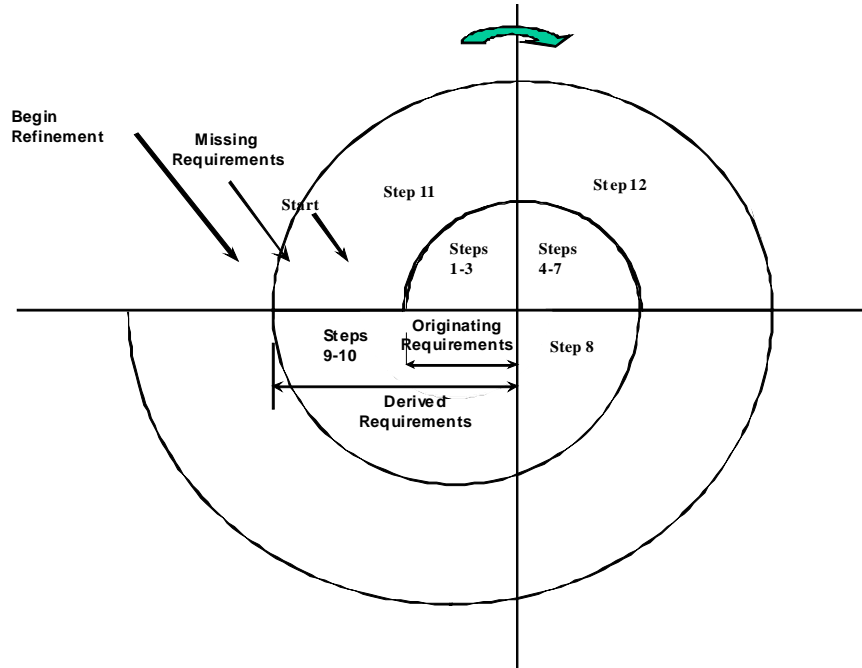


Fig. 1. The BPA modeling process.

- 14) Using the BPA Behavioral Patterns (from Step 9), identify the candidate Classes from the Event Roles (Participants) and Instrument. Draw the Class Diagram (Fig.10).
- 15) To illustrate the relationship between Events and States, optionally, using the BPA Behavioral Patterns, draw the Event/State History Chart (Optional – not shown) that includes the States before and after each Event for each identified Class whose instance is a participant in that Event.

The above procedure illustrates the BPA functional requirements development procedure. The Fig. 2 depicts the flow of the modeling activities (Steps 1 to 14) for the BPA procedure.

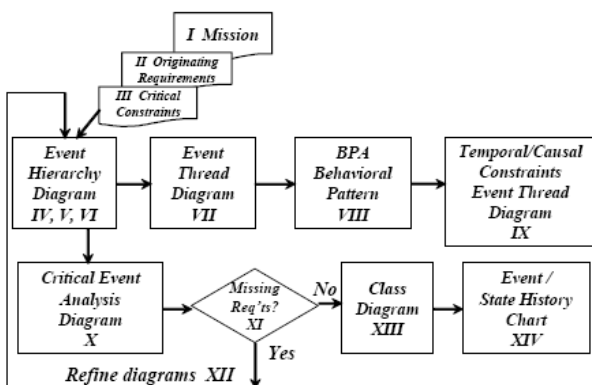


Fig. 2. Requirements development procedure.

A. Event Hierarchy Diagram (EHD)

Event Hierarchy (Fig. 3) is used to model the events at different levels of abstraction (event decomposition). A general problem with decomposition is when to stop the decomposition. The decomposition heuristic used in an **Event Hierarchy Diagram (EHD)** is one agent and one location. Using this heuristic, the leaf events in an Event Hierarchy are usually Simple Sequence Events. In other words, a leaf event is usually a set of Basic Events (atomic events) sequenced into episode1. The episode is marked with a location boundary. The following is the MCS detailed Event Hierarchy Diagram:

Using the identified main events, the high level EHD diagram (or the first level in a detailed EHD diagram) is drawn. Each main event is then decomposed further until one arrives at leaf events, each of which has one location or one locus of effect and control and one agent. The top level in this figure is Missile Control System. As per Steps 4, 5, and 6 in the BPA procedure, the second level includes the Base Controlling Missile, Controlling Gun Steering Missile Events, and the third level includes sub-events of the second level's event which are Calculating Missile Schedule and Sending Target To Missile (sub-events of Base Controlling Missile), Position Gun (sub-event of Controlling Gun), and Controlling Missile, Tracking Missile, Displaying Missile, Adjusting Target (sub-events of Steering Missile).

In order to model the sequence of events (and show the location / loci of control and effect view, or the temporal / causal constraints), one uses the event thread diagrams as

shown in the next subsections.

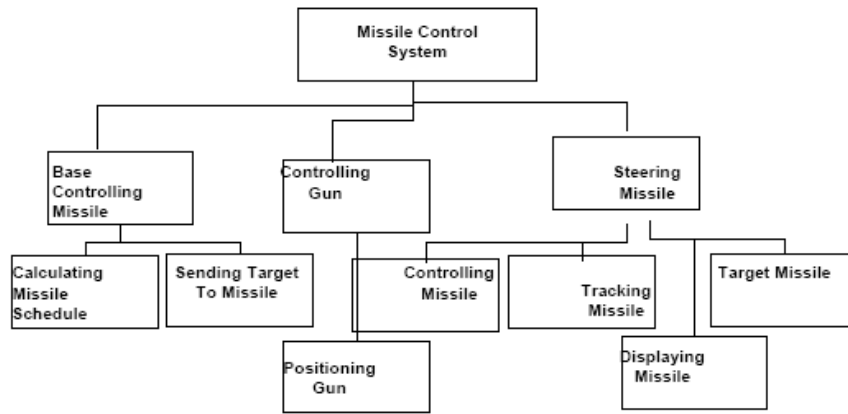


Fig. 3. Event hierarchy – missile control.

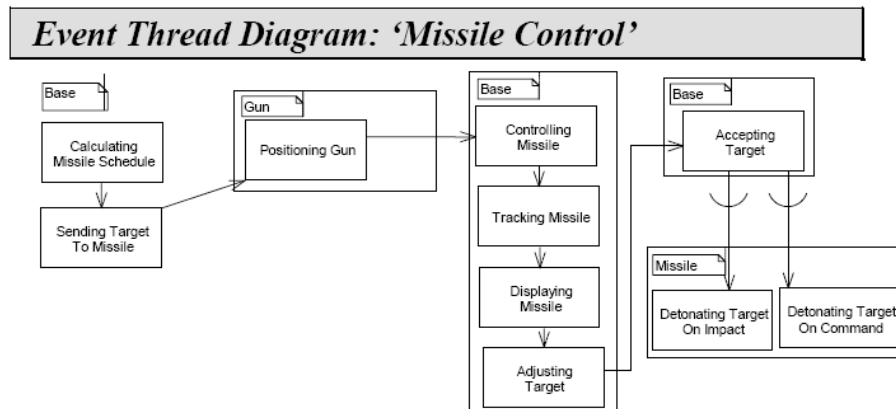


Fig. 4. Event thread –missile control system.

BPA BEHAVIORAL PATTERN - EXAMPLE

Event (WHAT?) Controlling Missile
Actions
 1. 2.
 3. 4.
 5. 6.
Agent a: Operator
 > Initial State: Scheduled Missile
 > Final State: Released Missile
Affected p: Missile
 > Initial State: Idle
 > Final State: Released
Modality (HOW?)
Instrument
 i: Base Processor
Circumstances
Manner
 m: Critical
Condition
 c1: Released c2:
Effect
 fl: Flying fl:
Date/Time (WHEN?)
 t: After Stopping Gun Positioning
Place (WHERE?)
Location
 l: Base
Path
Motion
 m: Flying
Direction
 d: Calculated Trajectory
Rationale (WHY?)
 g: Tracking Missile
Mental State
 bdi:
Caused-By
 e': Base Processor
End;

Fig. 5. BPA pattern – missile control.

B. Event Thread Diagram (ETD)

In BPA, as per Step 8, an Event Thread Diagram (ETD – Fig. 4) is drawn for each main event, and optionally drawn for any other event, subordinate to main event, depending on its complexity or its critical nature.

A Basic Event is defined as an event that cannot be decomposed into another set of events (atomic event). The heuristic used in decomposing is one agent, one location, one time interval, and one motion direction if the event involves any motion. The ETD, which one draws for an event, shows the sequence of the basic events of that event. The figure shows the following sequences:

- Calculating Missile Schedule, Sending Target to Missile (for location Base),
- Position Gun (for location Gun),
- Controlling Missile, Tracking Missile, Displaying Missile, Adjusting Target (for location Base) ,
- Accepting Target (for location Base) which results in either Detonating Target on Impact or Detonating Target on Command (for location Missile).

C. Behavioral Patterns

As explained in Step 8, the research goal is to develop a requirements definition mechanism (BPA Pattern – Fig. 5) that describes the What, Who, How, When, Where and Why.

D. Introducing Time

The key intuitions motivating the introduction of time are:

- Events take time. Yet, in most of the popular Object-Oriented Modeling methodologies such as OMT and UML, time is neglected in the event definition.
- Multiple events may occur at the same time, and could be unrelated, cooperating, or interfering with each other.
- Events may have temporal constraints. They may overlap, start or finish together, occur together, or disable (disjoint) each other. BPA uses the time intervals' relations that are described in the Interval Algebra framework [8] to model the temporal relationships between events. In this Interval Algebra framework, seven basic relations can hold between time intervals. Fig. 6 and Fig. 7 illustrates these basic relations for arbitrary events x and y .

REL	SYM	MEANING
x before y	b	
x meets y	m	
x overlaps y	o	
x starts y	s	
x during y	d	
x finishes y	f	
x equals y	eq	

Fig. 6. Time interval algebra – temporal relations.

E. Introducing Enable / Cause Relationships

The introduction of the Enable¹ / Cause relationships between events will enable the analyst to do cause effect analysis and reason about any possible failure of the system.

Temporal Relations

b Before
m Meets
o Overlaps
d During
s Starts
f Finishes
eq Equals
i Inverse

Fig. 7. Time interval algebra - temporal relations notation.

¹ 'Enable' is defined in the American Heritage Dictionary as: "To supply with the means, knowledge, or opportunity; make able: a hole in the fence that enabled us to watch; techniques that enable surgeons to open and repair the heart."

In the Temporal Constraint Diag., as described in Steps 9, and 10, the temporal relations that are displayed in Fig. 8 are written alongside the sequence relationships to represent the possible timing at which these events can occur. The figure shows that Calculating Missile Schedule is before $\{b\}$ Sending Target to Missile, which is $\{b\}$ Positioning Gun, which is before or during $\{b, d\}$ Controlling Missile, which is $\{b, d\}$ Tracking Missile, which is $\{b, d, m\}$ Displaying Missile, which is before or during or meets $\{b, d, m\}$ Adjusting Target which before or starts Accepting Target $\{s, b\}$, which is $\{b\}$ Detonating Target On Impact or On Command.

F. Failure Issues

The following is a list of reasons of possible failures in responding to events:

- Occurrence of a relevant event which the system does not handle
- Event rate exceeding the system's capacity
- Unsuccessful detection and acquisition of all events including manually captured events
- Non-capturing of all information triggered by event
- Failure across man-machine interface
- Failure of Software, Hardware, or Human.

The ability to provide requirements specification for safe behavior is very limited using the current modeling methodologies. Neither a safety analysis (anterior analysis) nor accident analysis (posterior analysis) can be achieved efficiently without event analysis. As will be explained below, the BPA modeling methodology provides the Critical Event Analysis (defined below) as an efficient solution to this problem.

G. Critical Events Analysis

The requirements should correctly reflect the critical properties of the environment in which software is to work. In order to gain as much confidence as possible in the software for a critical system, the analyst should perform a 'Critical Event Analysis'. The Critical Event Analysis procedure includes the following Steps:

- Identify Critical Events
- For each critical event, identify all possible ways in which it may fail
- Capture these possible failure modes using the undesired event notation
- Study each undesired related state to find out how to achieve protection against such possible failure

The following diagram (Fig. 9) illustrates the critical event analysis in BPA as described in Step 11:

In the Critical Analysis Diagram, the round ended rectangles represent the states of the critical events. The dashed rounded ended rectangles represent the failure that occurs due to these states.

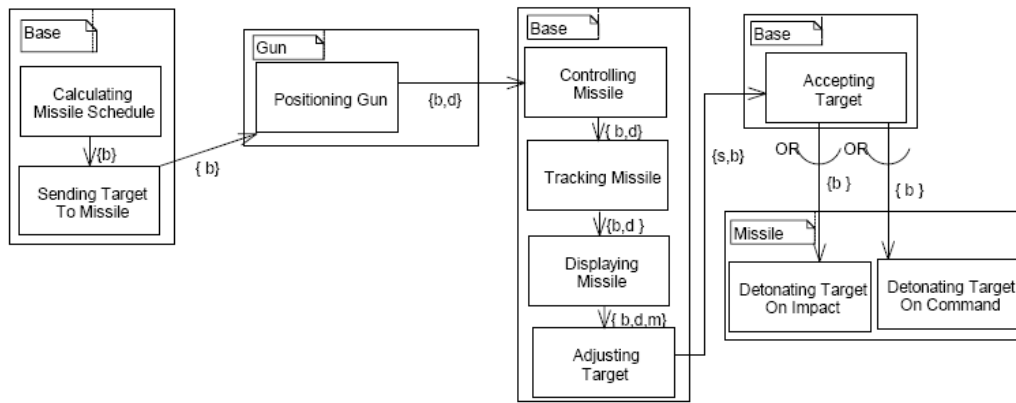


Fig. 8. Temporal constraint diagram – missile control.

V. MISSING REQUIREMENTS

There were no missing requirements that required generating a Derived Requirement Document.

VI. MCS CLASS DIAGRAM

The resulting Class Diagram is shown in Fig. 10.

The identified classes are Base that has zero to many Base Processor each of which schedule zero to many Missile Processor, Gun that has one Gun Processor which releases zero to many Missile Processor, Missile that has one Missile Processor and travels zero to many Trajectory, Target that has one to many Target Trajectory.

VII. EVALUATION OF THE EFFECTIVENESS OF THE BPA MODELING METHODOLOGY AND THE UCA MODELING METHODOLOGY

In this research, three real-life applications were used to illustrate the effectiveness of the new BPA modeling methodology in handling safety-critical real-time systems development:

- The Therac-25 Medical Device System [9]
- The Production Cell System [10]
- The Railroad Crossing System [11].

The UCA and the BPA modeling methodologies were used to define the requirements and model these systems. The first application was used, as a proof of concept, in a pilot case study. The last two applications were distributed as part of the case studies material to compare the UCA versus the BPA modeling methodologies using the pre-mentioned effectiveness criteria.

VIII. THE EFFECTIVENESS METRICS

The effectiveness metrics categories used in this research include:

- 1) System Effectiveness represented by safety
- 2) Requirements Engineering Process Effectiveness represented by the CMM [12] and CMMI repeatability

3) Definition of Requirements Effectiveness represented by the ANSI (NIST) / IEEE Std 830-1984 [13] for systems specifications:

- Unambiguous
- Complete
- Consistent
- Modifiable
- Traceable
- Usability

However, there is no evidence that Usability is an independent characteristic. It can mean Unambiguous, Complete, or Modifiable. Because we have included all of the pre-mentioned characteristics, Usability was taken out.

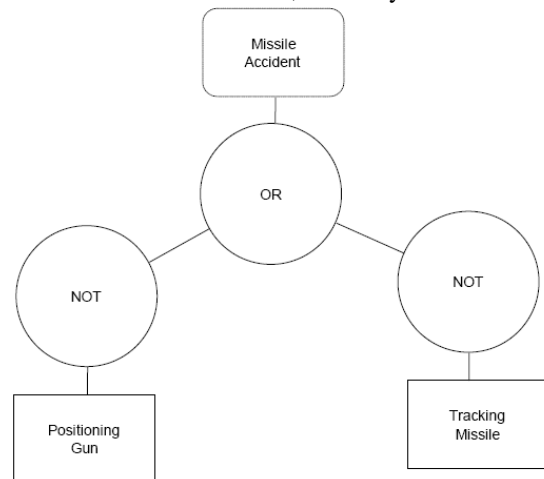


Fig. 9. Critical analysis diagram – missile control.

IX. THE PAIRWISE COMPARISON METHOD

A Multi-Criteria Decision Making (MCDM) Tool, named as DECISION, was developed by this researcher to evaluate the assessment results. The Decision tool uses a combination of the Analytic Hierarchy Process (AHP) and the ELECTRE Pairwise Comparison approaches. Pairwise Comparisons is the process in which experts rate a set of objects, events, or criteria, by comparing only two at a time. Most people are reliable estimators using pairwise comparisons because they only have to consider two things at a time [14]. The selected approaches, AHP and ELECTRE, are popular and have strong theoretical basis

[15], [16].

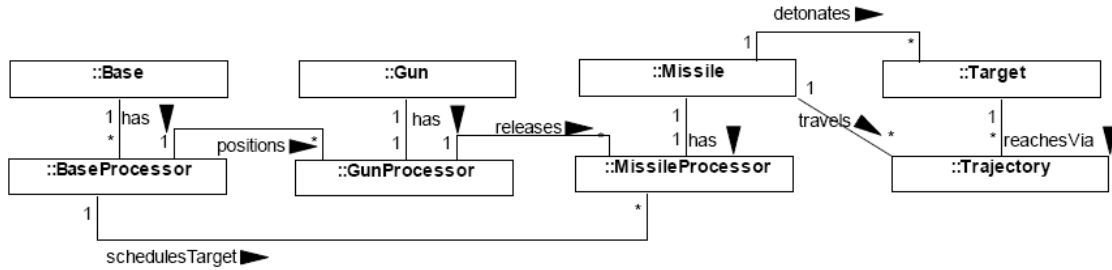


Fig. 10. Class diagram – MCS.

X. CASE STUDY PROTOCOL DESIGN

The protocol design stage is composed of two main tasks:

- 1) Determine the required skills of the subject matter experts (SMEs) in Structured Analysis and UML experience using questionnaires.
- 2) Develop and review the protocol whose guideline is to document the Steps to be able to repeat the procedures and arrive at the same results.

XI. THE CASE STUDY MATERIAL

Each SME was provided with a case study kit that contains the instructions, an application, an overview and a step by step procedure describing how to analyze and model requirements using the UCA and BPA modeling methodologies, two analyses of the given application; one using the UCA modeling methodology and the other using the BPA modeling methodology, explanation of the evaluation method (Pairwise Comparison) and the effectiveness criteria. The set of questions presented clearly in a table format (Evaluation Forms).

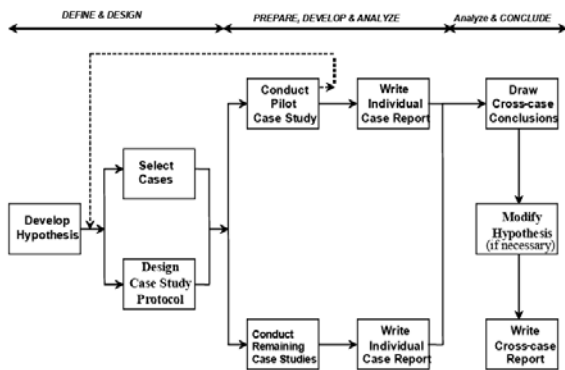


Fig. 11. Case study research method.

XII. SUBJECT MATTER EXPERTS' SELECTION

The SMEs were selected from two sets of SE professionals. The first set was composed of SE professionals who happen to be graduate students in the Information Technology School and attended one or more software engineering courses. The second set was composed of working professionals at Lockheed Martin Company and Federal Government. Email and surface mail letters were sent to more than 100 of these graduate students and

working software engineering professionals. Sixteen SE professionals, with the required experience to carry out the case studies, were selected out of these two sets of professionals that showed interest in participating. Questionnaires were sent to these SMEs to classify them according to their Structured Analysis (SA) methods and UML / UCA knowledge. From each set, two with the same kind of experience out of each group were selected to receive one of the two case studies' apps.

XIII. THE SUBJECT MATTER EXPERTS

The number of SMEs depends on the number of the controlled variables. The controlled variables are:

- The applications.
- The set of the SMEs.
- The SMEs' software engineering experience:
 - a) Structured Analysis
 - b) Use Case Analysis / UML.

Table I illustrates this break down by experience & App.

TABLE I: SME EXPERIENCE TO APPLICATION ASSIGNMENT MATRIX

Experience/Applic ation	Production Cell Case Study	Rail Crossing Case Study
Structured Analysis and Design	4	4
Use Case Analysis / UML	4	4

XIV. CASE STUDIES' RESULTS

A. Case Studies Results

1) AHP results

	BPA	UCA
SME 5C1	0.89	0.11
SME 4C1	0.75	0.25
SME 1C1	0.57	0.43
SME 8C2	0.75	0.25
SME 7C2	0.66	0.34
SME 3C2	0.8	0.2
SME 2C2	0.57	0.43
SME 6C1	0.47	0.53
SME9C1	0.73	0.27
SME10C1	0.79	0.21
SME11C1	0.76	0.24
SME12C1	0.85	0.15
SME13C2	0.77	0.23
SME 14C2	0.84	0.16
SME 15C2	0.81	0.19
SME 16C2	0.88	0.12

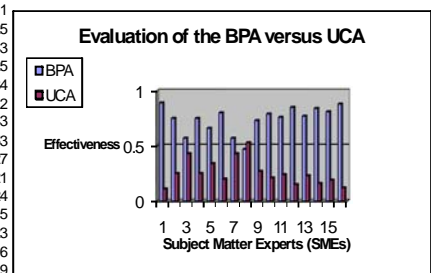


Fig. 12. Effectiveness evaluation of BPA versus UCA – results using AHP.

The summary of the assessment results using AHP is illustrated in Fig. 12 in a column chart format.

The above results show that:

- 15 SMEs out of 16 evaluated BPA as a more effective alternative to UCA in defining the requirements. This result gives an indication of about 93.8 % approval rate for the thesis hypothesis.
- 1 SMEs out of 16 evaluated UCA as more effective alternative to BPA in defining the requirements.
- The average of the overall priorities for BPA (total of the overall priorities of the SMEs/ the total number of SMEs, which is 11.89/16) is about 0.74.
- The average of the overall priorities for UCA (total of the overall priorities of the SMEs / the total number of SMEs, which is 4.11/16) is about 0.26.

The above results give an indication of about 93.8 % approval rate for the thesis hypothesis with about three times overall effectiveness for BPA over UCA.

2) Electre results

The following is a collective summary by number of SMEs using ELECTRE:

- Fourteen SMEs out of sixteen evaluated BPA as a more effective alternative to UCA in defining the requirements.
- 2 SMEs out of 16 evaluated UCA as more effective alternative to BPA in defining the requirements.

Fig. 13 shows that there is 87% approval rate for the thesis hypothesis.

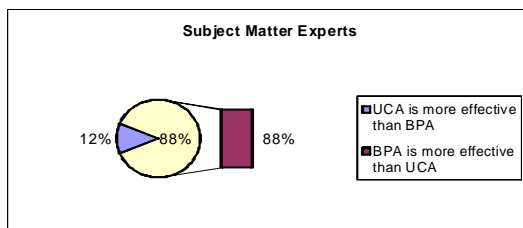


Fig. 13. Evaluation results' summary by number of subject matter experts (SMEs).

XV. RESEARCH CONTRIBUTION

The major contributions of this research are:

- The Behavioral Pattern Analysis (BPA) modeling methodology.
- Validation of the hypothesis that the Behavioral Pattern Analysis (BPA) modeling methodology is a more effective alternative to Use Case Analysis (UCA) in modeling the functional requirements of Human-Machine Safety-Critical Real-time Systems.
- Another contribution of this research was the development of an interactive software tool (DECISION) that is based on a combination of the Analytic Hierarchy Process (AHP) and the ELECTRE Multi-Criteria Decision Making (MCDM) methods. The DECISION software tool was used to process the assessment results of the case studies.

XVI. WHY THIS WORK IS IMPORTANT

A. Real-Time Systems

In most of the popular object-oriented development modeling methodologies state diagrams are used to model the behavior. By using state diagrams, one is focusing on an individual object's response to specific events rather than objects interaction. Hence, objects interaction must be reconstructed from the analysis of groups of diagrams. Such a task is at least complex and error-prone. By describing the requirements in terms of events, represented by the behavioral patterns, this perceived problem is reduced.

B. Multi-Agent Systems

There is a need for a multi-agent systems analysis and design method that is powerful enough to model interaction patterns involving autonomous agents.

C. Safety-Critical Systems

In these systems, analysts should perform a 'Safety Analysis'. Using BPA, one identifies and documents the critical events during the requirements definition stage.

GOD says [KORAN][TORAH], "... Whoever rescues a single life earns as much merit as though he had rescued the entire world." If the use of the BPA Modeling methodology may save one life, the significance of this modeling methodology is immeasurable.

REFERENCES

- [1] G. Ian, *Migrating to Object Technology*, Addison-Wesley, Reading, Massachusetts, 1995.
- [2] I. Acobson, M. Christeron, and Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, Reading, Massachusetts, 1992.
- [3] J. Michael, *Software Requirements and Specification, A Lexicon of Practice, Principles and Prejudices*, ACM Press, Addison-Wesley, Reading, Massachusetts, 1995.
- [4] M. F. Cockburn, *Question Time! About Use Cases, OOPSLA'98 Proceedings*, ACM Press, New York, NY, 1998.
- [5] E. Ansary and I. Assem, "Behavioral pattern analysis: towards a new representation of systems requirements based on actions and events," in *Proc. the 2002 ACM Symposium on Applied Computing*, ACM, New York, NY, pp. 984-991, 2002.
- [6] E. Ansary and I. Assem, "Behavioral pattern analysis: towards a new representation of systems requirements based on actions and events," Doctoral Thesis, George Mason University, 2005.
- [7] G. Sylvia, *A Practical Guide to Real-Time Systems Development*, Prentice Hall International, London, UK, 1995.
- [8] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of ACM*, vol. 26, 1983, pp. 832-843.
- [9] L. Nancy, *Safeware*, Addison-Wesley, Reading, Massachusetts, 1995.
- [10] L. Claus and L. Thomas, *Formal Development of Reactive Systems*, Springer-Verlag, NY, 1995.
- [11] H. Constance and M. Dino, *Formal Methods for Real-Time Computing: An Overview*, in *Formal Methods for Real-Time Computing*, John Wiley & Sons, Inc., NY, 1996.
- [12] H. Watts, *Software Process Maturity Framework*, Addison-Wesley, Reading, Massachusetts, 1989.
- [13] *IEEE Guide to Software Requirements Specification*, in *System and Software Requirements Engineering*, IEEE and ANSI, ANS/IEEE Std 830-1984, IEEE Computer Society Press, Los Alamitos, California, 1990, pp 170-192.
- [14] S. L. Thomas, *Decision Making for Leaders*, Wadsworth, Inc., 1982.
- [15] M. Meyer and J. Booker, *Knowledge Based Systems, Eliciting and Analyzing Expert Judgment, A Practical Guide*, Academic Press, Inc., 1991.
- [16] B. X. Tung and C. OP, *A Group Decision Support System for Cooperative Multiple Criteria Group Decision Making*, Springer-Verlag, 1987.



Assem I. El-Ansary was born on September 02, 1944, in Elfashn, Egypt. He earned his BSc in 1969 in electrical engineering from Cairo University. He earned his master's degree in MIS in 1979 from American University. He earned his PhD in IT in 2005 from George Mason University (GMU) in Fairfax, USA.

He has thirty years of experience in the Development of Large-Scale Software Systems. Assem's working experience includes serving as the head of Computer Department in AIDO industrial organization in Egypt, and Iraq. He has worked as a project manager and principal system analyst at Lockheed martin, and advanced technology systems in USA.

In February 2000, he established his own consulting firm, Emergent Technologies USA, Inc, in Vienna, VA, USA, in which he is working until now as a CEO/principal analyst. As an educator, he worked as a professor at GMU, PROF. at JHU, and an associate professor at MD University in USA. His research interests include software engineering, DSS, and AI. He has published a number of papers in the software engineering area:

Assem El-Ansary Assem developed a software system (Decision) to elicit and analyze experts' judgment using multi-criteria decision making (MCDM) techniques. He is a Member in ACM, and IEEE. He worked as technical committee member in CIMCA 2004, 2005, 2006, and ICSEM 2014. He received the "Developing Courses around the Clock" Award in 190 from Intervolve, and the "Year 2000" Award in 2000 from ATS.