
HIM 1990-2015

2011

Context-driven agents in computer supported cooperative works

Brian D. Lichtman
University of Central Florida

 Part of the [Computer Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/honorstheses1990-2015>

University of Central Florida Libraries <http://library.ucf.edu>

This Open Access is brought to you for free and open access by STARS. It has been accepted for inclusion in HIM 1990-2015 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

Recommended Citation

Lichtman, Brian D., "Context-driven agents in computer supported cooperative works" (2011). *HIM 1990-2015*. 1229.

<https://stars.library.ucf.edu/honorstheses1990-2015/1229>

CONTEXT-DRIVEN AGENTS IN
COMPUTER SUPPORTED COOPERATIVE WORKS

by

BRIAN D. LICHTMAN

A thesis submitted in partial fulfillment of the requirements
for the Honors in the Major Program in Computer Engineering
in the College of Engineering and Computer Science
and in The Burnett Honors College
at the University of Central Florida
Orlando, Florida

Fall Term 2011

Thesis Chair: Dr. Avelino J. Gonzalez

ABSTRACT

This thesis describes a research project that investigates the level of contextualization needed to successfully build context-driven agents that can manage a cooperative project. Many times in industry, collaborators in a large project may be located vast distances from each other. It is for this reason that management of such projects can often be difficult. The purpose of this research is to design an agent that can take on the role of a project manager (PM) to assist the human project manager. Specifically, this thesis looks to give such project management agents full situational awareness. It is hypothesized that only with situational awareness can an agent successfully act in the role of a project manager. This thesis describes the investigation into the use of Context-Based Reasoning and Contextual Graphs to create an agent with such situational awareness. This thesis shows that with enough situational awareness, an agent will have the ability to successfully take on the role of a project manager. In particular, this thesis looks at a PM-agent that can manage a simulated project to design and construct a small sounding rocket.

ACKNOWLEDGEMENT

This thesis would not have been possible without the support of many people. I would like to give special thanks to Dr. Avelino Gonzalez for all your dedication and support in helping complete this thesis. Without you, I would never have known the enjoyment of research. I would also like to give thanks to Dr. Setsuo Tsuruta for the guidance and experience you provided me during my stay in Japan, without which would have impeded the quality of the thesis. Additionally, I would like to thank my family for their constant support. You were always there to provide encouragement when times were tough. Lastly, I would like to thank the UCF Honors College staff for their support in helping to complete this thesis.

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION AND BACKGROUND	1
1.1 Computer Supported Cooperative Works	1
1.2 Contextual Reasoning	2
1.3 Contextual Graphs and Context-Based Reasoning	3
CHAPTER 2 – RECENT APPROACHES TO CONTEXTUAL AWARENESS	10
CHAPTER 3 – PROBLEM DEFINITION	17
3.1 The General Problem	17
3.2 The Specific Problem	18
3.3 Hypothesis	19
3.4 Contributions of Greater Contextualization	19
3.5 Novelty, Significance and Usefulness	19
CHAPTER 4 – THE NEW USER INTERFACE	21
4.1 Introduction to the User Interface	21
4.2 The Menu Bar	24
4.2.1 The File and Help Menu	24
4.2.2 The Action Buttons	25
4.3 Starting a New Test	25
4.4 Update Current Rocket Interface	28

4.5 External Event Interface.....	28
4.6 Pause/Resume Time Progression Functionality	29
CHAPTER 5 – THE CONTEXT-DRIVEN APPROACH	30
5.1 Original Design by Gonzalez et al.	30
5.2 The New Approach	32
5.3 The Auxiliary Agents	32
5.3.1 Auxiliary Agent AA-0: The Project Initialization Agent	32
5.3.2 Auxiliary Agent AA-1: The Rocket Simulator Agent.....	33
5.3.3 Auxiliary Agent AA-2: The Scheduler Agent.....	34
5.3.4 Auxiliary Agent AA-3: The Budget Agent	36
5.3.5 Auxiliary Agent AA-4: The Conflict Resolution Agent	38
5.3.6 Auxiliary Agent AA-5: The New Design Agent	40
5.4 The New Contexts.....	40
5.4.1 Initialize Major Context.....	41
5.4.1.1 Sub-Context Contextual Graph 1 (CxG-1)	42
5.4.2 No-Initial-Design Major Context	43
5.4.3 The Splitting of the Design-Change Major Context.....	45
5.4.3.1 Bad-Design Major Context.....	45
5.4.3.2 Specification-Violation Major Context.....	46

5.4.3.3 Design-Change Major Context	47
5.4.3.3.1 Sub-Context Contextual Graph 2 (CxG-2).....	48
5.4.4 External-Event Major Context.....	50
5.4.4.1 Sub-Context Contextual Graph 3 (CxG-3)	50
5.4.5 Multiple-Simultaneous-Events Major Context.....	51
5.4.5.1 Sub-Context Contextual Graph 4 (CxG-4)	52
5.4.6 Normal Major Context.....	53
5.4.7 Impasse Major Context.....	55
5.4.7.1 Sub-Context Contextual Graph 5 (CxG-5)	56
5.5 Overall System Summary.....	57
CHAPTER 6 – TESTING.....	61
6.1 The Original Test Cases	62
6.1.1 Test Set 1	62
6.1.1.1 Test Set 1 Conclusions.....	87
6.1.2 Test Set 2	87
6.1.2.1 Test Set 2 Conclusions.....	112
6.1.3 Test Set 3	112
6.1.3.1 Test Set 3 Conclusions.....	125
6.2 The New Test Cases	125

6.2.1 Test Case with Late Design Provided Within 2 Weeks of Start.....	126
6.2.2 Text Case with Late Design Provided Within 2 Weeks of Start Conclusions	130
6.2.3 Test Cases Involving Multiple Simultaneous Events	131
6.2.4 Multiple Simultaneous Events Conclusions	159
6.2.5 Test Case Involving a Bad Initial Design	161
6.3 Ability to Easily Change the PM-Agent Because of CxBR.....	165
6.3.1 Ease of Change Conclusions	178
6.4 Summary and Conclusion of Test Results	178
CHAPTER 7 – SUMMARY, CONCLUSIONS AND FUTURE RESEARCH	181
REFERENCES	183

LIST OF FIGURES

Figure 1 - Contextual Graph (reprinted without permission from Gonzalez, 2007)	4
Figure 2 - CxBR Paradigm (reprinted without permission from Gonzalez, 2007).....	5
Figure 3 - An Active Major Context (reprinted without permission from Gonzalez, 2008b)	6
Figure 4 - CxBR Driving Example (reprinted without permission from Gonzalez et al., 2008)....	8
Figure 5 - PM-agent User Interface	22
Figure 6 - File Menu	24
Figure 7 - Help Menu.....	24
Figure 8 - New Test Window.....	25
Figure 9 - Design Window – Design Tab (Reprinted without permission, NASA 2003)	26
Figure 10 - Design Window – Materials Tab (Reprinted without permission, NASA 2003)	27
Figure 11 - The Estimated Cost and FDD Window.....	28
Figure 12 - External Event Interface.....	29
Figure 13 - Rocket Launch Phase (Reprinted without permission, NASA 2003)	34
Figure 14 - AA-5 Resign Rocket Request	40
Figure 15 - Sub-Context CxG-1.....	44
Figure 16 - Sub-Context CxG-2.....	49
Figure 17 - Sub-Context CxG-3.....	51
Figure 18 - Sub-Context CxG-4.....	53
Figure 19 - Main Screen in Running in the Normal Major Context.....	54
Figure 20 - Impasse Popup Message	55
Figure 21 - Main Screen in Impasse Mode	56

Figure 22 - Sub-Context CxG-5.....	58
Figure 23 - The PM-agent Architecture.....	60

LIST OF TABLES

Table 1 - Control System Installation and Construction Times.....	35
Table 2 - Fin/Nose Cone Material Installation and Construction Times	36
Table 3 - Multiple Stage Installation Time	36
Table 4 - Engine Installation and Construction Times	36
Table 5 - Control System Cost.....	37
Table 6 - Fin/Nose Cone Material Cost	37
Table 7 - Multiple Stage Cost	37
Table 8 - Engine Cost	37

CHAPTER 1 – INTRODUCTION AND BACKGROUND

Current advances in computer networking provide a great opportunity to effectively and efficiently manage collaborative projects in which the collaborators may be located vast distances apart from each other. This has been a problem for a long time in large commercial projects such as large scale software development, major construction projects, and manufacturing of highly complex devices (airplanes, ships, cars, etc.), where components may be designed and built by contractors all over the world,. Because of this, a need has arisen for a way to effectively manage such large scale projects in an efficient and effective fashion. This research involves delving into the field of *Computer Supported Cooperative Work* (CSCW). We describe this field of research next.

1.1 Computer Supported Cooperative Works

“The term computer supported cooperative work (CSCW) was first coined by Irene Greif and Paul M. Cashman in 1984. . .” [Wikipedia, 2010] As the name states, CSCW is a study of methods that have a computer-based agent assist in the completion of works done by several collaborators, where the complexity of the project could make its ultimate success difficult to attain. Much research has been done into the field of CSCW. Gonzalez et al. [2008a] tell us that several systems have been built to assist project managers with the help of CSCW. Gonzalez et al. in their 2008[a] paper, “We note that in the reviewed literature on project management, the *situational awareness* of the project manager is not directly mentioned. Situational awareness is a term coined by Endsley to reflect an individual’s (and hence, an agent’s) ability to be fully cognizant of the current situation faced, and the implications it has on one’s actions.” Gonzalez

et al. assert that situational awareness is a necessity in order for a project manager to be successful.

Not surprisingly, Gonzalez et al. aren't the only authors to come to this conclusion. Borges et al. [2004] experienced the same dilemma during their literature reviews. As Borges et al. state, "There has been little emphasis on the understanding of the context concept in the field of CSCW." Borges et al. further explain that when reviewing works on CSCW, they found that the issues the other authors often faced in their work were often caused by the context of the situation, although they were never acknowledged as such. Borges et al. state "We believe that most misunderstanding is caused by not explicitly recognizing and representing the notion of context and its association with other elements of [collaborative software] systems." Bores et al. contest the approaches of many others researchers in CSCW, complaining that too much of the decision making process is being left to the user, and that it is context that should be used in guiding the decision making process in CSCW.

1.2 Contextual Reasoning

Our research involves exploring the field of CSCW. We developed and describe here an agent capable of making decisions, given that the agent has been provided adequate situational awareness. In our case, the agent takes on the role of a project manager (PM). The specific problem addressed was the appropriate means to provide such situational awareness.

We decided against the user-centered methods such as that used by Hur et al. [2008]. Instead, we investigated context-driven approaches to provide an agent such knowledge. We found two successful approaches to give an agent situational awareness using context-driven approaches. The first approach used Contextual Graphs (CxGs), which were developed by

Brézillon et al. [Brézillon et al., 2002] The second approach uses a method called Context-Based Reasoning (CxBR) developed by Gonzalez et al. [Gonzalez and Ahlers, 1998] Using a combination of both of these methods, we succeeded in providing a PM-agent the situational awareness given that agent is fully contextualized. Specifically, the PM-agent is responsible for managing a given project just as a human project manager does. Our work expands upon the work done by Gonzalez et al. [Gonzalez and Ahlers, 1998] and improves upon their implementation. We proceed by giving the reader background knowledge about CxBR and CxGs.

1.3 Contextual Graphs and Context-Based Reasoning

Contextual Graphs (CxGs) give an agent situational awareness through the use of contexts. CxGs work by contextualizing situations that may occur in solving the problem. It displays this in graphical form. CxG has several components. First, there are *contextual nodes*. These are the large circles in Figure 1 below. These nodes normally serve to ask a question of either the user or the agent. Contextual nodes branch in multiple directions. Based on the path chosen, the answer to the problem at hand becomes more defined as the context of the problem becomes more finely defined.

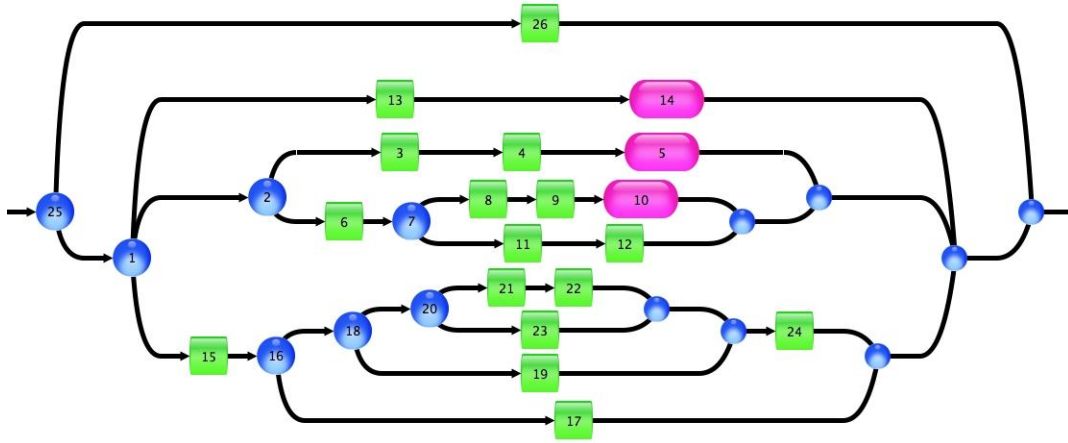


Figure 1 - Contextual Graph (reprinted without permission from Gonzalez, 2007)

Next there are *action nodes*. These are represented by squares in Figure 1. When one of these nodes is reached, some action is taken to help the user or the agent accomplish its goal. The rounded blocks in the graph are called *activity nodes*. These nodes are used for a group of actions. An activity node is best used when several actions must be taken; however, the order in which they are taken might not matter. Lastly, the small circles are *recombination nodes*. Their purpose is to re-unite the paths that previously split off from a contextual node. Because of this, the paths between a contextual node and a recombination node can also serve as a sub-graph within the larger graph.

Although the CxG paradigm seems quite well conceived and designed, CxGs seem only to work best for contexts that can be proceduralized. Specifically, CxGs set up contexts such that the actions taken are done based upon predefined procedures. Of course, tactical human do not act in a purely procedural manner. However, CxGs do have an effective role to play as a sub-system of Context-Based Reasoning (CxBR). CxBR was developed to create an agent that can be contextually aware in order to represent tactical human behavior. [Gonzalez et al., 2008a;

Gonzalez et al., 2008b; Gonzalez and Ahlers, 1998] CxBR was developed with the idea that contextual awareness could be created using a hierarchy of contexts that can be broken down into two types of contexts: descriptive contexts and control contexts. [Gonzalez et al., 2008b] “Descriptive contexts . . . serve to describe and/or define certain aspects of the environment and/or the mission.” [Gonzalez et al., 2008b] The *Mission Context*, usually just referred to as the mission, serves as the descriptive context for this system. The purpose of a mission context is not to directly control the agent, but to describe the goal and other critical information for the agent to know. The mission context is placed at the top of the context hierarchy as seen in Figure 2.

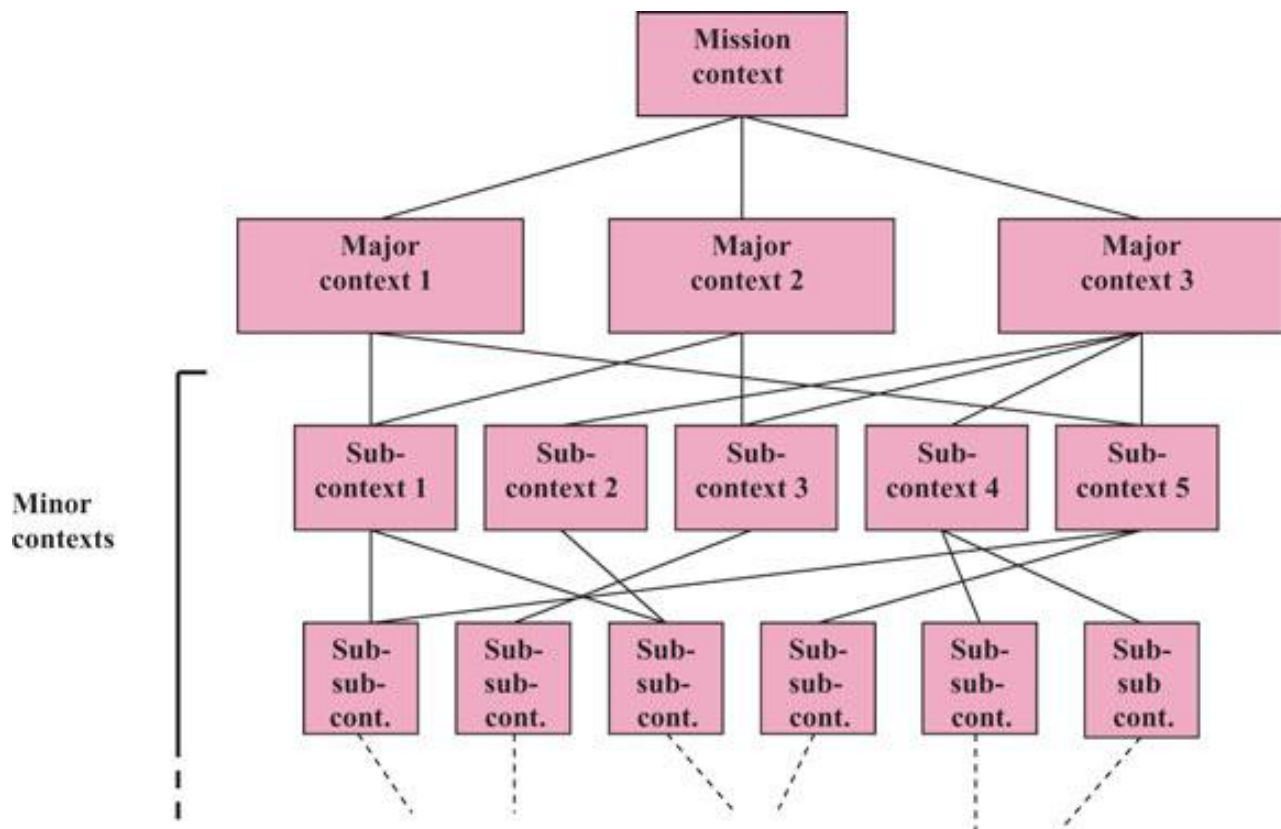


Figure 2 - CxBR Paradigm (reprinted without permission from Gonzalez, 2007)

Control contexts fulfill a greater role. “Control contexts form the core of CxBR and serve to control the agent whose behavior is being modeled.” [Gonzalez et al., 2008b] Control contexts use three types of knowledge: “. . . 1) **action knowledge** – that which prescribes how the agent is to act in the environment while under that context; 2) **transitional knowledge** – that which causes a context to know when it is no longer applicable to a new situation and a transition must be effected to another context whose action knowledge allows the agent to better handle the new situation; and 3) **declarative knowledge** – that which the agent needs to know about to carry out its assigned mission or perform its assigned task.”[Gonzalez et al., 2008a] Gonzalez et al. claim that with these three pieces of information, any situation can be accounted for. The *major contexts* of Figure 2 serve as the control contexts of the CxBR paradigm. The major contexts represent a multitude of major situations that an agent might face. A visual representation of a major context can be seen below in Figure 3. The *global fact base* shown in Figure 3 is defined later.

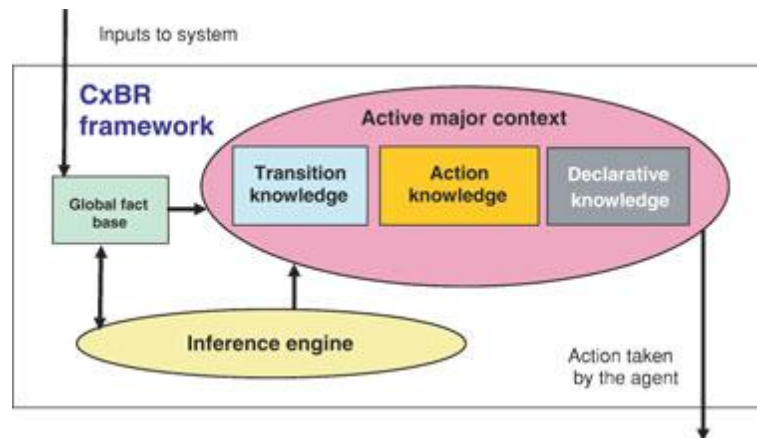


Figure 3 - An Active Major Context (reprinted without permission from Gonzalez, 2008b)

The CxBR paradigm sets up a set of major contexts that the agent may be in at any time. However, an agent can only be in one major context at a time. With that said, major contexts use

their transition knowledge to transition between major contexts, depending on the situation faced by the agent. When it is time to transition from one context to another, the current major context will be deactivated, allowing for the activation of a different major context. This can be seen as a horizontal movement within the CxBR architecture.

An example was formed using the concept of driving a car from Point A to Point B. [Gonzalez et al., 2008b] We will let the mission context be driving to Point B. While driving from Point A to Point B, we will transition between several different contexts. These situations can include, but are not limited to, **City Driving**, **Parking Lot Driving**, **Freeway Driving**, **Suburban Driving**, and **Neighborhood Driving**. Within this set of contexts, transitions between major contexts are limited based on the current context in which the agent finds itself. For instance, one cannot expect to transition from driving in their community (**Neighborhood Driving**) to driving on the freeway (**Freeway Driving**). The **Neighborhood Driving** major context restricts the agent to only contexts that are feasible. In the **Neighborhood Driving** major context, the transition rules would tell the agent that it is only capable of transitioning to **Suburban Driving**. Figure 4 below displays the transitions between these contexts in graphical form. The duration of time spent in a major context can vary, but a major context does have the ability to spend an indefinite period of time in one context without transitioning to another context.

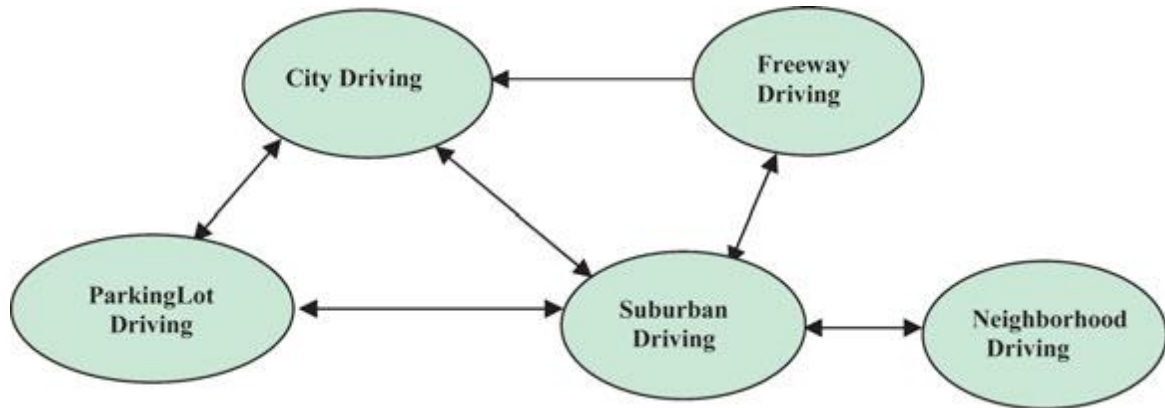


Figure 4 - CxBR Driving Example (reprinted without permission from Gonzalez et al., 2008)

Within the CxBR paradigm, there is another type of context called a *minor context*. A minor context's function is to assist with a complex task that might not be easily done by a simple function or method. Minor contexts are activated by the major contexts in the CxBR hierarchy. Minor contexts are expected to stay active only for a short period of time. However, unlike a major context, when a minor context is activated, the current major context does not become deactivated. There might be many layers of minor contexts within the CxBR hierarchy. The highest level of minor contexts is called the *sub-context*. For each sub-context, the possibility exists for sub-sub-contexts to be defined. A vertical relation is formed between major contexts, sub-contexts, sub-sub-contexts and so forth. Figure 2 shows this design. An example of sub-context might be school zone when the agent enters one. This sub-context would cause a reduction in speed limit and subsequent negative acceleration of the car agent. Be aware that during this brief example, the major context wouldn't change. However, the school zone sub-context is not applicable to all major contexts, as one would not expect a school zone in an interstate highway.

Within the CxBR architecture, a single location exists that contains all information about the system's environment. All components of the CxBR system have access to this knowledge base. Gonzalez et al. refers to this database as the *Global Fact Base* [Gonzalez et al., 2008b] “. . . [A]ny information about the internal state of the agent that is visible to the outside world is also projected on the global fact base.” [Gonzalez et al., 2008b] In addition, each agent is also given a *Local Fact Base* to reflect those things about the environment that would only be visible/known to the agent alone. To store the knowledge of each and every agent within the global fact base be unrealistic as well as inefficient. Therefore, each agent is given a local memory in which to store knowledge relevant only to it.

CxBR has been used in other applications as well, including the implementation of a submarine warfare simulation [Gonzalez and Ahlers, 1998] and an automobile driver going from point A to point B in a virtual city. [Gonzalez et al., 2008b]

This brings us to Chapter 2 where we discuss the current state of the art in contextual awareness. Following this, we continue into Chapter 3 where the problem is defined. Chapter 4 describes the new user interface. Chapter 5 then follows with our approach. Chapter 6 describes the methods in which we test our approach. Lastly, Chapter 7 summarizes and concludes our approach as well as discusses future research to be done.

CHAPTER 2 – RECENT APPROACHES TO CONTEXTUAL AWARENESS

Researchers have begun to find new ways to make efficient and effective use of CSCW. Then the idea of contextual awareness came along. Gonzalez et al. [2008a, b], Gonzalez and Ahlers [1998], Borges et al. [2004], Hu and Liu [2007], Kulkarni and Tripathi [2010], Hur et al. [2008], Mostéfaoui and Brézillon [2004], Nguyen et al. [2008], Brézillon et al. [2002] and Gonzalez [2007] all discuss the importance contextual awareness in some way or form. This is evidence that contextual awareness has become such an important factor in artificial intelligence. These authors have realized that it wasn't until programs began to become contextualized that this contextual awareness could be achieved. The authors talk about how any and every task has the ability to be contextualized. [Gonzalez et al., 2008a][Gonzalez et al., 2008b][Gonzalez and Ahlers, 1998][Borges et al., 2004][Hu and Liu, 2007][Kulkarni and Tripathi, 2010][Hur et al., 2008][Mostéfaoui and Brézillon, 2004][Nguyen et al., 2008][Brézillon et al., 2002][Gonzalez, 2007] We therefore assert that contextual awareness is an important factor in the design of a PM-agent.

However, just how and how much does one contextualize? Of course, this is where authors tend to differ. What we specifically want to know is how much contextualization is needed to provide an agent with tactical human behavior. This is especially pertinent in our research where we are trying to make such an agent. Specifically, we are looking at the design of an agent which can take on the role of a project manager.

Hur et al. [2008] explain that contexts should be centered around the user. They state that a “[c]ontext is information that represents the situations of entities related to the user.” With the

use of the user-centered contexts and the implementation of hierarchical ontology, Hur et al. explain how they are able to create an infrastructure that allows them both flexibility and efficiency in their program. Hur et al. explain that by using their user-aspect approach, one is able to eliminate any problem faced in an efficient manner. Through user-centered contexts, Hur et al. describe an implementation of a Relational Database system which creates a full layout of each user's aspects contextually. However, we disagree with their usage of contexts. We feel that contexts should not so much focus on the user, but they should focus on the conditions the user (or the agent) experiences.

As stated before in Chapter 1, Borges et al. [2004] explain that contextualization has been used incorrectly by many researchers in CSCW. Borges et al. state that most designs leave too much of the decision making up to the user instead of having the contexts guide the decision process. Our research shows the reader a means to properly use contexts in CSCW in order to bring a program situational awareness. Borges et al. also suggest that there should be multiple levels of context, and that there shouldn't just be contexts for each individual, but contexts for groups as well. We agree with the views this paper presented. We acknowledge that a program's decision making process should be led by the context of the situation. We also agree that by using a hierarchy of contexts, one is able to better provide an agent with situational awareness.

Brézillon et al. [2002] described the development of a system to monitor incidents that may occur on the Paris subway line. Brézillon et al. used contextual graphs (CxGs) as the framework to provide their incident management system with contextual-awareness. Mostéfaoui and Brézillon [2004] also use CxGs to design a context-based security system that has situational awareness. Although both these systems provided their agents with situational awareness, we

realize both implementations focused on proceduralized contexts. Specifically, proceduralized contexts are those which follow predefined acyclic processes. However, not all situations are naturally procedural in nature making proceduralization of such situations immensely difficult. Therefore, this system would fail the needs of our system. In order to create a PM-agent that acts with tactical human behavior, the agent must be able to handle all contexts. Additionally, CxG are acyclic, having only one beginning and one end. How does a PM-agent run a long term collaborative project with an acyclic design? As far as we could determine, this was impossible. A PM-agent needs to have a design in which its life can last months or years. A constant looping through a CxG would be very inefficient.

Gonzalez and Ahlers [1998] assert that tactical knowledge is needed in order to have Autonomous Intelligent Platforms (AIP's) act as if they had human intelligence. They describe AIP's as agents whose purpose is to represent either a person or a group of people controlling a physical platform such as a car, a plane etc. Their paper also tells of how situational awareness is key to an agent being able to gain the tactical knowledge needed in order for it to act realistically intelligent. Gonzalez and Ahlers use Context-based Reasoning (CxBR) as their method to provide AIP's with situational awareness. Specifically, they created a submarine warfare simulator that was able to account for many situations that could occur during wartime. The only problem with this implementation is that it doesn't account for all situations. A PM-agent, such as the subject of this research, would need to account for anything that might happen over its lifetime.

In 2008[b], Gonzalez et al. describes another project in which they created an agent that exhibited tactical human behavior. By using Context-Based Reasoning, Gonzalez et al. claim

they can successfully represent human tactical behavior. By using context transition knowledge, an agent is able to figure out when the current context is no longer valid and switch to a more appropriate one. Through the use of a hierarchy of contexts, Gonzalez et al. state that every situation can be accounted for, and the agent will be able to properly transition between contexts to act most appropriately. Nevertheless, the authors still don't show how much contextualization needs to go into such a system in order to account for every situation in an efficient and effective manner.

Gonzalez et al. [2008a] proposed the idea of using contextualization to form an agent that acts as a manager of a collaborative project. They discuss their interest in using such an agent in CSCW. Throughout this paper, Gonzalez et al. emphasize that for an agent to be successfully implemented in CSCW, it must have some form of situational awareness. The authors assert that once an agent has situational awareness, any such large scale project can be handled by the agent. Specifically, the authors investigated the design and construction of a small sounding rocket that was used as the test domain for their context-driven project manager. Gonzalez et al. took an approach that combined both CxBR and CxGs into their paradigm so as to cover both environmental situations and individual situations. This design was a good start in the usage of contextualization in the field of CSCW, but it did not address the issue of how much contextualization is needed. With the use of an open source rocket simulator from NASA [2003], Gonzalez et al. [2008a] were able to successfully develop a paradigm that would account for most situations that might occur in the design, manufacture and installation, of a rocket, given specific initial conditions from a user. Their design consisted of four Major Contexts: **Normal**, **Design-Change**, **External-Event**, and **Impasse**. A project will always begin with the **Normal**

Major Context. From there, the project manager agent can decide in which Major Context it should be next. If the project is proceeding according to plan, the project manager stays in the **Normal** Major Context. However, should an event happen, such as a change to the design by either a human or computer design agent, the project manager will take action and switch to the appropriate alternative Major Context. In this case, it would be the **Design-Change** Major Context, which deals with design changes, checks that the new design works properly and that the given design stays within the Mission Context. The **External-Event** Major Context deals with all external events. However, the research of Gonzalez et al. has limited this context's functionality to only deal with changes to schedule and cost, but not design, as this is addressed by the **Design-Change** Major Context. The last Major Context is the **Impasse** Major Context. This Major Context is only selected by the project manager agent as a last resort. If the project becomes over-constrained and the new events and/or new design cause the delivery date, the final cost or both to be exceeded[,]” [Gonzalez et al., 2008a] then the **Impasse** Major Context becomes activated.

Under the context-driven design conceived by Gonzalez et al. [2008a], there are five auxiliary agents (AA) and three contextual graphs used as sub-contexts. These sub-contexts are called upon by the PM-agent. Depending upon which major context is active for the PM-agent, the PM-agent determines which sub-context to call. AA-0 is in charge of the official schedule and cost. AA-1 is the rocket simulator that returns a Boolean value describing whether or not the rocket is capable of flight to the specified altitude carrying the specified payload. AA-2 is the cost agent of the system. It calculates all costs that the PM-agent might incur during the building and installation of the rocket. AA-3 is the PM-agent's scheduler. AA-3 computes the length of

time a rocket design will take. Lastly, AA-4 handles the conflict resolution load provided by the system.

The contextual graphs are used as a means to add contextualization by use of the procedural nature of certain actions the PM-agent takes. For example, after a design change occurs, the PM-agent needs to make sure that not just the design implements the rocket specification, but must also check that the new design is affordable and the rocket will be able to be completed in time. However, first, the PM-agent should check to make sure that the specified altitude is met while carrying the specified payload. It is only if this condition is met that the budget and schedule should be checked. By proceduralizing the activation of these actions to be taken by the PM-agent, we contextualize the actions taken by the PM-agent, thereby optimizing how the PM-agent handles the situation at hand, i.e. the PM-agent only analyzes budget and schedule implications of the design change if there is a valid design. Once in the implied context of a valid design from a physical standpoint, the PM-agent will check whether the design is affordable and the rocket of that design can be completed within the time specified. Based on the results found by the contextual graph, the PM-agent gains the knowledge necessary to know to which major context to transition next.

While the system works, this agent may not be able to handle the processing load of future problems. Moreover, this system does not account for every situation that might occur during the design of a small sounding rocket. More contextualization needs to be included in the system. Specific examples where the system failed include an initial design that doesn't comply with the mission context and an engine design being received too late. Although Gonzalez et al. claimed the current infrastructure was able to handle these cases, they nevertheless explain that

these cases were not handled properly. It is the purpose of this research to investigate and address these problems.

Lastly, this system does not account for multiple situations that might occur at once. What would happen if a design change to the rocket occurred while at the same time, the cost of a material rose? How would this system know which major context to transition to? Unexpected results are likely to occur. Our work also addresses this issue.

CHAPTER 3 – PROBLEM DEFINITION

In this section we describe the overall (general) problem and the specific problem faced in this research.

3.1 The General Problem

As stated before, how does one manage large collaborative projects in which the collaborators may be located vast distances apart from each other? This has been a problem for a long time in large commercial projects such as large scale software development, major construction projects, and the manufacturing of highly complex devices (airplanes, ships, cars, etc.). At times, components for these projects may be designed and built by different contractors all over the world. For example, in the design of a rocket, an American company might need to order parts that were designed outside of the U.S. The engines might come from multiple companies located in China. The rocket body may be built in South Korea. The control system could be imported from France. Then, in the end, when all the parts arrive in the U.S., they have to be shipped to a Texas plant to be assembled into the rocket. Finally, the completed rocket has to be transported to Florida where it is launched.

The complexity of such projects has given rise to the need for a way to effectively manage such large scale projects in an efficient and effective fashion. With the help of Computer Supported Cooperative Works (CSCW), this task can be greatly simplified. But, how does one design an agent that can manage a project as a human would?

3.2 The Specific Problem

The specific purpose of this research is to define a method for creating such PM-agents as described in section 3.1 that can optimally and efficiently react to a range of predefined situations. From previous research done by Gonzalez et al. [Gonzalez et al., 2008a] [Gonzalez et al., 2008b] [Gonzalez and Ahlers, 1998], it has been determined that situational awareness is the key component that an agent must have to act as a tactical human. Because of this, Gonzalez developed the paradigm known as Context-Based Reasoning (CxBR). [Gonzalez, 2007] Gonzalez explains that CxBR by its nature imbues situational awareness in agents.

However, in one of Gonzalez et al.'s [2008a] most recent projects using CxBR, they explain that their constraint-based agent AA-4 was prone to overstrain. This leads to the conclusion that either AA-4 was not sufficiently contextualized to ameliorate their problem or AA-4 was just too heavily relied upon. We take the stance that AA-4 was too heavily relied upon. Therefore, if the PM-agent was given enough contextualization, problems that arose could have been handled by the PM-agent in a more efficient manner instead of just relying on one agent to be the project's savior. This is because when we increase contextualization, the PM-agent will be able to disperse the load normally given to AA-4 to other contexts, thereby decreasing the PM-agent's reliance on it. Additionally, Gonzalez et al. recognize that the PM-agent they created could not properly account for situations such as the late delivery of design specifications and a bad initial design. We realize that again, their system was not contextualized enough. They only had four major contexts: **Normal**, **Design-Change**, **External-Event**, and **Impasse**. This span of contexts is too narrow to provide full situational awareness to the PM-agent. Lastly, Gonzalez et al. failed to account for situations where multiple major context changes could occur at the same

time during the project. Again, we hypothesize that our approach of greater contextualization of the PM-agent can solve the problem.

3.3 Hypothesis

The current PM-agent system is under-contextualized. By increasing the system's contextualization, we can account for all the previously mentioned situations as well as other possibly unforeseen situations that may occur.

3.4 Contributions of Greater Contextualization

Our research will provide the following contributions to the current state of the art:

- An enhancement to the legacy system that will give agents full situational awareness that the original systems did not provide.
- A PM-agent that can account for more situations than its predecessor.
- The PM-agent will also be able to account for multiple situations that may occur simultaneously.
- Lastly, the PM-agent acts as a fully functional tactical human with the purpose of serving as a project manager or a cooperative project run over the internet.

3.5 Novelty, Significance and Usefulness

Creating a PM-agent is not a difficult task if one was able to recognize all the contexts of a situation. Nevertheless, no author seems to have built a PM-agent which is fully able to efficiently and effectively account for all situations that might occur over the lifetime of a large scale CSCW project. Therefore, this is the reason why a system that uses a greater level of contextualization is needed. Although complete coverage of all possible situations is impossible

from a practical standpoint, a higher level of contextualization will provide the situational awareness an agent needs to handle most of them. Using the CxBR paradigm, we will be able to add the contextualization needed to the PM-agent so that it will be able to encompass almost any situation.

With the implementation of a PM-agent that can successfully manage a simulated project to design and construct a small sounding rocket, we could apply the same approach to other cooperative works that require managing by a PM-agent. It is because any project can be contextualized that we realize that with enough contextual knowledge, a PM-agent has the ability to act with tactical human-likeness.

CHAPTER 4 – THE NEW USER INTERFACE

In the original system created by Gonzalez et al., no user interface existed, let alone a graphical one. The system was designed as such: there was an input file containing the parameters of the rocket and there was an input file containing the CxBR contextual layout for the CxBR engine. Anytime a test was run, the events that occurred had to be hardcoded into the system. Then, in order to run the test, the program had to be run from command line. Lastly, the original system only output test results using print statements. The original system printed large amounts of data to the screen. When using the command line to display output from a system, only so much text can be stored in the command line's buffer. Because of this, it would have been very easy to overflow the buffer. In effect, the user would have lost part of the test results. Also, test results would then have had to be copied from the command line to a file using copy and paste for each test, if one wanted a test record. In turn, the user would have had to run the program in a modified version of command line in order to see and record all the results.

This is neither a very efficient nor effective way to do testing of a complex system. In addition, no user who wasn't a skilled coder and familiar with the system could have run test cases. For this reason, we created our own user interface in order to make it possible for any user to run our system. In this chapter, we will discuss all the features of our user interface as well as the basics of its use.

4.1 Introduction to the User Interface

For a system, we went with a graphical user interface since this would be an easy way for a user to interact with the system. We begin by describing the main screen. When a user opens the system, the user interface will appear as shown in Figure 5. The main screen was built with 4

sections: Schedule/Budget, Body/Nose Cone, Control System, and Engine. Within each of these sections, different parameters known to the PM-agent are shown. Values in the Schedule/Budget section contain the parameters of the Mission Context as well as the current cost and estimated

The screenshot shows a software window titled "CxBR Status Window" with a menu bar containing "File", "Help", "Update Current Rocket", "Insert External Event", and "Pause/Resume Time Progression". The main area is titled "Context-Based Reasoning Rocket Design Simulation" and is divided into four sections:

- Schedule/Budget:**
 - Current Day:
 - Event Number:
 - Maximum Budget in MYen:
 - Estimated Cost in MYen:
 - Maximum Deadline in Days:
 - Estimated FDD in Days:
- Body/Nose Cone:**
 - Body Diameter in cm:
 - Body Length in cm:
 - Nose Cone Material:
 - Nose Cone Density in g/cm³:
 - Nose Cone Length in cm:
 - Parachute Diameter in cm:
- Control System:**
 - Number of Fins:
 - Fin Material:
 - Fin Length in cm:
 - Fin Width in cm:
 - Fin Height in cm:
 - Fin Density in g/cm³:
 - Control System:
- Engine:**
 - Number of Stages:
 - Stage One Engine:
 - Stage Two Engine:
 - Stage Three Engine:
 - Recovery System and Payload Mass in g:

Below these sections, the text "Design Validity Undetermined" is displayed. A status box shows "Status: Test Uninitialized". At the bottom right, there are five buttons: "Previous Event", "Next Event", "Previous Day", "Next Day", and "Current Event".

Figure 5 - PM-agent User Interface

delivery date of the system. The three sections just contain the design parameters of the rocket. Since no test has started yet, these values are all currently blank. However, once a test begins, all

of the values displayed will be populated with the corresponding information contained in the Global Fact Base. This can be seen in Figure 19 in Chapter 5.

Next, on the bottom left side of the window there is a Design Validity text box and a status window. The Design Validity text box shows whether or not a design was able to reach the specified altitude with the specified payload. Depending on whether or not the design is able to meet these requirements, the sign will display Design Valid in green text or Invalid Design in red text. In addition, if no design is received when a mission begins, the text will show No Initial Design in red text instead. The status window, displayed below the Design Validity text box shows all actions taken by the PM-agent. Such actions include a transition rule being checked or an action rule being activated. Again, by referring to Figure 19 in Chapter 5, one can see an example of the output.

On the right side of the screen, there are five buttons. The Previous Event button and Next Event button will allow the user to display through the events that have occurred so far. If an event is the last event in a day and the Next Event button is clicked, the system will roll over to the first event of the next day. If an event is the first event in a day and the Previous Event button is clicked, the system will roll over to the last event of the previous day. Every day will contain one or more events. Under these buttons are a Next Day and a Previous Day button. When the Next Day button is clicked, the system will display the first event of the next day. If the Previous Day button is clicked, the first event of the previous day will be displayed on the screen. Lastly, there is a Current Event button. When this button is clicked, the current event will be displayed on the screen.

4.2 The Menu Bar

On the top of the user interface, a toolbar was created for easy interaction with the system. Section 4.2.1 will introduce the File and Help menu, Section 4.2.2 introduces the toolbar's Action Buttons, Section 4.3 discusses the process of starting a new test, Section 4.4 discusses the Update Current Rocket interface, Section 4.5 discusses the Insert External Event interface, and Section 4.6 discusses the Pause/Resume Time Progression functionality.

4.2.1 The File and Help Menu

When the File button is clicked, the user is given the option to start a new test (this can occur during the middle of a test as well), update the current rocket design, pause/resume the time progression of the system, or exit the system. (Note: Once the system begins, every $\frac{1}{2}$ second, the system will progress to the next day if in the **Normal** Major Context. A detailed description of our system's **Normal** Major Context is described in Chapter 5. A screenshot of the File Menu can be seen in Figure 6 below.



Figure 6 - File Menu

The Help button contains a menu with 1 button. The button is labeled About CxBR and gives a brief statement about the CxBR concept. The expanded Help menu is shown in Figure 7 below.



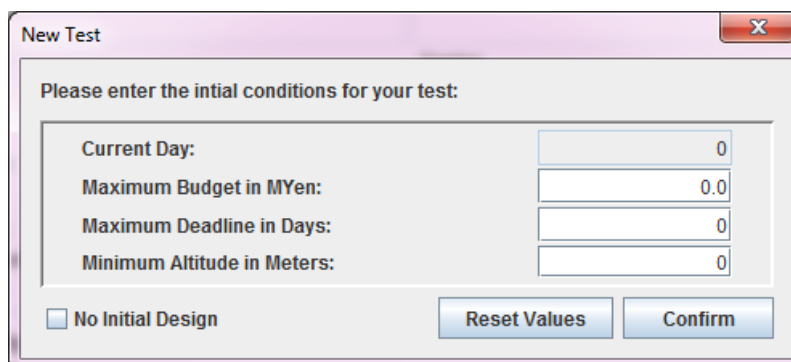
Figure 7 - Help Menu

4.2.2 The Action Buttons

On the toolbar, there are three actions buttons in addition to the File and Help Menu. The Update Current Rocket Button provides the exact same functionality as the Update Current Rocket button in the File Menu. Clicking on it allows the user to update the rocket design. The Insert External Event allows the user to submit one external event into the system. Lastly, the Pause/Resume Time Progression button provides the same functionality as the Pause/Resume Time Progression button in the File Menu.

4.3 Starting a New Test

When the system first starts, in order to start a new test, one must click File→New Test. This will bring up a new test pop-up window allowing the user to specify the Mission Context described in Chapter 3. In Figure 8 below, the New Test Window is shown. As one can see, the user has the ability to specify the Maximum Budget, Maximum Deadline, and Minimum Altitude



Please enter the initial conditions for your test:	
Current Day:	0
Maximum Budget in MYen:	0.0
Maximum Deadline in Days:	0
Minimum Altitude in Meters:	0

☐ No Initial Design

Figure 8 - New Test Window

the rocket must reach. In addition, in the bottom left of the window, there is a No Initial Design checkbox. This allows the user to inform the PM-agent that there is no initial design being submitted. Once the Confirm button is clicked, the Mission Context will be submitted to the PM-agent. As a note, the reason why the payload is not specified in the new test window is because

the payload must be specified to the NASA rocket simulator directly. A detailed description as to why is given in Chapter 5, Section 5.2.1.

Next, if the No Initial Design checkbox is left unchecked, the NASA rocket simulator will appear, allowing the user to submit a design to the PM-agent after which the system will transition back to the main window. However, if the checkbox is checked, the system will skip the attempt to receive an initial design and transitions back to the main window. At the main window, the PM-agent will begin progressing through the days as described in Section 4.2.1.

The Design Window of the simulator has two tabs, the Design Tab and the Materials Tab. The Design Tab, which allows the user to select geometric parameters of the rocket, the number of stages of the rocket, the number of fins of the rocket, and the control system used by the rocket. The Design Tab is shown below in Figure 9.

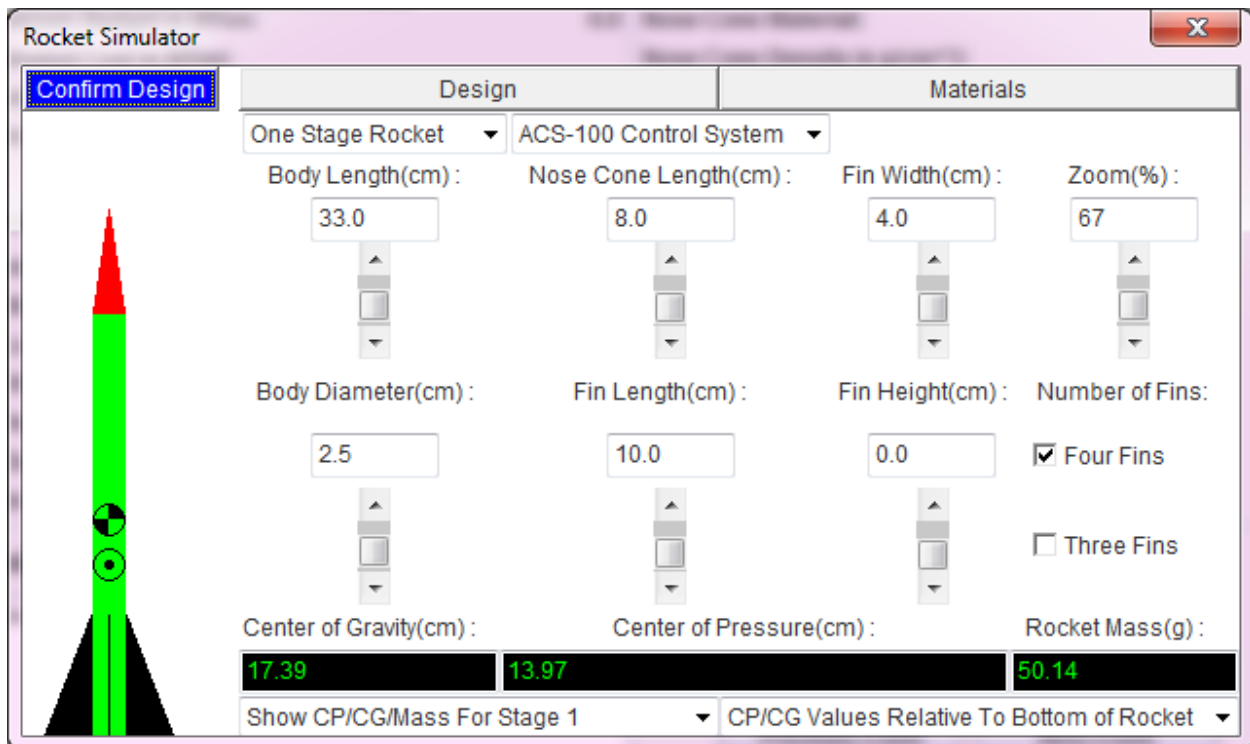


Figure 9 - Design Window – Design Tab (Reprinted without permission, NASA 2003)

The second tab of the Design Window is the Materials Tab, shown in Figure 10 below. In this tab, the user has the ability to select the material for the nose cone and fins, the engines to be used for each stage of the rocket, the payload mass, and the parachute diameter.

Rocket Simulator

Confirm Design | **Design** | **Materials**

Nose Cone Material: Balsa | Fin Material: 1/8" Balsa

Custom Nose Cone Density(g/cubic cm): | Custom Fin Density(g/cubic cm):

Stage One Engine: A8-3 | Stage Two Engine: | Stage Three Engine:

Recovery System and Payload Mass(g): 15.0

Parachute Diameter: 1 foot (30 cm) | Drag Coefficient: 0.7

Center of Gravity(cm): 17.39 | Center of Pressure(cm): 13.97 | Rocket Mass(g): 50.14

Show CP/CG/Mass For Stage 1 | CP/CG Values Relative To Bottom of Rocket

Figure 10 - Design Window – Materials Tab (Reprinted without permission, NASA 2003)

As a note, we have made three modifications to the rocket simulator's design interface. First, to allow for the selection of a control system during design selection, a dropdown menu option was added to the Design Tab of NASA's system. Second, in the original simulator, a launch button existed in the design interface. We have modified this to a submit button so that the design phase can submit the design to the PM-agent without launching the rocket. Third, an estimated cost and FDD window has been added to the display to show the effects on the cost and schedule based on changes made to the design. This allows for easier rocket design by the user. Figure 11 displays a screenshot of the Estimated Cost and FDD Window.

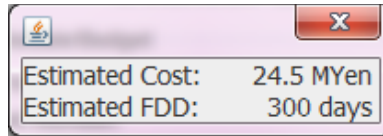


Figure 11 - The Estimated Cost and FDD Window

4.4 Update Current Rocket Interface

Clicking on one of the Update Current Rocket buttons activates the Update Current Rocket interface. Once clicked, the NASA rocket simulator will reappear with the values entered from the previous valid design. Here, the user creates a design to submit a design to the PM-agent. If no initial design was received at the beginning of the test, the submitted design will be treated as a late initial design for the system. Additionally, should one of the Update Current Rocket buttons be clicked by accident, as long as none of the rocket design values are changed, clicking the submit design button will cancel the update current rocket design operation.

4.5 External Event Interface

Clicking the Insert External Event button allows the user to submit an external event to the PM-agent. Any number of external events may be submitted to the PM-agent; however, each external event must be submitted individually. There are four types of external events available: 1) a cost increase external event, 2) an engine delivery delay external event, 3) a control system delivery delay external event, and 4) any other delay which affects the overall FDD. Below in Figure 12 one can see a screenshot of the external event interface. To submit an external event user must do is enter the value of the external event they are reporting and click the corresponding checkbox.

External Event

Please Enter A External Event and Select the Corresponding Check Box

Added Cost in MYen:	0
Engine Delivery Delay in Days:	0
Control System Delivery Delay in Days:	0
Other Delay in Days:	0

☐ Cost Increase ☐ Engine Delay
☐ Control System Delay ☐ Other Delay

Reset Values Confirm

Figure 12 - External Event Interface

4.6 Pause/Resume Time Progression Functionality

The last major components of the system are the Pause/Resume Time Progression buttons. As mentioned before, the day will automatically progress every $\frac{1}{2}$ second. Clicking either button in the File Menu or on the toolbar will cause the test to pause or resume, depending on whether the system is running or already paused. In addition, any time an external event or new design is reported to the PM-agent, the system will pause its progression. In order to resume the progression of the system, one of the Pause/Resume Time Progression must be clicked. This is done to allow for easy insertion of multiple events in one day.

CHAPTER 5 – THE CONTEXT-DRIVEN APPROACH

We created a PM-agent that can manage a simulated project to design and construct a small sounding rocket. We build upon the original design created by Gonzalez et al. [2007] [2008a]. Gonzalez et al.'s original design used the CxBR and GxGs to provide their PM-agent with situational awareness. Given a final delivery date, a specific budget, a minimum altitude to obtain, and a minimum payload to carry, the PM-agent is expected to ensure a client's specification is met by staying within those bounds. Explicit use is made of NASA's open source rocket simulator. [NASA, 2003] NASA's rocket simulator has the ability to simulate the launch of a rocket and determine whether the launch was successful or not given a specific design and a specification of altitude and payload.

5.1 Original Design by Gonzalez et al.

The design of Gonzalez et al. consists of four Major Contexts: **Normal** Major Context, **Design-Change** Major Context, **External-Event** Major Context, and **Impasse** Major Context. The **Normal** Major Context is designed as the starting point for a given mission context. As long as the rocket design is within the client's specification, the PM-agent will stay in the **Normal** Major Context. Nonetheless, there is always a chance something can occur that will cause the rocket's design to cease to meet the specification, such as external events or design changes during the course of a mission. Because of this, transition rules are set such that the PM-agent can transition to different contexts to account for different situations. For instance, if a price increase occurred to one of the parts, the PM-agent would transition to the **External-Event** Major Context where it would check to see the implications of the external event. If the external

event did not put the project out of specification, the PM-agent would then transition back to the **Normal** Major Context.

The last context available to the PM-agent is the **Impasse** Major Context. Should the corpus not be able to take the corrective actions necessary to cause the design to fall back into the specification laid out by the Mission Context, the **Impasse** Major Context is activated. Because the system has become over-constrained, the PM-agent asks for support from an outside human entity. If the human is able to take the appropriate corrective actions, then the PM-agent transitions back to the **Normal** Major Context. Should the outside human deem that there is no solution to the problem, the project is cancelled.

There are currently five auxiliary agents (AA) and three contextual graphs in the original system by Gonzalez et al. These act as the sub-contexts for their system. AA-0 is in charge of the official schedule and cost. AA-1 is the NASA rocket simulator which determines whether or not the rocket is capable of flight and able to reach the specified altitude carrying the required payload. AA-2 is the cost agent of the system and responsible for determining the cost of the current design. AA-3 is the scheduler agent. AA-3 is responsible for determining the expected FDD. Lastly, AA-4 is a conflict resolution agent. This agent tries its best to take corrective action to conflicts the current design might have with the Mission Context without changing the design. Such actions currently include reducing cost by reducing overhead staff and paying overtime in order to speed up production. A detailed description of the previous system can be found in the paper “Using Contexts to Control a Collaborative Process” by Gonzalez [2007].

5.2 The New Approach

Our approach takes what was done by Gonzalez et al. as described in Section 5.1 and improves upon it. We expand upon the current list of major contexts in order to provide a more robust system. Our expectation is that by increasing the contextualization of the system, we can encompass many situations that the PM-agent might face as well as reduce the system's overdependence on the use of AA-4. Lastly, through the increased contextualization of our system, the new PM-agent's design encompasses the ability to account for multiple events occurring simultaneously. We now begin by discussing our implementation of the Auxiliary Agents followed by descriptions of the Major- and Sub-Contexts of the PM-agent.

5.3 The Auxiliary Agents

This section describes the Auxiliary Agents called upon by the PM-agent. The new design implements six Auxiliary Agents, one of which is a human. We begin with the discussion of Auxiliary Agent AA-0: The Projection Initialization Agent. We then follow with a discussion of Auxiliary Agent AA-1: The Rocket Simulator Agent, Auxiliary Agent AA-2: The Scheduler Agent, Auxiliary Agent AA-3: The Budget Agent, and Auxiliary Agent AA-4 the Conflict-Resolution Agent. Lastly, we describe Auxiliary Agent AA-5, the New Design Agent.

5.3.1 Auxiliary Agent AA-0: The Project Initialization Agent

Whenever a project begins, a specification must be defined. Our new AA-0 has been given this task. (Note: There is no correlation between the previous AA-0's design and the new one's except for the name.) AA-0 is a new agent whose function is to create and maintain the PM-agent simulation. AA-0 has the ability to start a project, where it will obtain the Mission

Context containing the maximum budget, maximum FDD, and the minimum altitude to be reached, as well as end the currently running mission. In Figure 8, a screen shot can be seen from the GUI interface of the PM-agent simulation where AA-0 has been called to obtain the new Mission Context. As noted earlier, for this system, the payload must be directly specified to AA-1, the Rocket Simulation Auxiliary Agent. This is because the rocket simulation aspect of AA-1 was written by NASA and direct injection of the payload value into AA-1 was not possible in a practical way.

Lastly, since values cannot be directly injected into the NASA rocket simulator, part of the NASA rocket simulator is used to allow the user, acting as both upper management and the design agent, to configure a rocket's design. In addition, three modifications have been made to the rocket simulator's design interface. First, to allow for the selection of a control system during design selection, a dropdown menu option was added to NASA's system. Second, in the original simulator, a launch button existed in the design interface. We have modified this to a submit button so that the design phase can submit the design to the PM-agent without launching the simulated rocket. The launch phase has been assigned to AA-1, which is described in Section 5.2.2. Third, an estimated cost and FDD window has been added to the display to show the effects on the cost and schedule based on changes made to the design. This allows for easier rocket design by the user. Figure 9, Figure 10, and Figure 11 from Chapter 4 display screenshots taken from design phase portion of AA-0.

5.3.2 Auxiliary Agent AA-1: The Rocket Simulator Agent

Just as in the previous system, AA-1 is based on NASA's rocket simulator. AA-1 is given the responsibility of determining whether or not the rocket is able to meet the specified altitude

while lifting a given payload. AA-1 returns that a launch is successful as long as the rocket doesn't crash and the previous conditions are met. The NASA rocket simulator's launch phase is used to determine whether the rocket crashed and the max altitude reached the given specified value. Figure 13 below shows the launch phase of AA-1 using the NASA rocket simulator.

Too much white space here, Brian. Manipulate the text paragraphs to fill it up

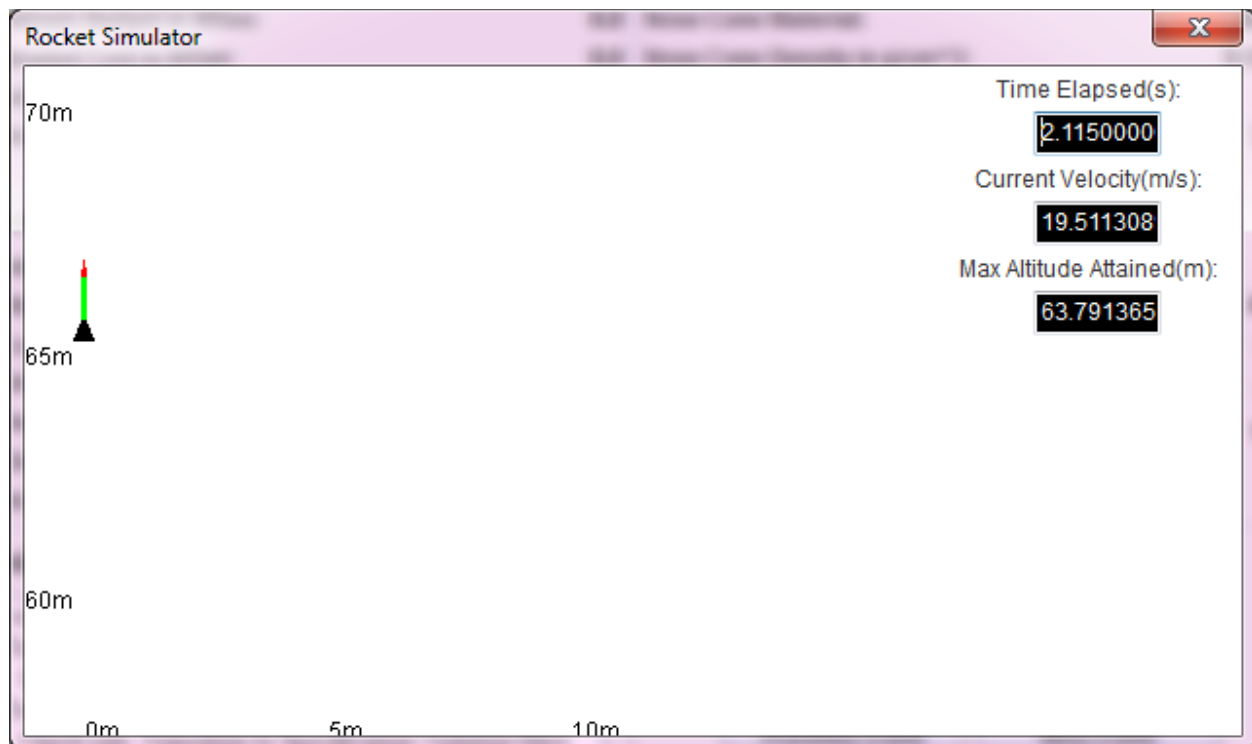


Figure 13 - Rocket Launch Phase (Reprinted without permission, NASA 2003)

5.3.3 Auxiliary Agent AA-2: The Scheduler Agent

The scheduler agent has the same responsibilities as in the previous system. AA-2 is responsible for determining the FDD of the project whenever it is called. AA-2 will return an integer representing the FDD in days. In addition, we would like to note that the previous system only did not seem to take into account the Installation and Construction time of the nose cone. For no known reason, they assumed the installation and construction time of the fins covered the

nosecone. Therefore, our FDD will be larger than the previous system. AA-2 calculates the schedule using the following assumptions:

1. All construction is done concurrently.
2. Construction/Delivery time of all the parts is set to that of the part that will take the longest to arrive.
3. Installation cannot occur until all parts are delivered.
4. All parts are installed sequentially.
5. New parts that are ordered while the project is running have their construction time calculated based on the current day.
6. If a new part is ordered, installation will still occur based on the previous assumptions.
7. If we have an n-stage engine and one of the engines is replaced, the additional schedule cost for multiple stages is not added again. The same goes for the control system. Only if the number of stages changes does the schedule see an affect.
8. $FDD = (\text{Construction/Delivery of all parts}) + \sum (\text{Installation time of each part})$
9. The installation/construction times of the parts are given in Table 1, Table 2, Table 3, and Table 4.

CS Name	No. of stages	Delivery Time
ACS-100	1	6 months
ACS-200	2	6 Months
ACS-300	3	6 months

Table 1 - Control System Installation and Construction Times

Material	Hollow Plastic	Balsa wood	Plastic
Install time	2 months	1.5 months	1 month
Construct time	2 months	1.5 months	1 month

Table 2 - Fin/Nose Cone Material Installation and Construction Times

Structure Name	No. of Stages	Installation (months)
One SR	1	0
Two SR	2	1
Three SR	3	3

Table 3 - Multiple Stage Installation Time

Engine Name	Construct Time in Months	Install Time in Months
1/2A6-2	0.5	0.5
A8-3	2	1
A8-5	2	1
B6-2	4	2
B6-4	4	2
B6-6	4	2
C6-3	6	3
C6-5	6	3
C6-7	6	3
D12-3	8	3
D12-5	8	3
D12-7	8	4
D12-0	8	4
C6-0	6	3
B6-0	4	2
A8-0	2	1

Table 4 - Engine Installation and Construction Times

5.3.4 Auxiliary Agent AA-3: The Budget Agent

AA-3 has also been given the same responsibilities as it was given in the previous system. In the previous system, the documentation claimed the Budget was calculated by adding the cost of all the parts together. If a part(s) was changed, the cost would just be added by the part(s) would just be added to the total cost. Our system followed these criteria. However, for some reason, in the previous system, it failed to take into account the cost of both the nosecone and fin material. The cost was calculated only using the fin material. Therefore, our AA-3 returns higher

costs than calculated by the previous system, as we are taking into account the cost of all the parts. We also take into account the following criteria:

1. If a Part A is removed for a Part B, and then on another date replaced back with Part A, we assume the original Part A was disposed of.
2. If we have an n-stage rocket and one of the engines is replaced, the additional cost for multiple stages is not added to the budget again. The same goes for the control system. Only if the number of stages changes does the cost see an affect.
3. The cost of the parts is given in Table 5, Table 6, Table 7, and Table 8.

CS Name	No. of stages	Cost
ACS-100	1	3.5 MYen
ACS-200	2	4 MYen
ACS-300	3	4.5 MYen

Table 5 - Control System Cost

Material	Hollow Plastic	Balsa wood	Plastic
Cost	20 MYen	10 MYen	5 MYen

Table 6 - Fin/Nose Cone Material Cost

Structure Name	No. of Stages	Cost
One Stage	1	0 MYen
Two Stage	2	1 MYen
Three Stage	3	5 MYen

Table 7 - Multiple Stage Cost

Engine Name	Cost	Engine Name	Cost
1/2A6-2	0.5 MYen	C6-3	9 MYen
A8-3	1 MYen	C6-5	10 MYen
A8-5	1.5 MYen	C6-7	8 MYen
A8-0	1.5MYen	C6-0	10MYen
B6-2	3 MYen	D12-3	20 MYen
B6-4	4 MYen	D12-5	18 MYen
B6-6	5 MYen	D12-7	12 MYen
B6-0	5MYen	D12-0	20MYen

Table 8 - Engine Cost

5.3.5 Auxiliary Agent AA-4: The Conflict Resolution Agent

The AA-4 agent is designed to assist the PM-agent when a specification becomes violated. In the original system, AA-4 was claimed to be a constraint-based agent. However, since AA-4 has only a limited number of constraints, we constructed our AA-4 to act as a quasi-constraint-based agent via the use of conditional statements. Therefore, although no constraint-based reasoning engine was used, our AA-4 does act as a constraint based system.

AA-4 contains two corrective action operators, Corrective Action Operator 1 and Corrective Action Operator 2. Below we have listed descriptions of the corrective action operators along with the constraints necessary for their activation. In addition, we have provided the condition statement equivalents used to implement the constraints. Should AA-4 be activated and neither corrective action operator is able to provide a solution, AA-4 will declare that an impasse has occurred.

- CA1: Pay overtime at a cost of 30KYen per day as long as the constraints below are met.
 - Constraints to be met:
 - Constraint 1: The FDD must exceed the maximum time allotted in the specification.
 - Constraint 2: Overtime cannot be paid for more than 30 days.
 - Constraint 3: Paying overtime cannot cause the budget to be exceeded.
 - Conditional Statement Equivalent:

- *If* the FDD exceeds the maximum time allotted to complete the project, overtime has not been paid for more than 30 days and the cost of the necessary overtime does not cause the budget to be exceeded, *Then* pay overtime.
- CA2: Reduce overhead as by either 10% or 5% as long as the constraints below are met.
 - Constraints to be met:
 - Constraint 1: The cost of the project must exceed the specified budget.
 - Constraint 2: Employment reduction cannot occur more than 2 times.
 - Constraint 3: First time employment reduction is by 10%.
 - Constraint 4: Second time employment reduction is by 5%.
 - Constraint 5: Overtime must not have just been paid.
 - Conditional Statement Equivalent:
 - *If* the cost of the project exceeds the specified budget, employment reduction has not occurred twice, and overtime has not just been paid, *Then*:
 - *If* overhead has been reduced 0 times before, *Then* reduce employment by 10%.
 - *If* budget specification not met, *Then* reduce employment by an additional 5%

- *Else If* overhead has been reduced once before, *Then* reduce overhead by 5%.

5.3.6 Auxiliary Agent AA-5: The New Design Agent

The AA-5 agent is designed to request a new design on behalf of the PM-agent. Since agents can be either human or artificial, we have decided to use a human to act as AA-5. Therefore, upon activation of AA-5, a request is made of the human running the project to submit a new design as seen in Figure 14 below.

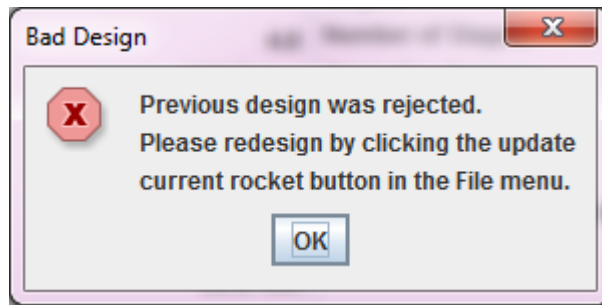


Figure 14 - AA-5 Resign Rocket Request

5.4 The New Contexts

This section describes the Major and Sub-Contexts used by the PM-agent. There are nine Major Contexts and five Sub-Contexts. We begin by describing the **Initialize** Major Context. Following this, we describe the **No-Initial-Design** Major Context followed by the splitting of the **Design-Change** Major Context from the original system developed by Gonzalez et al. This subsection describes the **Bad-Design** Major Context, the **Specification-Violation** Major Context, and the **Design-Change** Major Context. Afterward, we describe the **External-Event** Major Context, the **Multiple-Simultaneous-Events** Major Context, and the **Normal** Major Context.

Lastly we discuss the **Impasse** Major Context. Throughout these sections the Sub-Context Contextual Graphs 1-5 are described within the Major Context they are called in.

5.4.1 Initialize Major Context

When we initially delved into the reconstruction of the system implemented by Gonzalez et al., we'd found it in our best interest to change their approach on how to initialize their system. In the system they created, the PM-agent always began by starting in the **Normal** Major Context. From here the system would determine what state to place itself in. The normal course of action to be taken by the **Normal** Major context was usually to then call upon the original AA-0 (a new AA-0 has been created but has a different design) to initialize the system. We have determined this is not a suitable way to start a project as many times in project management, the project does not start with perfect conditions. For this reason, we developed the **Initialize** Major Context.

As it is known, all projects must start somewhere. The **Initialize** Major Context is designed to determine just where that "somewhere" is. Of course, as just stated earlier, sometimes projects don't always begin as we expect. Therefore, when creating the **Initialize** Major Context, three considerations were taken into account:

1. Does the initial design meet specification from the start?
2. If it doesn't, why not? Is it because of a specification violation or a bad design?
3. Is there an initial design? It is not uncommon for a design to be received late.

With these questions answered, the PM-agent would then be able to begin the project with an understanding of the situation at hand, giving the PM-agent proper situational awareness. Once these questions have been answered, then the appropriate Transition Rule will be activated and

allow the PM-agent to continue to next appropriate context. Below, we have listed a detailed description of the **Initialize** Major Context.

Major Context Name: *Initialize*

Action Functions:

- Determine if initial design is within the specifications of the mission context by activation of appropriate action rule.

Transition Rules

- If initial design is not valid, then activate ***Bad-Design Major Context***
- If initial design not within specification, then activate ***Specification-Violation Major Context***
- If there is not initial design, then activate ***No-Initial-Design Major Context***
- If initial design is valid and is within spec., then activate ***Normal Major Context***

Action Rules:

- Activate CxG-1 to determine if there is an initial design and if such design is valid. A valid design constitutes as one in which the rocket doesn't crash, meets the minimum altitude requirements while carrying the specified payload, and is able to meet the given specification.

Sub-Contexts

- CxG-1

5.4.1.1 Sub-Context Contextual Graph 1 (CxG-1)

Originally, when the **Normal** Major Context was designed as the beginning context for all projects, AA-0 was called upon to handle the task of initialization. However, AA-0 was not properly suited to handle this task. This is because AA-0 was designed to always assume a project starts normally and not to anticipate inadequate initial conditions. A more contextualized approach was needed to be able to provide this level of information to the PM-agent. Therefore, we created CxG-1 to handle this task. Below we have laid out the details of CxG-1's design. A graphical representation of the logic for CxG-1 is shown in Figure 15.

Sub-Context Name: CxG-1

Action Functions:

- Determines the next Major Context transition by activating the CxG
 - See logic in Figure 15
- Determines if there is no initial design
 - If no initial design, *No Initial Design* is announced
- Determines if the initial design is a valid design
 - If initial design is invalid, *Bad Design* announced
- Determines if the initial design is within spec.
 - If initial design is out of spec., *Specification Violation* announced

Action Rules:

- Activates AA-1 to determine design feasibility
- Activates AA-2 to determine schedule impact
- Activates AA-3 to attempt to determine cost impact

5.4.2 No-Initial-Design Major Context

As mentioned before, there is a chance that a mission can start off without an initial design. Being that such a situation is a context all its own, we deemed it only natural to make it its own Major Context. In a situation where no initial design exists, a PM-agent has two options: 1) wait for a design or 2) cancel the project. For our design, upper management is given two weeks to come up with a design if one is not provided at the project's outset. Should 14 days pass and no initial design is received, the PM-agent will transition to the **Impasse** Major Context, described in Section 5.2.13, where it will request one last time for upper management to submit an initial design or cancel the project. Below is an outline of the **No-Initial-Design** Major Context.

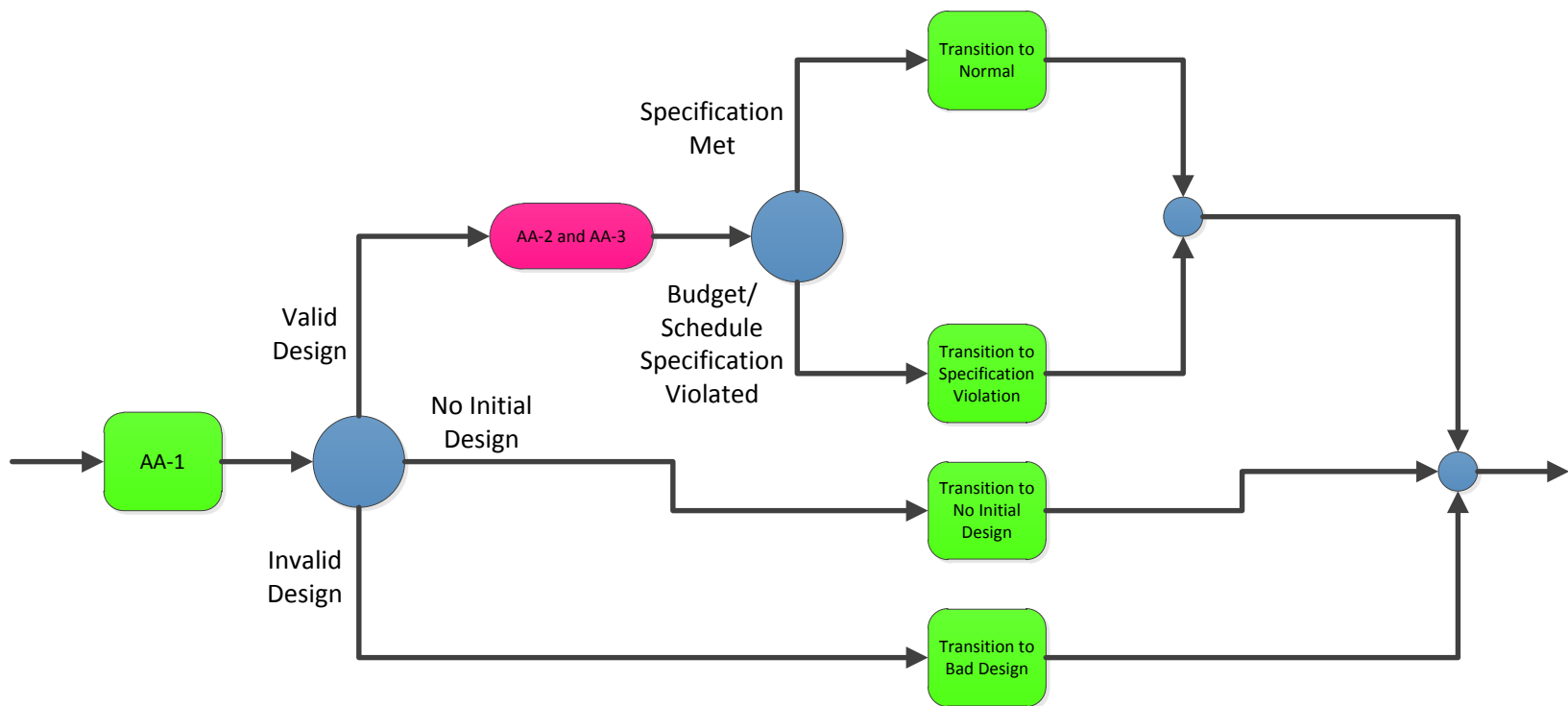


Figure 15 - Sub-Context CxG-1

Major Context Name: *No-Initial-Design*

Action Functions:

- Wait up to two weeks for GFB to retrieve an Initial Design

Transition Rules

- If initial design is received, then activate *Initialize Major Context*
- If no initial design received after two weeks, then activate *Impasse Major Context*

Action Rules:

- None

Sub-Contexts

- None

5.4.3 The Splitting of the Design-Change Major Context

In the previous system, the **Design-Change** Major Context was responsible for all situations relating to changes in the design and the reactions the PM-agent would take. However, doing such limits the PM's situational awareness since all outcomes would be handled by one context. For this reason, we have split the **Design** Major Context's responsibilities by adding two major contexts. These contexts are the **Bad-Design** Major Context and the **Specification-Violation** Major Context.

5.4.3.1 *Bad-Design Major Context*

The **Bad-Design** Major Context was designed to handle any situation in which a design is given which the PM-agent deems needs to be replaced. Such situations include a design which cannot cause the rocket to reach the specified altitude with the defined payload, and a new design that violated the budget and/or schedule specification parameters by more than 10%. When the **Bad-Design** Major Context becomes the PM's active context, AA-5 will then be activated to request a new design with which the rocket will be able to meet the required altitude, while carrying the specified payload and satisfying specification. In addition, the **Bad-Design** Major Context was also designed as a means to remove some of the strain from AA-4, the conflict

resolution agent described earlier. By having the **Bad-Design** Major Context, the PM-agent now has the situational awareness necessary to know that the problem can't be solved using simple conflict resolution. It now knows the scope of the problem, thereby allowing it to recognize that the optimal solution is to just scrap the new design that was just received, not order the new parts, and try a new design all together. Thereby, we are using AA-5, activated using this major context's action rule, to take on some of AA-4's responsibility. Below we outline this major context's definition.

Major Context Name: *Bad-Design*

Action Functions:

- Requests a new design to present to the PM-agent by activation of the appropriate action rule.

Transition Rules

- When new design received, activate *Design-Change Major Context*

Action Rules:

- Activates AA-5 to request a new design which will be valid and meet specification.

Sub-Contexts

- None

5.4.3.2 Specification-Violation Major Context

The **Specification-Violation** Major Context was designed to handle situations in which a design change and/or external event(s) caused the design to no longer meet specification. Originally however, this major context was designed to only handle specification violations that occurred as a result of a design change. Nonetheless, since external events can also cause a project to violate specification, it seemed only natural that the same context would handle such events. (Note: We would like here to emphasize that it is because of the CxBR paradigm, which uses contexts as its means situational awareness, that such an easy pairing was able to be made with the pre-existing **Specification-Violation** Major Context and the **External-Event** Major

Context, described in Section 5.2.10. As done before, we will now list an outline of this major context.

Major Context Name: *Specification-Violation*

Action Functions:

- Tries to resolve specification violations by activation of the appropriate action rule.

Transition Rules

- If project is normal, activate *Normal Major Context*
- If project is over-constrained, activate *Impasse Major Context*

Action Rules:

- Activates AA-4 to attempt conflict resolution to put project into spec.

Sub-Contexts

- None

5.4.3.3 Design-Change Major Context

Now that we have the **Bad-Design** Major Context and the **Specification-Violation** Major Context defined, combined with the new **Design-Change** Major Context, the PM-agent can handle any situation imposed upon it by a change in the design. Via the use of its sub-context, CxG-2, described in Section 5.2.9.3.1, a transition state will be announced to the PM-agent to allow it to know whether the PM-agent should transition to the **Bad-Design** Major Context, the **Specification-Violation** Major Context, or back to the **Normal** Major Context. Below we list an outline of the **Design-Change** Major Context.

Major Context Name: *Design-Change*

Action Functions:

- Activates sub-context CxG-2 through appropriate action rule

Transition Rules

- If CxG-2 reports the project condition is normal, then activate *Normal Major Context*
- If CxG-2 reports the project's new design is invalid, then activate *Bad-Design Major Context*
- If CxG-2 reports the project's design doesn't fall with spec., activate *Specification-Violation Major Context*

Action Rules:

- Activates sub-context CxG-2 to determine if the design change is feasible

Sub-Contexts

- CxG-2

5.4.3.3.1 Sub-Context Contextual Graph 2 (CxG-2)

CxG-2 is designed to determine the next state of the **Design-Change** Major Context. As stated earlier, when a design change occurs, there are multiple situations that can occur. These situations include a design change which causes the rocket not to be able to reach the specified altitude while carrying the defined payload, designated as an invalid design. Other such situations includes a design change in which the specification is violated by either less than or greater than 10% of the budget and/or schedule, and a design change in which the design is valid and is within specification. Below, we list an outline of CxG-2 as well as provide a graphical representation of CxG-2 in Figure 16.

Sub-Context Name: CxG-2**Action Functions:**

- Determines the next Major Context transition by activating the CxG
 - See logic in Figure 16
- Determines if the design is a valid design and within spec.
 - If new design is invalid, *Bad Design* announced
- Determines if the new cost is within budget
 - If new design makes the budget exceed specified cost or delivery by less than 10%, *Specification Violation* announced
 - If new design puts budget out of specification by more than 10%, *Bad Design* announced
- Determines if the new schedule is within specification
 - If new design puts schedule out of specification by less than 10%, *Specification Violation* announced
 - If new design puts schedule out of specification by more than 10%, *Bad Design* announced

Action Rules:

- Activates AA-1 to determine design feasibility
- Activates AA-2 to determine schedule impact
- Activates AA-3 to attempt to determine cost impact

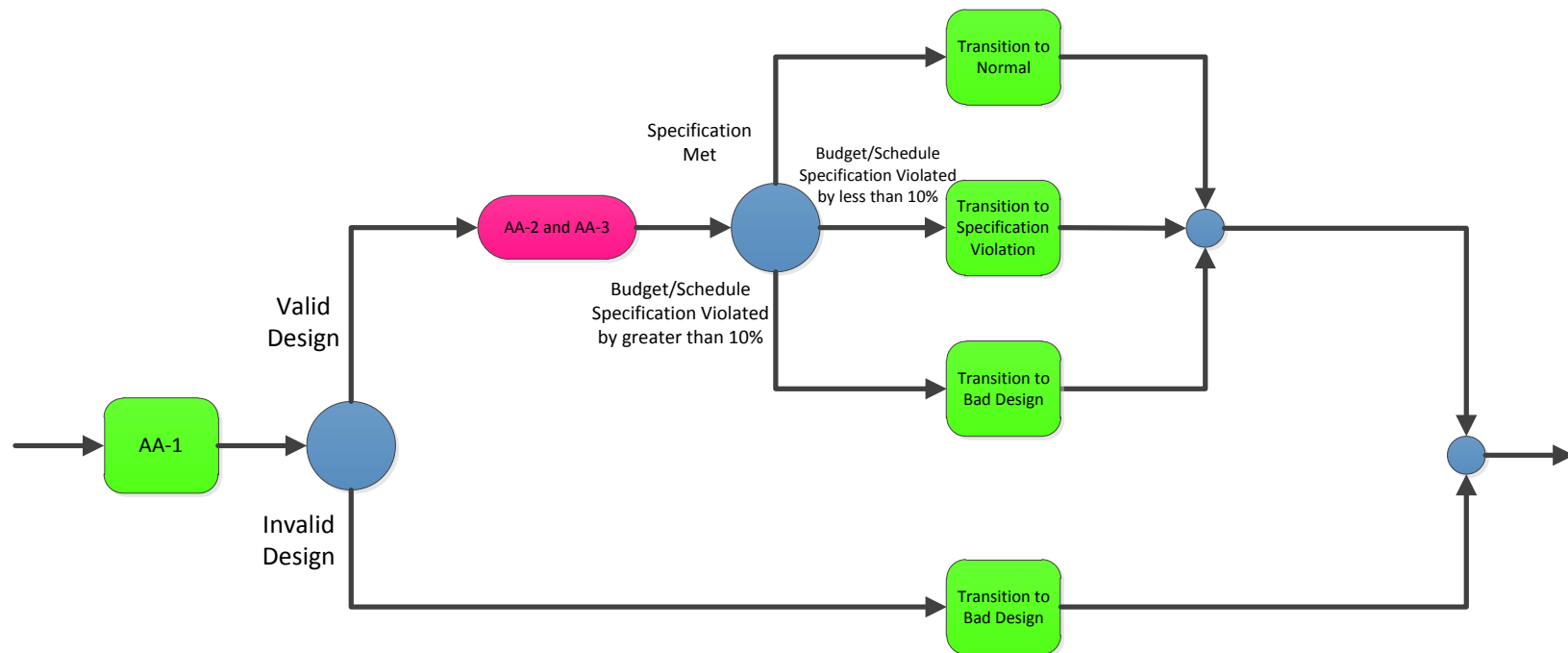


Figure 16 - Sub-Context CxG-2

5.4.4 External-Event Major Context

Whenever the PM-agent leaves specification because of an event that causes either an increase in cost or the length of the schedule, such as a delay or a price increase, the **External-Event** Major Context is activated. From here, the PM-agent would determine whether to call AA-4 to do conflict resolution or send the PM-agent back to the **Normal** Major Context. However, with the preexisting major context of **Specification-Violation**, it was illogical to call upon AA-4 directly to handle specification violation situations because there already is a context to handle this. In addition, by using the **Specification-Violation** Major Context to handle specification violations, we can now have the **External-Event** Major Context handle multiple external events that occur simultaneously. Therefore, the **External-Event** Major Context is now designed only to either transition the PM-agent back to the **Normal** Major Context or to the previously defined **Specification-Violation** Major Context described in Section 5.2.9.2, thereby increasing the abilities and efficiency of the PM-agent. Via the use of the **External-Event** Major Context's sub-context, CxG-3, the PM-agent can determine whether or not the specification has been violated, in which case the PM-agent will just transition to the **Specification-Violation** Major Context. Listed below is an outline of the **External-Event** Major Context.

5.4.4.1 Sub-Context Contextual Graph 3 (CxG-3)

CxG-3 is designed to announce to the PM-agent which context to transition to next while in the **External-Event** Major Context. Depending on whether or not the external events cause a violation of the specification, either a transition to Specification Violation or Normal will be announced. Below we have outlined CxG-3 and provide a graphical representation in Figure 17.

Sub-Context Name: CxG-3

Action Functions:

- Determines the next Major Context transition by activating the CxG
 - See logic in Figure 17
- Determines if the external event has a schedule impact
 - If new schedule is out spec., *Specification Violation* announced
- Determines if the external event has a cost impact
 - If new cost is out spec., *Specification Violation* announced

Action Rules:

- Activates AA-2 to attempt to determine schedule impact
- Activates AA-3 to attempt to determine cost impact

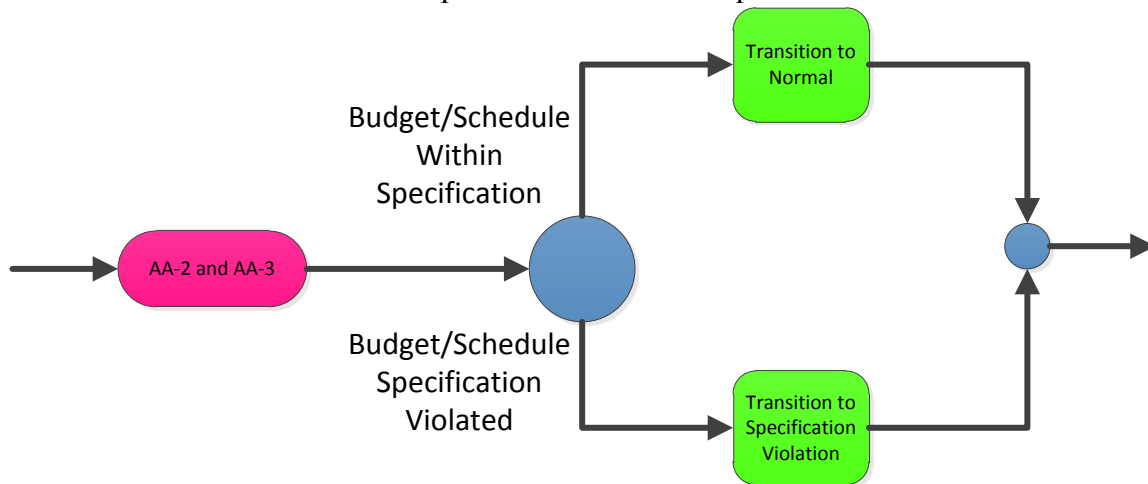


Figure 17 - Sub-Context CxG-3

5.4.5 Multiple-Simultaneous-Events Major Context

Every project manager knows that there is always the possibility that multiple events can occur at the same time that could possibly jeopardize the success of their mission. The previous system that was developed was not able to account for such occurrences. This prompted the creation of the **Multiple-Simultaneous-Events** Major Context. The **Multiple-Simultaneous-Events** Major Context allows for the PM-agent to be able to account for any number of events that occur whether they are Design Changes or External Events. Through the activation of CxG-4, the PM-agent is able to determine how to handle the multiple simultaneous events. Based on the results CxG-4 announces, the PM-agent will either transition to the **Design-Change** Major

Context or the **External-Event** Major Context. Below we have outlines the **Multiple-Simultaneous-Events** Major Context.

Major Context Name: *Multiple-Simulations-Events* (Design and External)

Action Functions:

- Activates sub-context CxG-4 through appropriate action rule

Transition Rules

- If CxG-4 there was a change to the design, then transition to ***Design-Change Major Context***
- If CxG-4 reports there were only multiple external events, then transition to ***External-Event Major Context***

Action Rules:

- Activates sub-context CxG-4 to determine whether to transition to ***Design-Change Major Context*** or ***External-Event Major Context*** depending on the cause of the multiple events

Sub-Contexts

- CxG-4

5.4.5.1 Sub-Context Contextual Graph 4 (CxG-4)

As just described, CxG-4 is called by the **Multiple-Simultaneous-Events** Major Context. When multiple events happen at the same time, if any of the events happens to be a design change, CxG-4 will announce a the PM-agent needs to transition to the **Design-Change** Major Context since it is capable handling changes to the budget and schedule in its process of determining a new design's feasibility. However, if only external events occur, the PM-agent will be informed to transition to the **External-Event** Major Context, since the feasibility of the design will not be altered. Below is an outline of CxG-4 as well as graphical representation, labeled as Figure 18.

Sub-Context Name: CxG-4

Action Functions:

- Determines the next Major Context transition by activating the CxG
 - See logic in Figure 18
- Determines if one of the events that has occurred is a change in design
 - If there is a change in design, then announce *Design Change*
- Determines if there has been multiple external events

- If there have just been multiple External Events, then announce *External Event*

Action Rules:

- Check GFB for event occurrences

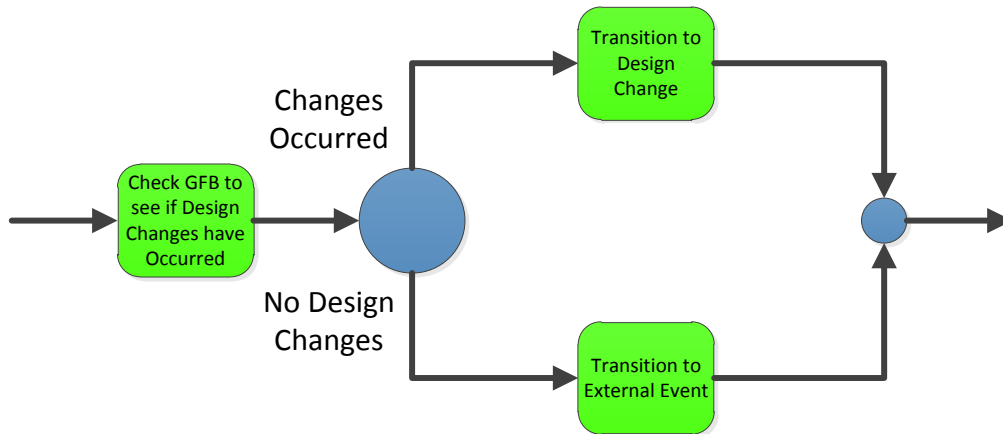


Figure 18 - Sub-Context CxG-4

5.4.6 Normal Major Context

The **Normal** Major Context is the context in which the PM-agent stays when the project is within specification. This major context has only one action rule, check the GFB to see if an event has occurred. While in the **Normal** Major Context, the user has the ability to insert External Events and Design Changes via the user interface. Once one of these two events is inserted, the system will pause allowing for the insertion of multiple events if needed. Once the user is ready for the system to resume, the resume button can be hit, and the system will continue. Figure 19 shows a screen shot of the user interface. Based on the events that occur, the system will transition to the appropriate context to handle the event(s) accordingly. If the project finishes successfully in the **Normal** Major Context, the message window will show the text “Project Has Completed Successfully” and an output file containing data from every event will be outputted. (Note, this is not an action function of the **Normal** Major Context) An outline of the **Normal** Major Context” is listed below.

Major Context Name: *Normal*

Action Functions:

- Sample GFB for any event occurrences

Transition Rules

- If a design change occurs, then activate ***Design-Change Major Context***
- If one new external event occurs, then activate ***External-Event Major Context***
- If multiple simultaneous events occur (design change and external event), then activate ***Multiple-Simultaneous-Events Major Context***

Action Rules:

- None

Sub-Contexts

- None

The screenshot shows a software window titled "CxBR Status Window" with a menu bar (File, Help, Update Current Rocket, Insert External Event, Pause/Resume Time Progression). The main area is titled "Context-Based Reasoning Rocket Design Simulation". It contains two tables of parameters:

Schedule/Budget	
Current Day:	5
Event Number:	1
Maximum Budget in MYen:	550
Estimated Cost in MYen:	24.5
Maximum Deadline in Days:	550
Estimated FDD in Days:	300

Body/Nose Cone	
Body Diameter in cm:	2.5
Body Length in cm:	33.0
Nose Cone Material:	Balsa
Nose Cone Density in g/cm^3:	0.1281
Nose Cone Length in cm:	8.0
Parachute Diameter in cm:	30.0

Control System	
Number of Fins:	4.0
Fin Material:	1/8" Balsa
Fin Length in cm:	10.0
Fin Width in cm:	4.0
Fin Height in cm:	0.0
Fin Density in g/cm^3:	0.04067175
Control System:	ACS-100

Engine	
Number of Stages:	1
Stage One Engine:	A8-3
Stage Two Engine:	N/A
Stage Three Engine:	N/A
Recovery System and Payload Mass in g:	15.0

Below the tables, it says "Design Valid". A scrollable log shows the following text:

```
* Active SubContext: null
* Evaluating rules
* Transition to Design Change
* Check rule: Transition to Design Change false
* Transition to External Event
* Check rule: Transition to External Event false
* Transition to Multiple Simultaneous Events
* Check rule: Transition to Multiple Simultaneous Events false
* Evaluating rules
* Sample GFB
* Check rule: Sample GFB true
Normal Action Function
```

At the bottom right, there are five buttons: "Previous Event", "Next Event", "Previous Day", "Next Day", and "Current Event".

Figure 19 - Main Screen in Running in the Normal Major Context

5.4.7 Impasse Major Context

The **Impasse** Major Context is designed to act very similarly as to that of the original system. Should the **Impasse** Major Context be called, CxG-5 is activated to contact upper management to determine the best course of action. From the **Impasse** Major Context, one of two things can happen. Either upper management fixes the problems with the project, allowing the project to transition to the **Normal** Major Context, or the project is cancelled. The user interface adds the option to change the projects specification when an impasse occurs. Figure 20 shows a screenshot of what happens when an impasse occurs. Figure 21 shows a screenshot of the main screen of the user interface after an impasse has occurred. Below is an outline of the **Impasse** Major Context.

Major Context Name: *Impasse*

Action Functions:

- Request information from external agents (upper management)
- Receive response and determine its effect
- Activates CxG-5 through appropriate action rule

Transition Rules

- If CxG-5 reports that the projects condition is normal, then activate *Normal*
- If CxG-5 reports management has canceled the mission, then STOP.

Action Rules:

- Activates CxG-5 to obtain management's decision on the project

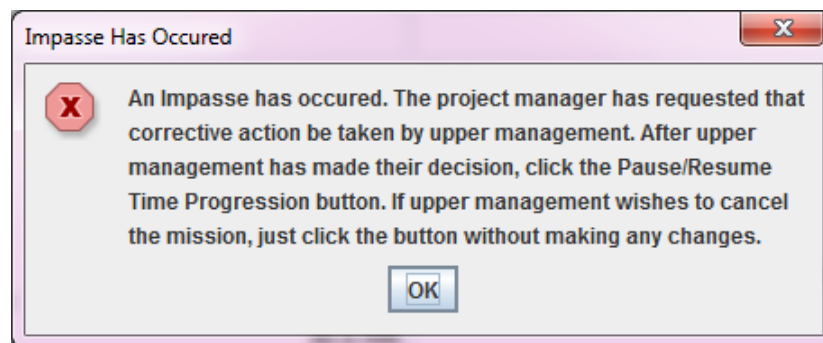


Figure 20 - Impasse Popup Message

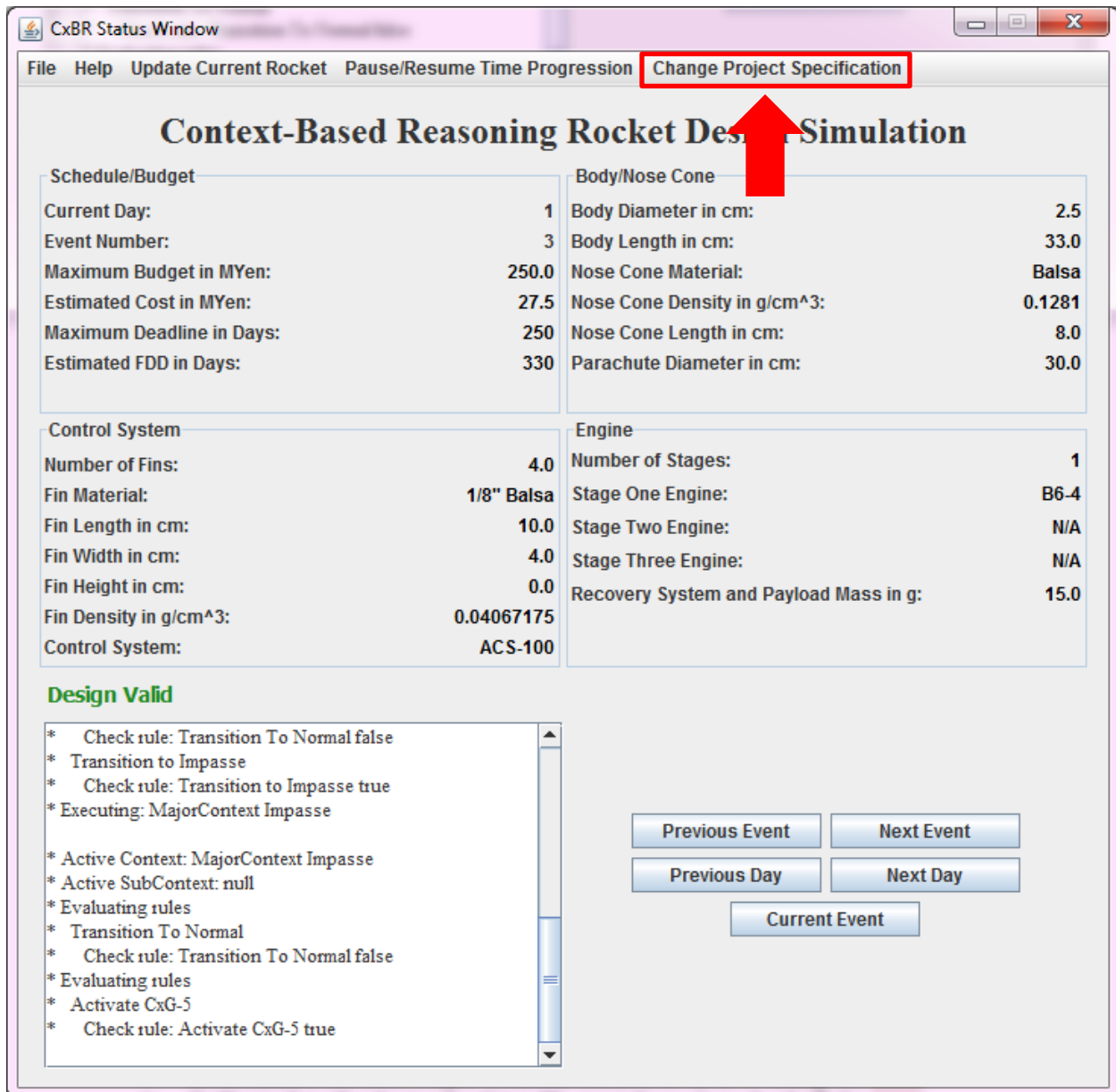


Figure 21 - Main Screen in Impasse Mode

5.4.7.1 Sub-Context Contextual Graph 5 (CxG-5)

CxG-5 is based on the original CxG-3 developed in the previous system described by Gonzalez et al. [2008a] and Gonzalez [2007]. The main difference between our CxG and their original one is that we check to make sure modifications made by upper management actually solve the problem. In the previous system, their CxG would have assumed that upper

management gave corrections that solved the problem and simply transitioned to the **Normal** Major Context. Nonetheless, as this may not always be the case, we have our CxG check to make that the project is back within specification before transition to the **Normal** Major Context takes place. Should management provide inadequate changes, the project will be cancelled. Below we outline CxG-5. Figure 22 displays a graphical representation of CxG-5.

Sub-Context Name: CxG-5

Action Functions:

- Determines the next Major Context transition by activating the CxG
- Collects data for design and its effect on the cost and schedule
- Requests a change on mission constraints from human superior
- Receives and implements changes in mission constraints.
- If changes relieve over-constrained situation, transitions to *Normal*.
- If management decision is negative or not sufficient to relieve over-constrained situation, simulation ends
- See logic in Figure 22

Action Rules:

- Contact is made to management for changes
- Activates AA-1 to determine design feasibility based on changes
- Activates AA-2 to attempt to determine schedule impact based on changes
- Activates AA-3 to attempt to determine cost impact based on changes

5.5 Overall System Summary

Chapter 5 gives a review of all the Major- and Sub-Contexts of the PM-agent. By increasing the contextualization of the PM-agent, our system is now able to handle situations which the original system could not, in addition to the situations the original system could. Such situations include no initial design, a bad initial design, and multiple simultaneous events.

Furthermore, with the addition of the **Bad-Design** Major Context and CxG-2, our system relieves some of the burden placed upon AA-4. This is done by having CxG-2 determine when a specification violation is too great for the **Specification-Violation** Major Context, and in turn

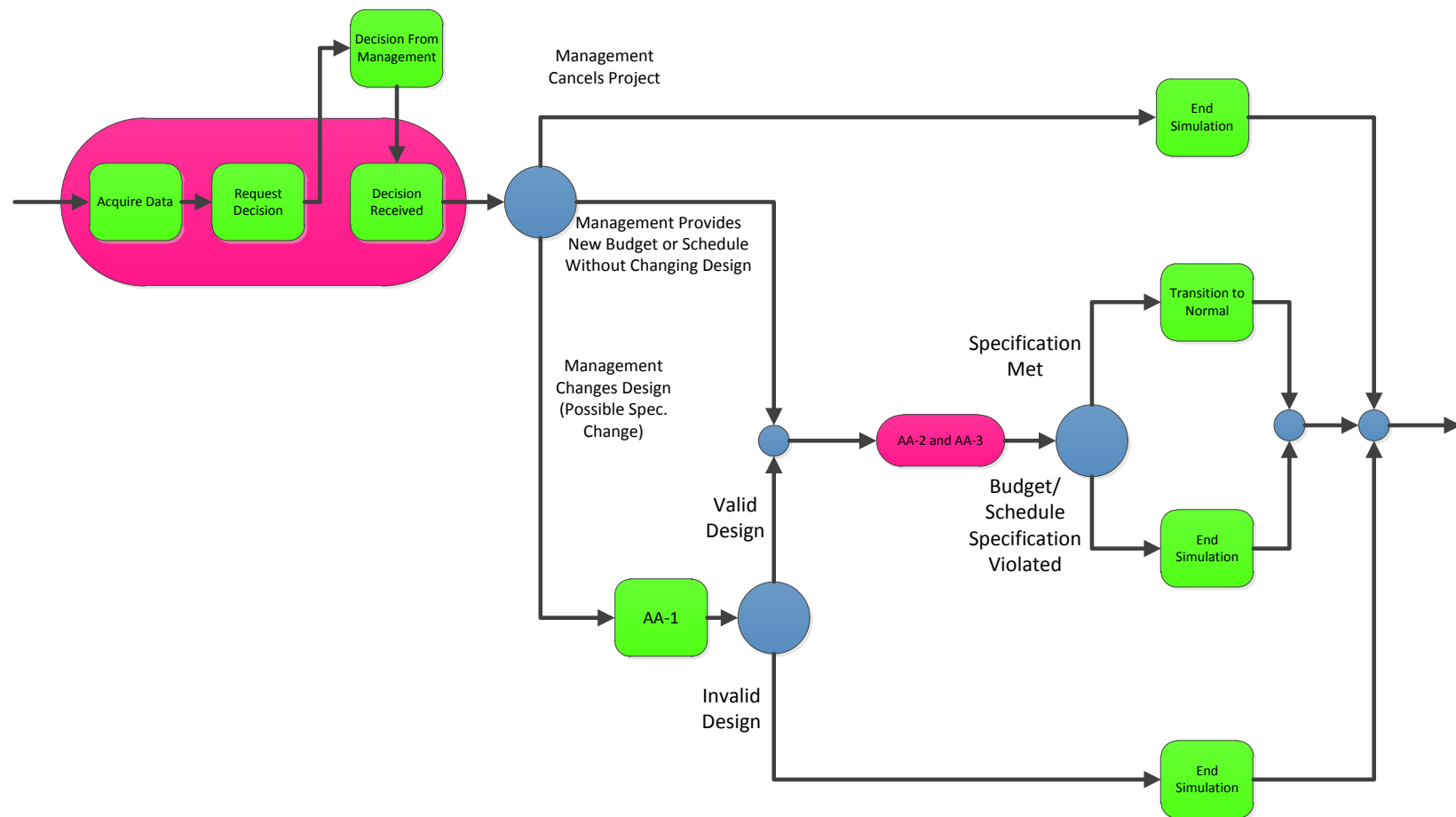


Figure 22 - Sub-Context CxG-5

AA-4, to handle. In such situations, the PM-agent will transition to the **Bad-Design** Major Context where a new design will be requested, in turn removing some of the PM-agent's reliance on AA-4. For sake of completeness, we conclude this chapter with Figure 23, a graphical overview of the CxBR-based PM-agent.

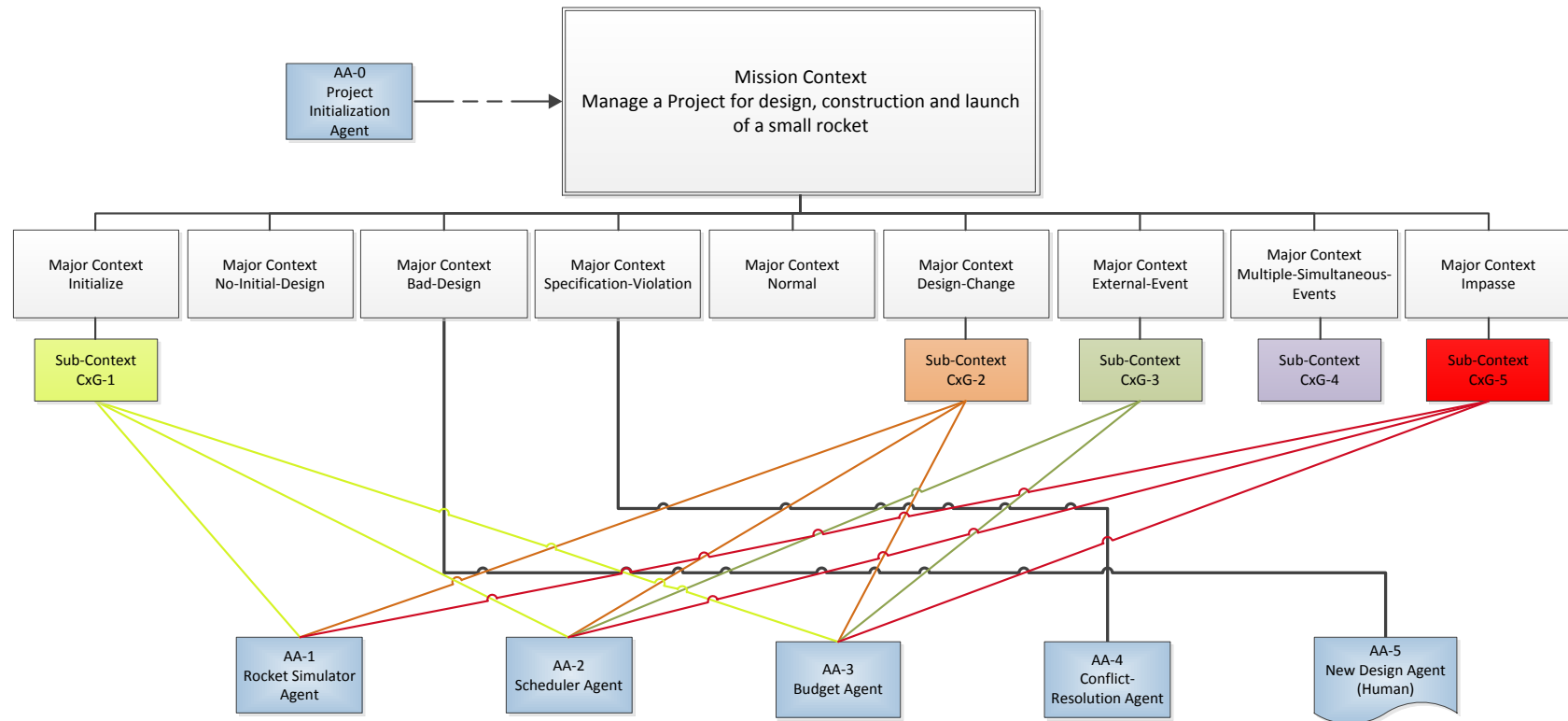


Figure 23 - The PM-agent Architecture

CHAPTER 6 – TESTING

We ran all of the same tests executed and listed in [Gonzalez et al. 2008a] with some minor modifications to account for the changes made to the Auxiliary Agents and the contextualization of the system. Additionally, changes have been made because of the inconsistencies found between the original tests and the defined specification. The original test cases have all been directly copied from Gonzalez et al.'s [2007] paper. Changes made to tests are listed in text with a shaded background. Explanations of the changes made are listed in a “notes section” listed under each test.

As a note, in the previous system design it was stated that the schedule and cost would include effects cause by the parts chosen for the fin and nose cone material. However, it seems that this was not done. Because our system was designed based on the original specification, we include the effects of both. Additionally, since our scheduling agent, AA-3, has been slightly modified from the previous design, calculated–Total FDD will sometimes be slightly different than if they were calculated using the methods of the original AA-3's design. Refer to Chapter 5 for a detailed description of how AA-3 works. Since this change applies to all tests, we have listed it here as to not repeat it for every test's notes section.

By covering all of these test cases we can first show that our approach can at least obtain comparable results to the original design. Additionally, our testing covered the two situations that the original system was not able to handle successfully. Lastly, new test cases were added to show that the new, more extensively contextualized design can effectively and efficiently handle scenarios with near-simultaneous multiple issues.

6.1 The Original Test Cases

The original test cases were divided into three sets. Test set 1 was designed to test the capabilities of the system to react to external events that could occur during the lifetime of the project. In these cases, it is expected that there would be no design changes. Test set 2 was designed to assess the system's ability to handle design changes that could occur during the projects lifetime. These tests include not just design changes by themselves, but also what would happen should external events occur during it as well. Test set 3 was the last of the tests run by Gonzalez et al. The purpose of test set 3 was to stress their system by subjecting it to tests which it clearly was not designed to handle. Of course, with the new changes made by our design, we show that it can.

6.1.1 Test Set 1

For Test Set 1, the mission has the following initial conditions:

- First, the maximum budget must be no greater than 30 MYen. In the original system, this value was 10 MYen, but because of the changes made to our system as described in Chapter 4, the increase in budget accommodates the need for the system to run the old tests.
- Second, the delivery date is no greater than 10 months. The original value was 6 months, but had to be changed for the same reasons as for the budget above.
- Third, the payload is set to 49g. This value is lowered from the original 490g because not even the most powerful engine combination could lift that payload. We assumed that 490g must have been a mistake for 49g.

- Last, we set the minimum altitude to be reached to 0m from 100m. Since the goal of Test Set 1 was not to hit a specified altitude but to handle external events, we determined that it was unnecessary to have a minimum altitude specification.

From here on until Section 6.1.1.1, the text has been quoted from Gonzalez [2007] using the formatting specified earlier. Again, changes made to tests are listed in **text with a shaded background**. Explanations of the changes made are listed in a “notes section” listed under each test.

Test 1.1: This test sets a basic benchmark by confirming the ability of the prototype to retrieve the design information and determine the impacts on the schedule and cost. The initial design of the rocket in this test meets acceptability in terms of technical feasibility, cost and schedule. Furthermore, there are no external events of system redesigns or cost or schedule changes to cause any deviations from the *Normal* Major Context. The design of the rocket is as follows:

Initial Design:

Number of stages: One
Engine selected: A8-3
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 33 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 10.0 cm
Fin Height: 0.0 cm
Payload Weight: 49 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: The design characteristics are such that they are acceptable to lift the required payload to the required altitude.

Predicted total cost of project:

Engine:	1.0 MYen
Control System:	3.5 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	29.5 MYen

Predicted Schedule:

Engine:	2 mo. construction; 1 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 4 months = 10 months

Storyboard: None

Expected Results: Because no external events occur or design changes take place, the PM-agent should remain in **Normal** Major Context throughout the six-month duration of the project.

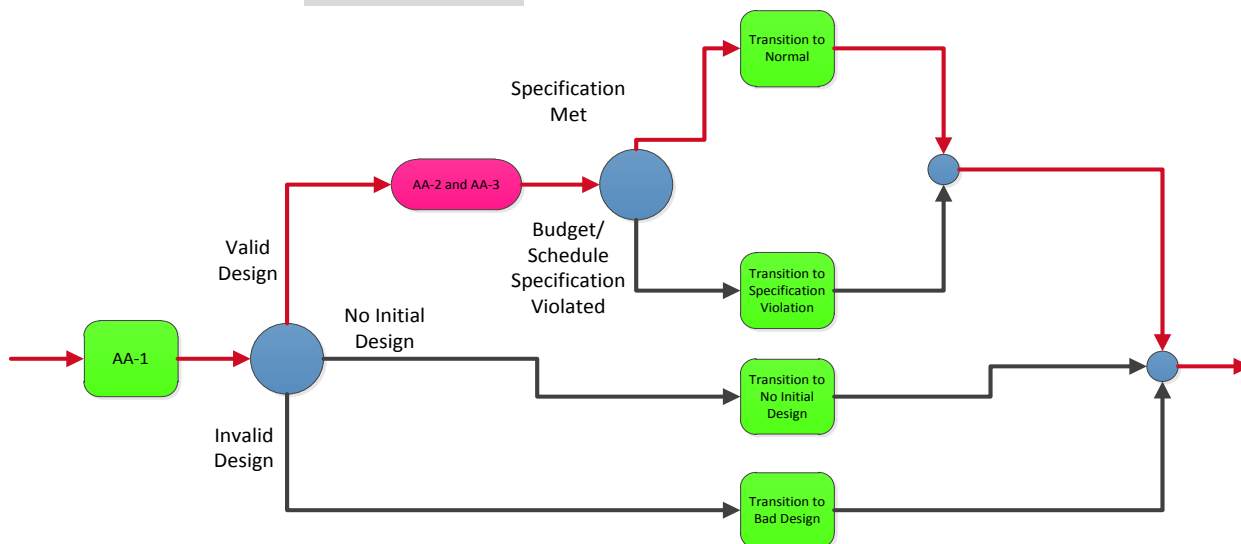
Notes: None

Test Results:

✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated

✧ Day 1 – 300

- Major Context = **Normal**
- Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 1.2: This test uses the same initial conditions of Test 1.1, but introduces an external event that causes the cost to rise, but within the allowed budgetary limitation. Therefore, the description below for initial design, cost and schedule are identical to Test 1.1. Any differences with Test 1.1 are highlighted in italics for ease of detection.

Initial Design:

Number of stages: One
Engine selected: A8-3
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 33 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 10.0 cm
Fin Height: 0.0 cm
Payload Weight: 49 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: As in Test 1.1, the design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	1.0 MYen
Control System:	3.5 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	29.5 MYen

Predicted Schedule:

Engine:	2 mo. construction; 1 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. Installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 4 months = 10 months

Storyboard: 1) *On day 127, a price increase of 0.5 MYen is announced by the control system supplier.*

Expected Result: *External event #1 - PM-agent transitions to **External-Event Major Context** and after evaluating the situation, determines that external cost increase does not exceed the maximum budget and transitions back to Normal.*

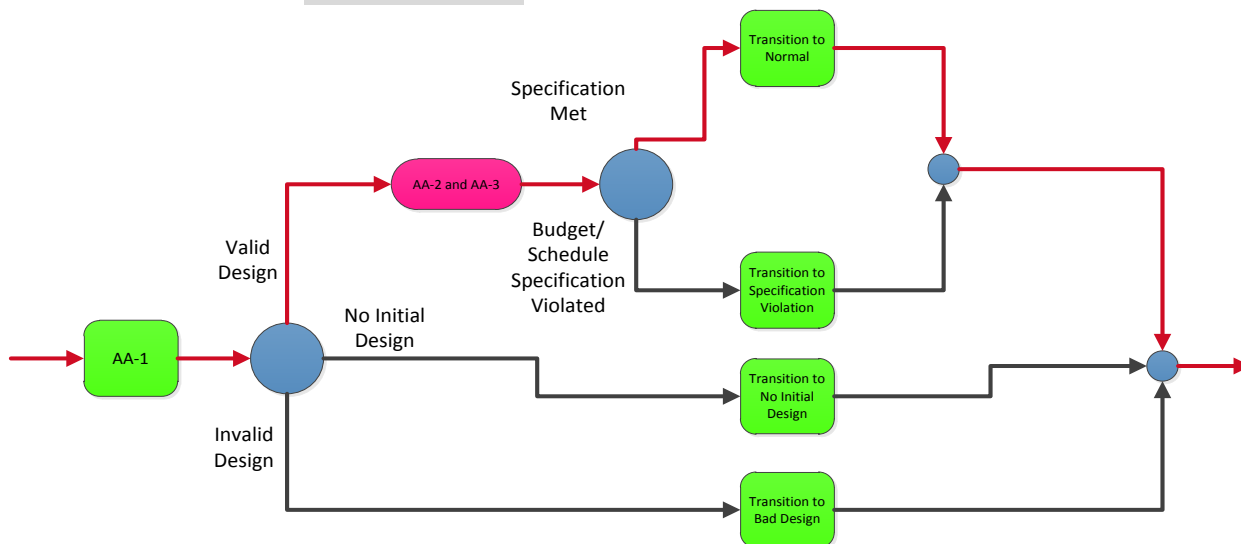
Notes: None

Test Results:

✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize Major Context Activated**
- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated

✧ Day 1 – 126

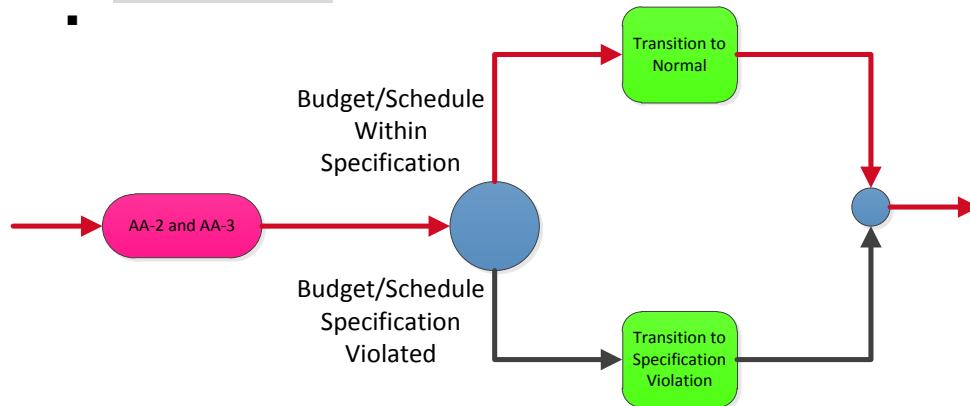
- Major Context = **Normal**
- Normal Action Function Activated

✧ Day 127

○ Event 1:

- External Event Announced – 0.5 MYen Cost Increase
- Transition to External Event → True

- CxG-3 Activated
-



- ✧ Day 128
 - Event 1: Transition To **Normal** Major Context → True
 - Event 2: Major Context **Normal** Activated
- ✧ Day 129 - 300
 - Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 1.3: Same as Test 1.1, except that the external event makes the new cost exceed the maximum budget. All the design information is the same as Tests 1.1 and 1.2, but is repeated here for the sake of completeness. Any differences are marked as italics for ease of identification.

Initial Design:

Number of stages: One
Engine selected: A8-3
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 33 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 10.0 cm
Fin Height: 0.0 cm
Payload Weight: 49 g
Parachute Diameter: 1 ft (default)
Control System: ACS-100

Feasibility of Design: As in Test 1.1, the design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	1.0 MYen
Control System:	3.5 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	29.5 MYen

Predicted Schedule:

Engine:	2 mo. construction; 1 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. Installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 4 months = 10 months

Storyboard: 1) On day 127, a price increase of 1.5 MYen is announced by the control system supplier.

Expected Result: *External event #1 – PM-agent transitions to **External-Event** Major Context and after evaluating the situation, it determines the specification has been violated. The PM-agent will then transition to **Specification-Violation** Major Context where AA-4, the corrective action Auxiliary Agent, will be activated to apply the cost reduction operator and reduce cost by 10% by laying off overhead staff. A 10% reduction of the new cost of 31 MYen will bring the total cost back within the limit of 30 MYen.*

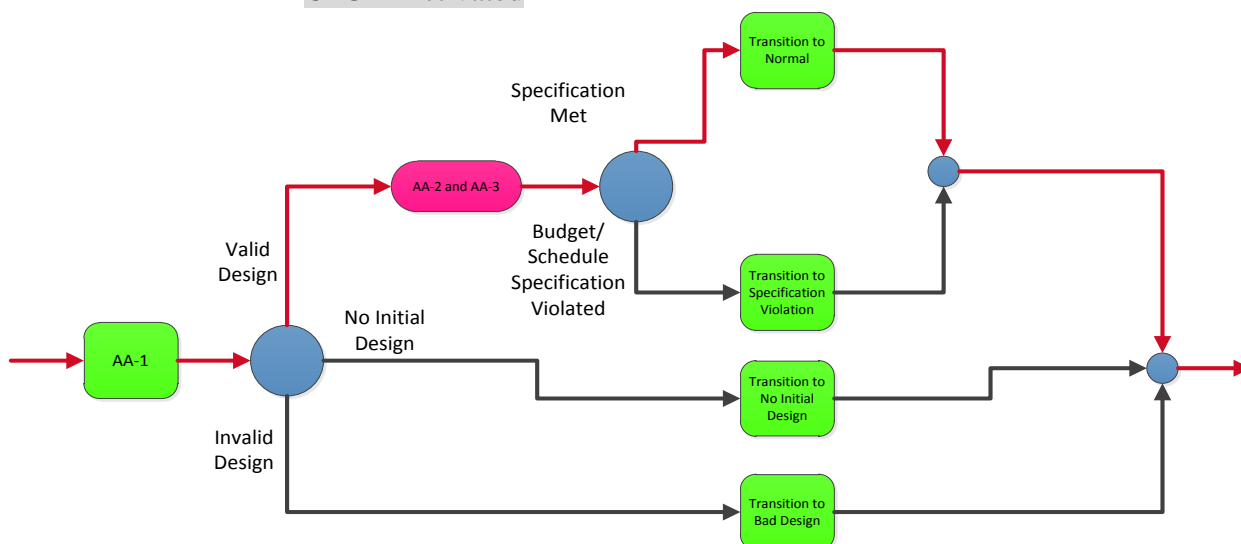
Notes: In the previous design, the PM-agent would have just called upon AA-4 directly. With this new more contextualized system, the system will first go to **Specification-Violation** Major Context before calling upon AA-4.

Test Results:

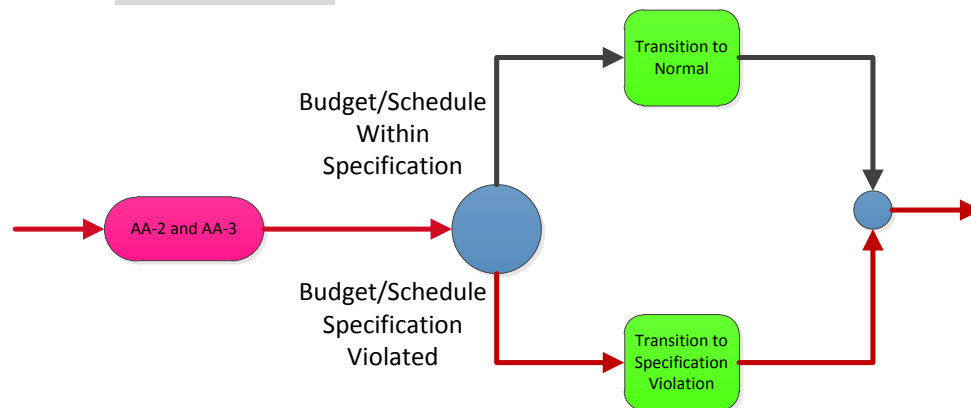
✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated
- ✧ Day 1 – 126
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 127
 - Event 1:
 - External Event Announced – 1.5 MYen Cost Increase
 - Transition to **External Event** Major Context → True
 - CxG-3 Activated



- ✧ Day 128
 - Event 1:
 - Transition to **Specification-Violation** Major Context → True
 - AA-4 Activated
 - Overhead Reduction – First Time
 - 10% Cost Reduction of 3.1 MYen
 - Project Returned to Spec.
 - Transition to Normal Declared
 - Event 2: Transition To **Normal** Major Context → True
 - Event 3: Major Context **Normal** Activated
- ✧ Day 129 - 300
 - Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 1.4: This test is designed to evaluate the prototype's reaction to schedule changes. The initial design, schedule and costs are the same as Tests 1.1, 1.2 and 1.3. Once again, for the sake of completeness, they are repeated below. The differences are written in italic font for ease of detection.

Initial Design:

Number of stages: One
Engine selected: A8-3
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 33 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 10.0 cm
Fin Height: 0.0 cm
Payload Weight: 49 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: As in Test 1.1, the design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	1.0 MYen
Control System:	3.5 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	29.5 MYen

Predicted Schedule:

Engine:	2 mo. construction; 1 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 4 months = 10 months

Storyboard: 1) On day 10, the engine manufacturer informs the PM-agent that the schedule for

the engine has been delayed 70 days beyond the original delivery date.

Expected Result: *External event #1 – PM-agent transitions to **External-Event** Major Context and after evaluating the situation, determines that corrective action is not necessary because the final schedule still meets the maximum ten-month FDD. This is because the time it will take for the delivery of the rocket with the 70 day increase in manufacturing time is still less than the time it will take for the delivery of the control system. Additionally, since we must wait until all parts are received before installation, this external event will have no effect on the project.*

Engine Schedule Change: 2 months + 70 days = 4.33 months

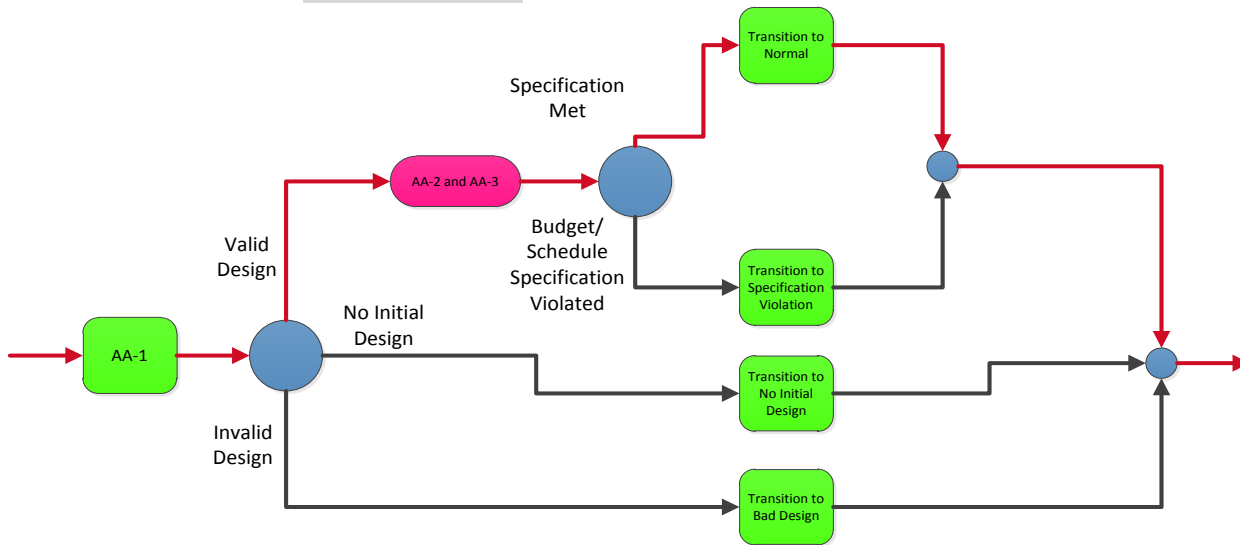
Max Delivery Time: 4.3 months (engine delivery time) < 6 months (control system delivery time)

Notes: The changes to External Event #1's expected results were made because of the changes made to the previous system's design. Nevertheless, these results are equivalent to those that would have been given by the previous system.

Test Results:

- ✧ Day 0:
 - Event 1:
 - Initial Specification and Design Given to PM
 - **Initialize** Major Context Activated

- CxG-1 Activated



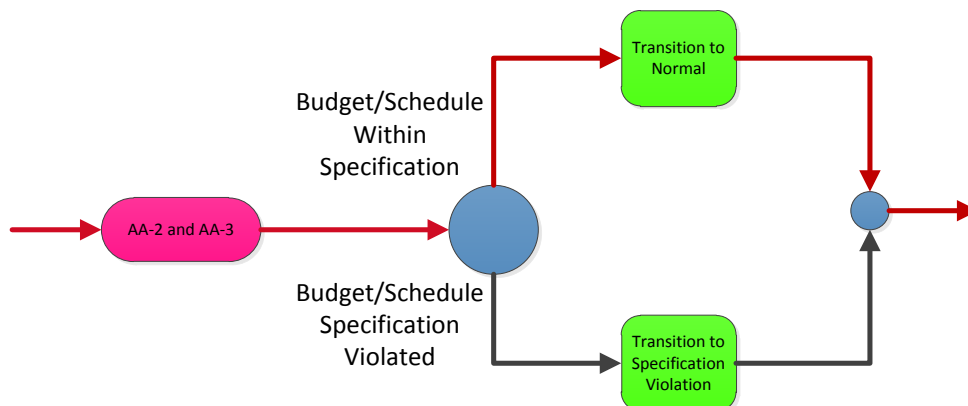
- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated

- ✧ Day 1 – 9

- Major Context = **Normal**
- Normal Action Function Activated

- ✧ Day 10

- Event 1:
 - External Event Announced – 70 Day Delay to Engine Delivery
 - Transition to **External Event** Major Context → True
 - CxG-3 Activated



- ✧ Day 11

- Event 1: Transition To **Normal** Major Context → True
- Event 2: Major Context **Normal** Activated

- ✧ Day 12 - 300

- Major Context = **Normal**
- Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 1.5: This test is designed to evaluate the prototype's reaction to schedule changes that cause it to exceed the FDD. However, the initial design, schedule and costs are the same as Tests 1.1, 1.2, 1.3 and 1.4. Once again, for the sake of completeness, they are repeated below. The differences are written in italic font for ease of detection.

Initial Design:

Number of stages: One
Engine selected: A8-3
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 33 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 10.0 cm
Fin Height: 0.0 cm
Payload Weight: 49 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: As in Test 1.1, the design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	1.0 MYen
Control System:	3.5 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	29.5 MYen

Predicted Schedule:

Engine:	2 mo. construction; 1 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 4 months = 10 months

Storyboard: 1) On day 40, the engine manufacturer informs the PM-agent that the schedule for

the engine has been delayed 130 days beyond the original delivery date.

Expected Result: *External event #1 – PM-agent transitions to **External-Event** Major Context upon detection of external event. Upon evaluating the situation, it determines the specification has been violated. The PM-agent will then transition to **Specification-Violation** Major Context where AA-4, the corrective action Auxiliary Agent, will be activated to apply the schedule reduction operator which uses overtime payments to shorten delivery.*

Revised Engine schedule: 2 months construction + 130 days delay = 6.33 months delivery time

Schedule reduction operator: 50 KYen x 10 days = 0.5 MYen; 29.5MYen + 0.5MYen = 30MYen; 30MYen ≤ 30MYen (Max Budget) → OK

Schedule reduced by 10 days = 190 days – 10days = 180days; 6 months ≤ 6 months (Max FDD) → OK

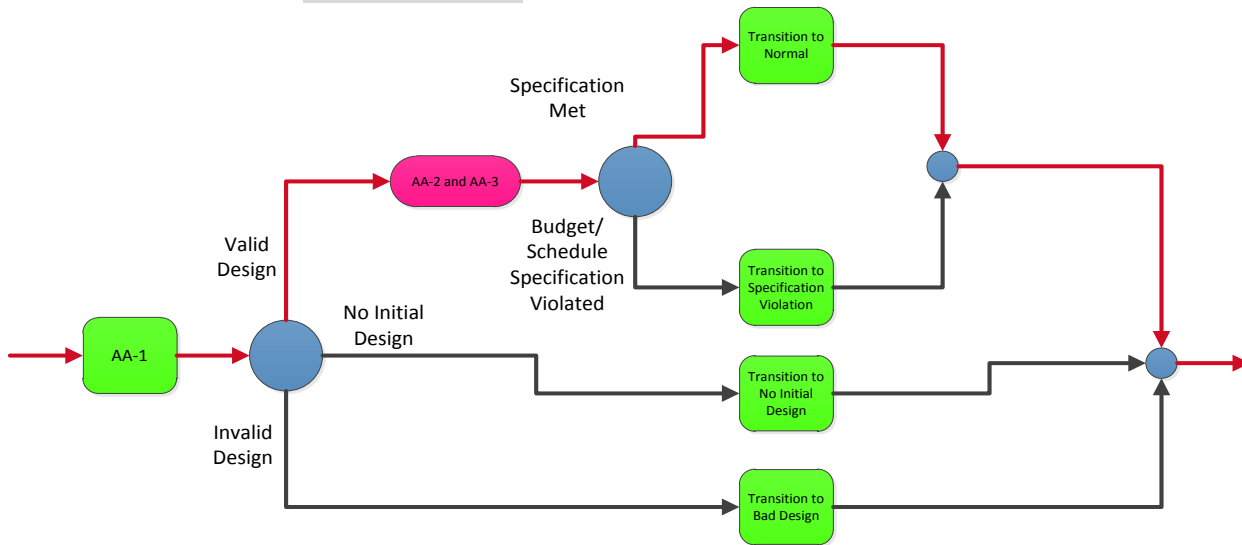
*The PM-agent should now realize it is back in spec. and should transition to the **Normal** Major Context.*

Notes: In the original test, the delay was for 100 days. However, since AA-3 in our system determines the FDD differently from the previous system, we had to increase the delay to 130 days to get the desired conditions this test was looking for. With the delay of 130 days for the engine, the engine now becomes the component with the longest delivery time allowing for the FDD to increase past mission specification by 10 days, as was the goal of the test.

Test Results:

- ✧ Day 0:
 - Event 1:
 - Initial Specification and Design Given to PM
 - **Initialize** Major Context Activated

- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True

- Event 3: Major Context **Normal** Activated

✧ Day 1 – 39

- Major Context = **Normal**

- Normal Action Function Activated

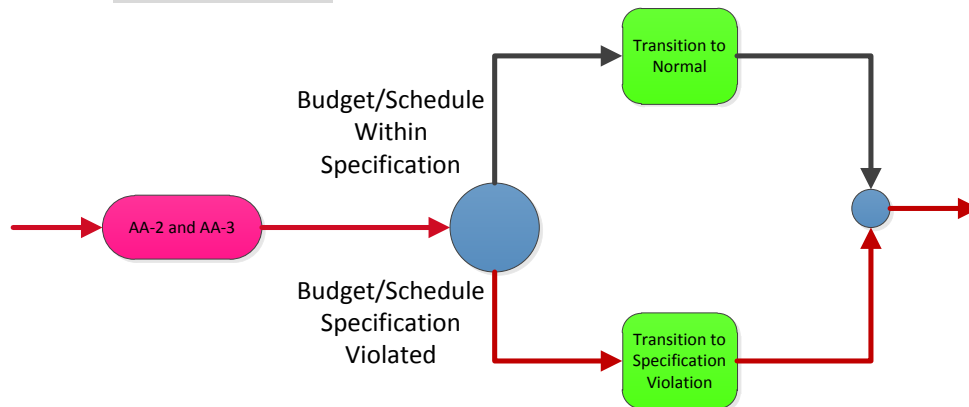
✧ Day 40

- Event 1:

- External Event Announced – 130 Day Delay to Engine Delivery

- Transition to **External-Event** Major Context → True

- CxG-3 Activated



✧ Day 41

- Event 1:

- Transition to **Specification-Violation** Major Context → True

- AA-4 Activated
 - Overtime Paid – 10 days
 - Cost Increase of 0.5 MYen
 - Project Returned to Spec.
 - Transition to Normal Declared
 - Event 2: Transition To **Normal** Major Context → True
 - Event 3: Major Context **Normal** Activated
- ✧ Day 42 - 300
- Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 1.6: This test is designed to evaluate the prototype's reaction to schedule changes and cost increases in the same test. However, the situation is not over-constrained. The initial design, schedule and costs are the same as Tests 1.1, 1.2, 1.3, 1.4 and 1.5. Once again, for the sake of completeness, they are repeated below. The differences are written in italic font for ease of detection.

Initial Design:

Number of stages: One
Engine selected: A8-3
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 33 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 10.0 cm
Fin Height: 0.0 cm
Payload Weight: 49 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: As in Test 1.1, the design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	1.0 MYen
Control System:	3.5 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	29.5 MYen

Predicted Schedule:

Engine:	2 mo. construction; 1 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 4 months = 10 months

Storyboard: 1) On day 40, the engine manufacturer informs the PM-agent that the schedule for the engine has been delayed 130 days beyond the original delivery date. 2) On day 100, the control system manufacturer announces a price increase of 1 MYen.

Expected Result: After **external event #1** is detected, PM-agent is expected to act just as it did in Test 1.5. The PM-agent will transition to **External-Event** Major Context upon detection of external event. Upon evaluating the situation, it determines the specification has been violated. The PM-agent will then transition to **Specification-Violation** Major Context where AA-4, the corrective action Auxiliary Agent, will be activated to apply the schedule reduction operator which uses overtime payments to shorten delivery.

Revised Engine schedule: 2 months construction + 130 days delay = 6.33 months delivery time

Schedule reduction operator: 50 KYen x 10 days = 0.5 MYen; 29.5MYen + 0.5MYen = 30MYen;
 $30MYen \leq 30MYen$ (Max Budget) → OK

Schedule reduced by 10 days = 190 days – 10days = 180days; 6 months ≤ 6 months (Max FDD)
→ OK

The PM-agent should now realize it is back in spec. and should transition to the **Normal** Major Context.

External event #2 causes PM-agent to transition once again to **External-Event** Major Context. This subsequent cost increase now causes the budget limit to be exceeded by 1 MYen. The PM-agent will then transition to **Specification-Violation** Major Context where AA-4, the corrective action Auxiliary Agent, will be activated to apply the operator that reduces cost by 10% by reducing overhead staff: 31.0 MYen x 0.10 = 3.1 MYen. 31.0 MYen – 3.1 MYen = 27.9 MYen, which is within the maximum budget limit.

PM-agent should now realize that all is within spec once again and transition back to **Normal** Major Context. New schedule = 300 days; New cost = 27.9 MYen.

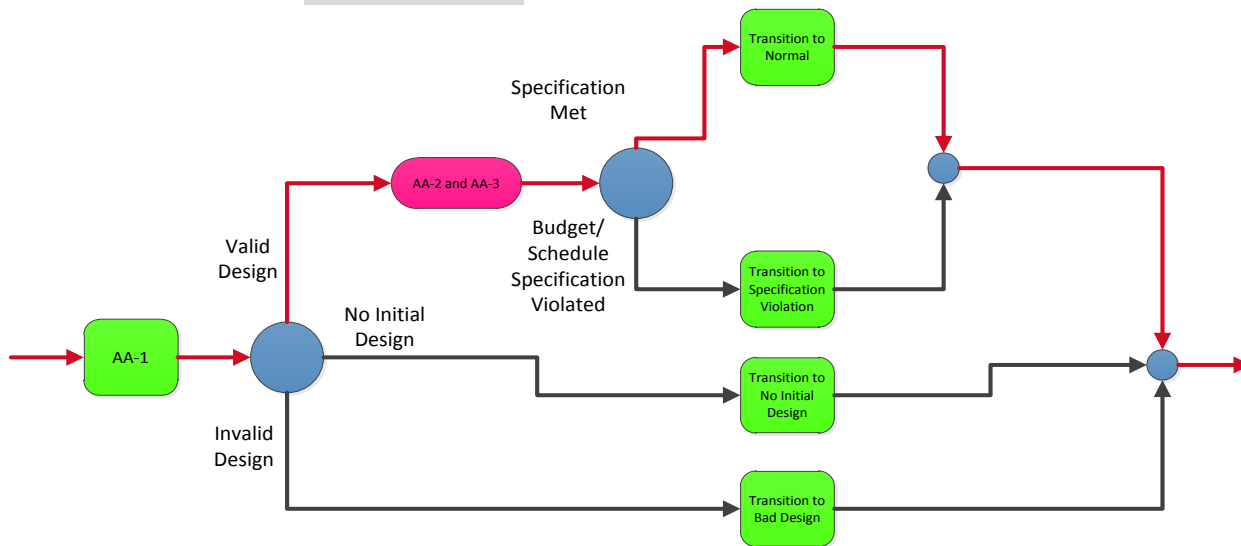
Notes: Changes made for External Event #1 are the same as in Test 1.5. In External Event #2, the reason for the changes made to the expect results are for the same reasons they were made for External Event #1 described in Test 1.5's notes section.

Test Results:

✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True

- Event 3: Major Context **Normal** Activated

✧ Day 1 – 39

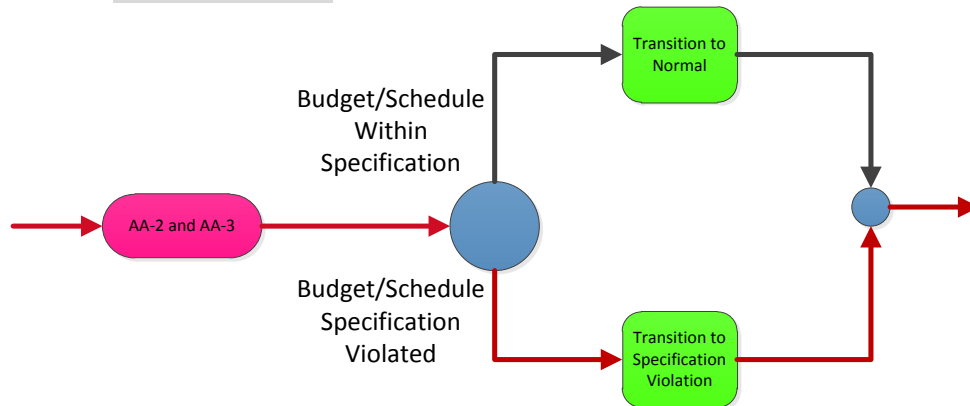
- Major Context = **Normal**
- Normal Action Function Activated

✧ Day 40

○ Event 1:

- External Event Announced – 130 Day Delay to Engine Delivery
- Transition to **External-Event** Major Context → True

- CxG-3 Activated



✧ Day 41

○ Event 1:

- Transition to **Specification-Violation** Major Context → True
- AA-4 Activated
 - Overtime Paid – 10 days
 - Cost Increase of 0.5 MYen
 - Project Returned to Spec.
 - Transition to Normal Declared

○ Event 2: Transition To **Normal** Major Context → True

○ Event 3: Major Context **Normal** Activated

✧ Day 42 - 99

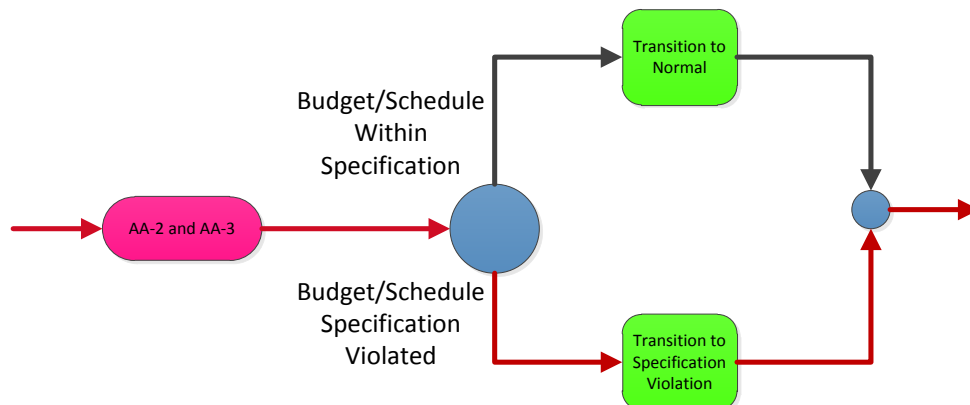
○ Major Context = **Normal**

○ Normal Action Function Activated

✧ Day 100

○ Event 1:

- External Event Announced – 1 MYen Cost Increase
- Transition to **External-Event** Major Context → True
- CxG-3 Activated



✧ Day 101

○ Event 1:

- Transition to **Specification-Violation** Major Context → True
 - Major Context = **Specification-Violation**
 - AA-4 Activated
 - Overhead Reduced – First Time
 - 10% Cost Reduction = 3.1 MYen
 - Project Returned to Spec.
 - Transition to Normal Declared
 - Event 2: Transition To **Normal** Major Context → True
 - Event 3: Major Context **Normal** Activated
- ✧ Day 102 – 300
- Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 1.7: This test is designed to evaluate the prototype's reaction to schedule changes and cost increases in the same test. However, the situation is not over-constrained, although it requires multiple applications of corrective operators. The initial design, schedule and costs are the same as Tests 1.1, 1.2, 1.3, 1.4, 1.5 and 1.6. Once again, for the sake of completeness, they are repeated below. The differences are written in italic font for ease of detection.

Notes: It has been deemed that this test should have been impossible for the original system as it did not have capabilities to handle multiple simultaneous events. The test that was actually run was not that which was described above. Instead, their test showed that AA-4 could activate two correction operators at the same time. Nevertheless, this should not have been allowed to happen according to the original specification given for AA-4. Their results showed that overtime was paid in order to account for a schedule specification violation which caused then cause a budget specification violation. Nevertheless, our new system does account for the test described above. However, since this is a new test that was not actually capable of running on the original system, it has been relabeled as Test 4.1

Test 1.8: This test is designed to evaluate the prototype's reaction to schedule changes and cost increases in the same test. However, the situation is now over-constrained, leading to an impasse. The initial design, schedule and costs are the same as Tests 1.1, 1.2, 1.3, 1.4, 1.5, 1.6 and ~~1.7~~. They are repeated below for the sake of completeness, the differences are written in italic font for ease of detection.

Notes: Again for the same reasons as listed in the notes section of Test 1.7, this test has been relabeled. It is now Test 4.2.

6.1.1.1 Test Set 1 Conclusions

As can be seen from the tests above, Test Set 1 was a group of simple tests which assessed the system's ability to handle external events. Just as the original system was able to accomplish, our more contextualized system was able to provide the expected results for the test cases from Test Set 1. We will now move on to test set two to verify the system's ability to handle these tests of the original system as well.

6.1.2 Test Set 2

For Test Set 2, the mission will each have the following initial conditions: First, the maximum budget must be no greater than 38 MYen. In the original system, the value was 18 MYen. Since this is 8 MYen more than Test Set 1's original maximum budget, we found increasing our modified budget by 8 MYen would allow us to maintain a comparable system. The same also applies to the schedule. The previous system increased the FDD by 6 months for Test Set 2. Therefore, for this system, we decided to use a FDD of 16 months, a 6 month increase on Test Set 1's 10 months. For the minimum altitude to be reached, we chose a value of 350m. The original test had a value of 400m, however, it seemed that a few of the test cases could not actually obtain this altitude despite the claims that the test cases did. Last, the payload of the rocket will be 20g. This value is identical to original test case and did not need to be modified. From here on until Section 6.1.2.1, quoted from Gonzalez [2007] using the formatting specified at the beginning of this chapter. Again, changes made to tests are listed in **text with a shaded background**. Explanations of the changes made are listed in a "notes section" listed under each test.

Test 2.1: This test suite uses Mission #2, defined above. This test is designed to verify that the design change evaluation mechanism works as designed. A design change in the form of an engine change comes in on day 100, but this design change introduces no conflict. The differences are written in italic font for ease of detection.

Initial Design:

Number of stages: One
Engine selected: C6-3
Fin material: 1/8" Balsa
Nose Cone material: Balsa
Body Length: 10 cm
Nose Cone Length: 4.0 cm
Fin Width: 2.0 cm
Body Diameter: 2.5 cm
Fin Length: 4.0 cm
Fin Height: 0.0 cm
Payload Weight: 20 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: The design characteristics are such that they are acceptable to lift the required payload to the required altitude.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	3.5 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	32.5 MYen

Predicted schedule:

Engine:	6 month construction; 3 month installation
Control System:	6 months
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. Installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 6 months = 12 months

Storyboard: 1) Major design change occurs on day 100. Engine design is now B6-6, with a

delivery of 4 + 2 months and a cost of 5 MYen

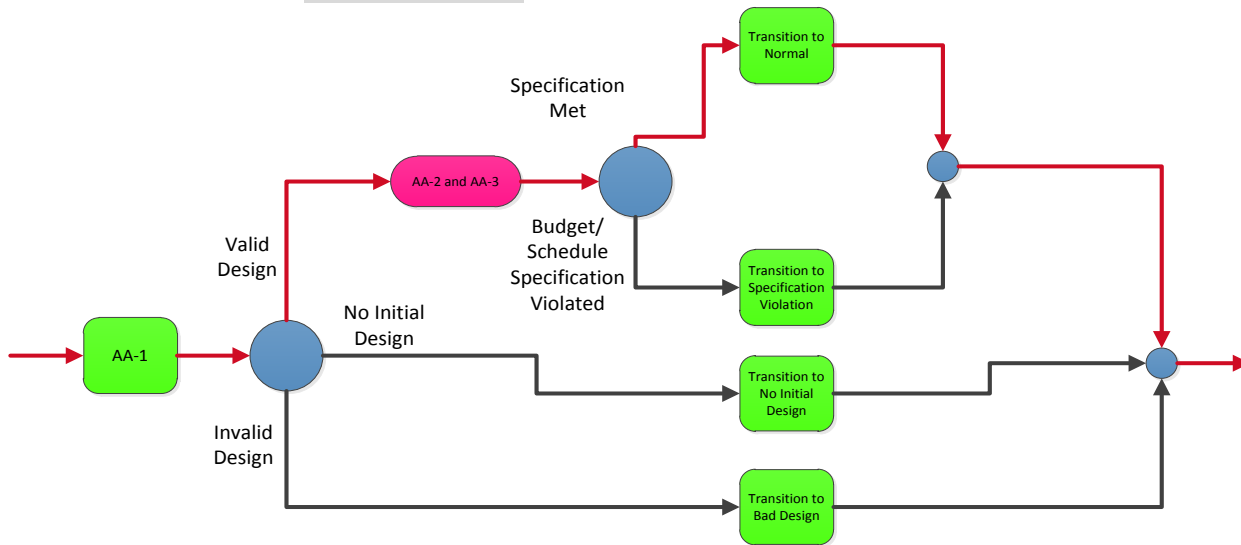
Expected Results: The situation does not represent a conflict because the design change is technically acceptable, the cost is lower and the delivery times are shorter. Therefore, the design change should be accepted by the *PM-agent*. *PM-agent* should start in the **Initialize** Major Context and then transition to the **Normal** Major Context. Then on day 100 the PM-agent will transition to the **External-Event** Major Context upon detection of the external event, and transition back to Normal upon realization that there is no conflict.

Notes: For this test, the original design change set the engine to B6-2. However, we had to modify the design change to B6-6 because B6-2 could not hit the altitude that was specified for the test. Also, in the expected results section, we modify the statement that the system starts in **Normal** Major Context to that the system starts in the **Initialize** Major Context before it transitions to **Normal** Major Context. This is because our new system now always starts in the **Initialize** Major Context.

Test Results:

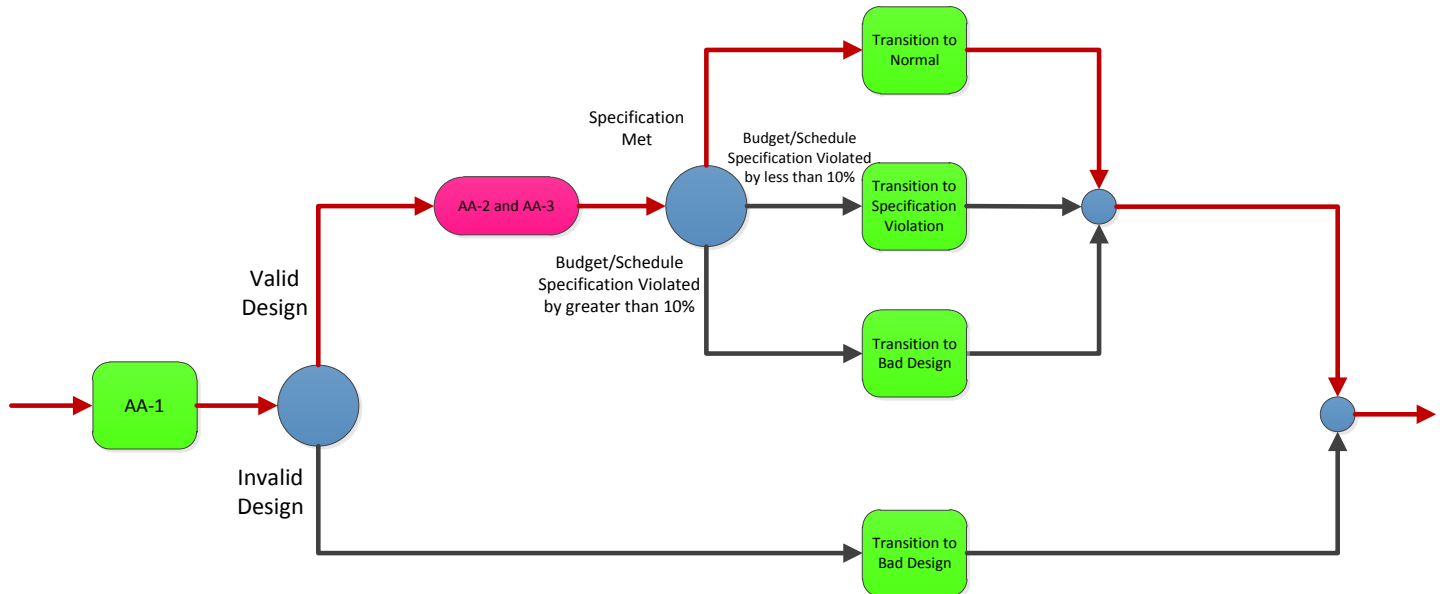
- ✧ Day 0:
 - Event 1:
 - Initial Specification and Design Given to PM
 - Initialize Major Context Activated

- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated
- ✧ Day 1 – 99
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 100
 - Event 1:
 - Design Change Given To PM
 - Event 2:
 - Design Change Announced
 - Transition to **Design-Change** Major Context → True

- CxG-2 Activated



- ✧ Day 101

- Event 1:
 - Transition to **Normal** Major Context → True
- Event 2:
 - Major Context = **Normal**
 - Normal Action Function Activated

- ✧ Day 102 - 370

- Major Context = **Normal**
- Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 2.2: This test is designed to verify that the design change evaluation mechanism works as designed. The initial rocket design parameters are the same as Test 2.1. It is repeated here for the sake of completeness. An engine design change occurs on Day 5, but the new design is inadequate for the performance specification requirements. It should be rejected outright by PM-agent. The differences from Test 2.1 are written in italic font for ease of detection.

Initial Design:

Number of stages: One
Engine selected: C6-3
Fin material: 1/8" Balsa
Nose Cone material: Balsa
Body Length: 10 cm
Nose Cone Length: 4.0 cm
Fin Width: 2.0 cm
Body Diameter: 2.5 cm
Fin Length: 4.0 cm
Fin Height: 0.0 cm
Payload Weight: 20 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: The design characteristics are such that they are acceptable to lift the required payload to the required altitude.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	3.5 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	32.5 MYen

Predicted schedule:

Engine:	6 month construction; 3 month installation
Control System:	6 months
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 6 months = 12 months

Storyboard: 1) *Major design change occurs on day 5. Engine design is now A8-5, with a delivery of 2 + 1 months and a cost of 1.5 MYen.*

Expected Results: *The situation does not represent a conflict from a cost and schedule standpoint, but the new engine selection does not permit the rocket to achieve the specified performance specifications. This will cause the PM-agent to transition to **Design-Change Major***

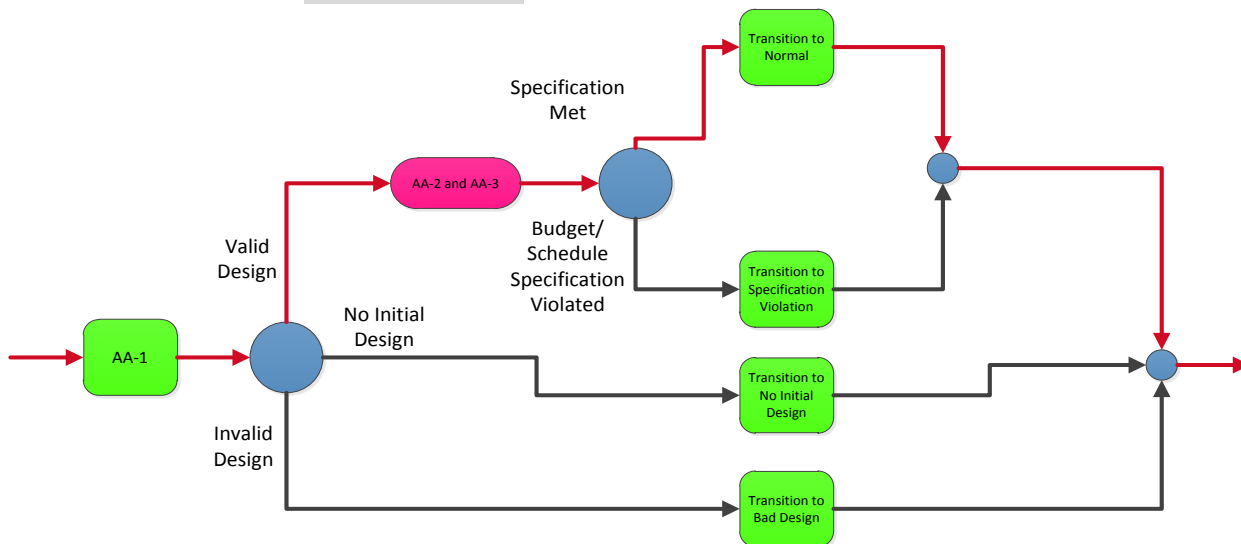
Context. *The design change should be rejected outright by PM-agent as unacceptable. This will then transition the system to the **Bad-Design Major Context** where the PM-agent will have the DA replace it. We will have the DA choose B6-6 which will meet specification. This should then have the PM-agent transition to the **Design-Change Major Context**.*

Notes: Again, because of our new more contextualized design, the expected results of the system have changed slightly. Before, the results of the original system claimed the system would have just stayed in the **Design-Change Major Context** while awaiting a new design from the DA. Now, we have a Major Context design specifically to handle a rejected design. The new system transitions to the **Bad-Design Major Context** to await results from the DA.

Test Results:

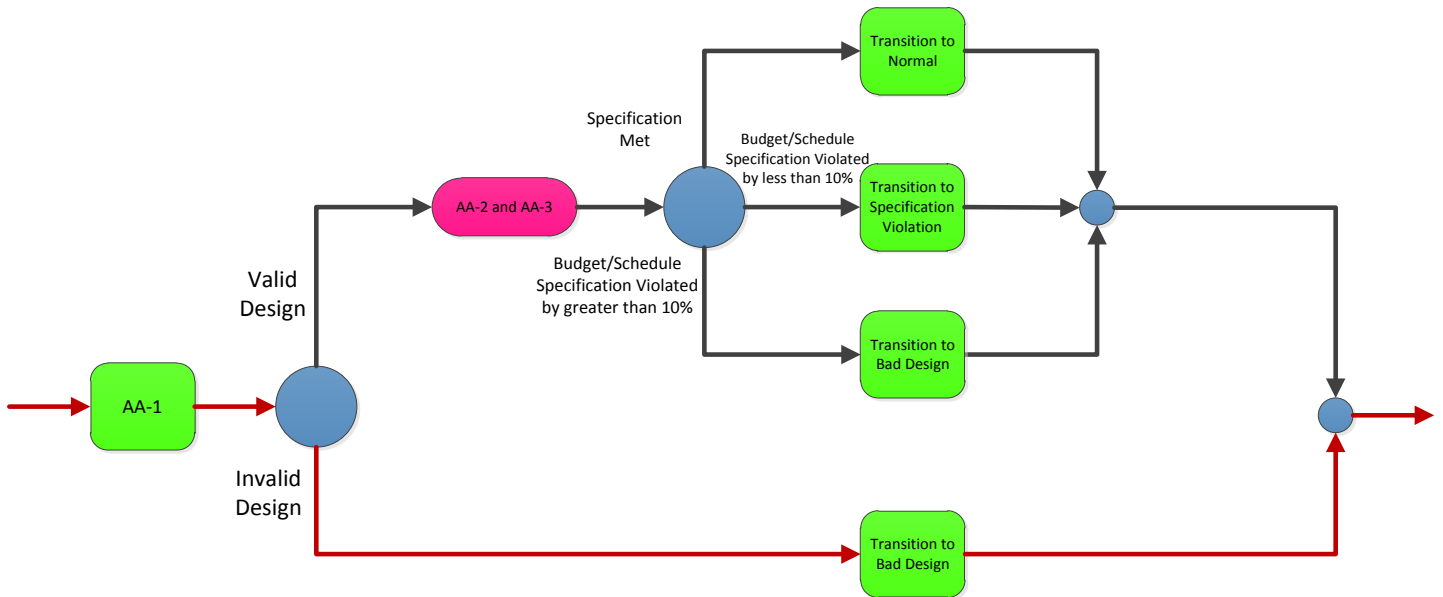
- ✧ Day 0:
 - Event 1:
 - Initial Specification and Design Given to PM

- **Initialize** Major Context Activated
- **CxG-1** Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated
- ✧ Day 1 – 4
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 5
 - Event 1:
 - Design Change Given To PM
 - Event 2:
 - Design Change Announced
 - Transition to **Design-Change** Major Context → True

- CxG-2 Activated



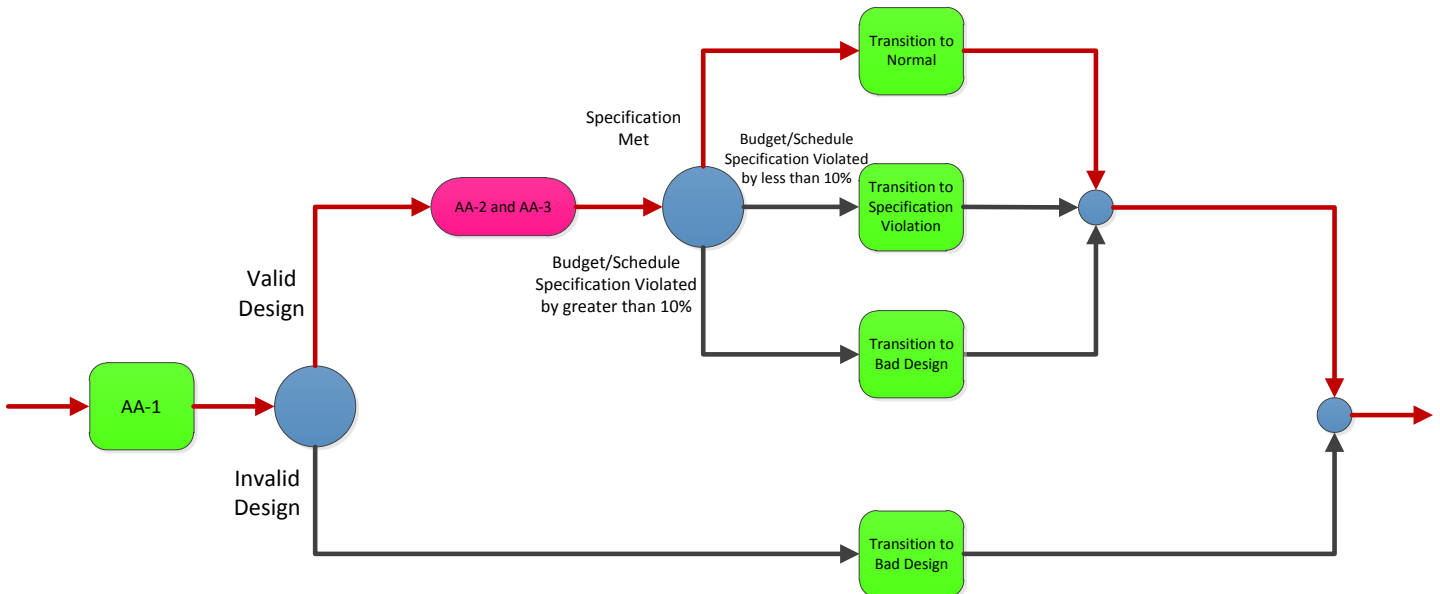
✧ Day 6

○ Event 1:

- Transition to **Bad-Design** Major Context → True
- Rule Request New Design → True
- DA Provides New Design
- Transition to **Design-Change** Major Context Announced

○ Event 2:

- Transition to **Design-Change** Major Context → True
- CxG-2 Activated



- Event 3:
 - Transition to **Normal** Major Context → True
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 7 - 330
 - Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 2.3: This test is designed to verify that the design change evaluation and recovery mechanism works as designed. The initial rocket design parameters are the same as Test 2.1 and 2.2. It is repeated here for the sake of completeness. The differences from Test 2.1 are written in italic font for ease of detection. This test introduces a design change that while technically acceptable, causes problems with schedule and cost. However, the situation is not over-constrained and a resolution should be found.

Initial Design:

Number of stages: One
Engine selected: C6-3
Fin material: 1/8" Balsa
Nose Cone material: Balsa
Body Length: 10 cm
Node Cone Length: 4.0 cm
Fin Width: 2.0 cm
Body Diameter: 2.5 cm
Fin Length: 4.0 cm
Fin Height: 0.0 cm
Payload Weight: 20 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: The design characteristics are such that they are acceptable to lift the required payload to the required altitude.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	3.5 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	32.5 MYen

Predicted schedule:

Engine:	6 month construction; 3 month installation
Control System:	6 months
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. installation
Premium for multiple stages:	0.0 months

Total FDD

6 months + 6 months = 12 months

Storyboard: 1) Major design change occurs on day 60. Engine design is now C6-7, a very powerful but expensive engine, with a delivery of 6 + 3 months and a cost of 8 MYen.

Expected Results: The new engine design is clearly capable of providing the required performance but the cost is prohibitive. Because the design causes a conflict which is less than 10% of the total budget, the PM-agent cannot outright reject the design, so it must seek corrective actions to reduce the cost.

Cost of new rocket = 32.5 MYen + 8 MYen = 40.5 MYen

40.5 MYen > 38 MYen Max Budget

40.5 MYen x 0.10 = 4.05 MYen reduction by applying corrective action operator #1.

New cost = 36.45 MYen.

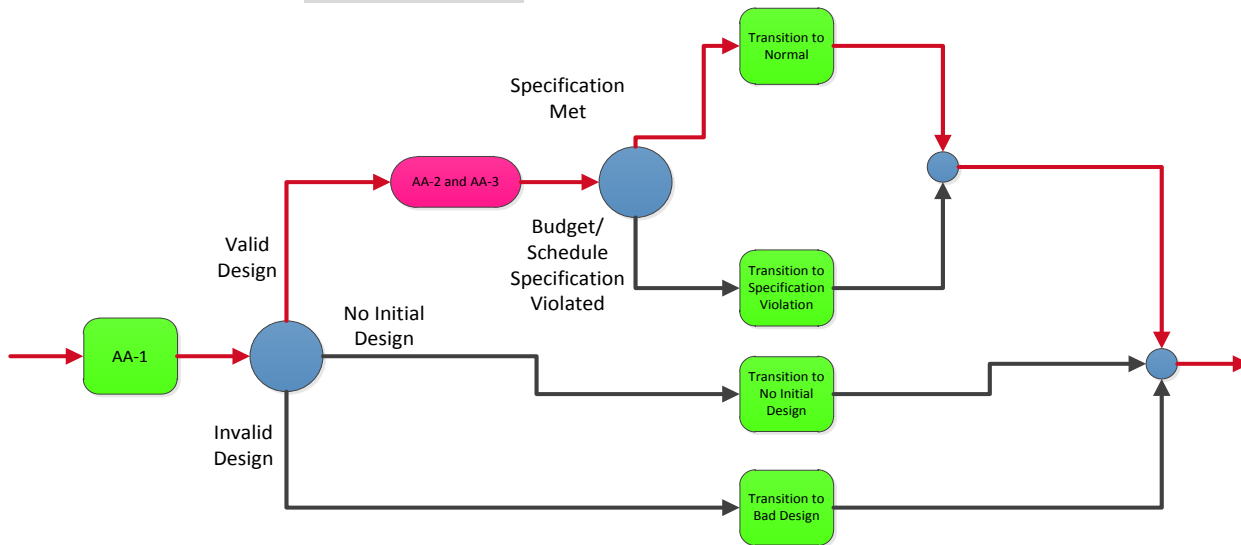
PM-agent should then accept the design change with the cost reduction measures. It should then transition to **Normal** Major Context.

Notes: Because of modifications to how the auxiliary agents work in our system, the use of engine D12-7 in a design change would cause an impasse. Since the goal of this test was to the recovery mechanisms of the system, we have changed the engine chosen by the DA to be C6-7. This will still cause the increase in cost that was expected in the old test allowing us to show that the system is clearly able to handle this situation in an equivalent way the original system was able to.

Test Results:

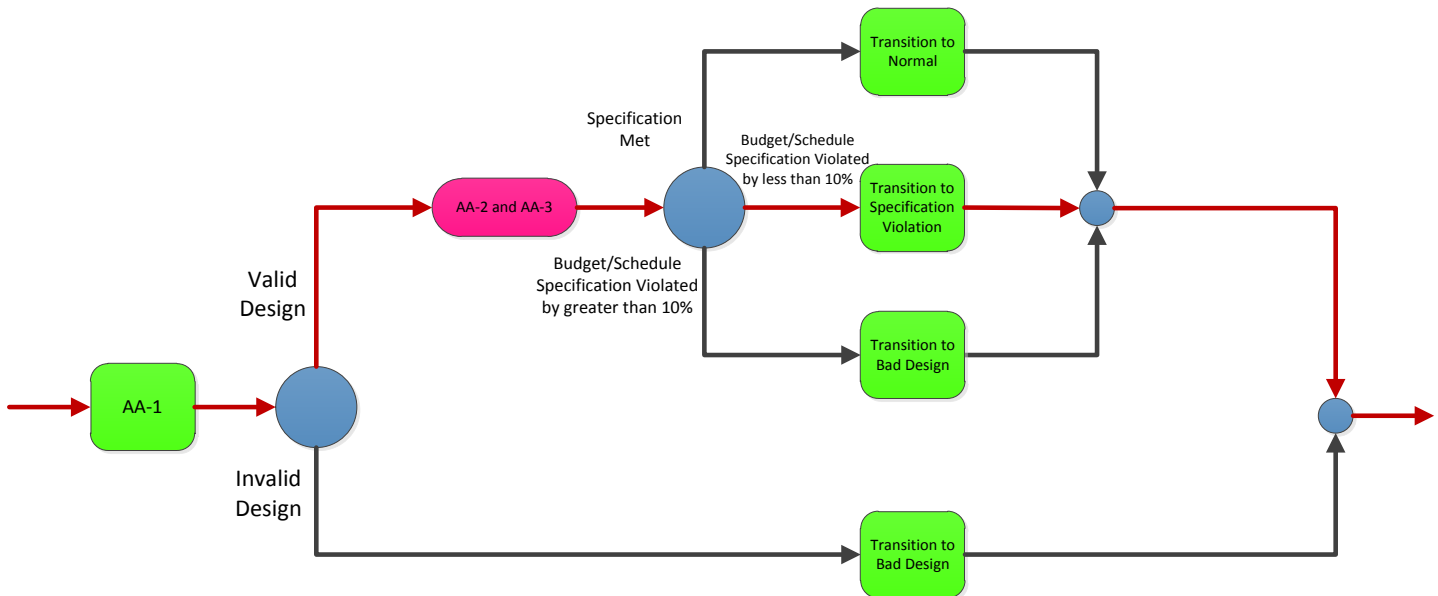
- ✧ Day 0:
 - Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated
- ✧ Day 1 – 59
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 60
 - Event 1:
 - Design Change Given To PM
 - Event 2:
 - Design Change Announced
 - Transition to **Design-Change** Major Context → True

- CxG-2 Activated



✧ Day 61

- Event 1:

- Transition to **Specification-Violation** Major Context → True
- AA-4 Activated
- Overhead Reduction – First Time
- 10% Cost Reduction = 4.05 MYen
- Project Returned to Spec.
- Transition to Normal Declared

- Event 2:

- Transition to **Normal** Major Context → True
- Major Context = **Normal**
- Normal Action Function Activated

✧ Day 62 - 420

- Major Context = **Normal**
- Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 2.4: This test is designed to verify that the design change evaluation and recovery mechanism works as designed. The initial rocket design parameters are the same as Tests 2.1, 2.2 and 2.3. It is repeated here for the sake of completeness. The differences from Test 2.1 are written in italic font for ease of detection. This test introduces a design change that while technically acceptable, causes schedule and cost problems **so great that the DA will have to be called to choose a new design for the system.** ~~The situation in this test is over-constrained, leading to an impasse.~~

Initial Design:

Number of stages: One
Engine selected: C6-3
Fin material: 1/8" Balsa
Nose Cone material: Balsa
Body Length: 10 cm
Nose Cone Length: 4.0 cm
Fin Width: 2.0 cm
Body Diameter: 2.5 cm
Fin Length: 4.0 cm
Fin Height: 0.0 cm
Payload Weight: 20 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: The design characteristics are such that they are acceptable to lift the required payload to the required altitude.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	3.5 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	32.5 MYen

Predicted schedule:

Engine:	6 month construction; 3 month installation
Control System:	6 months
Fin Material:	1.5 mo. construction; 1.5 mo. installation

Nosecone Material: 1.5 mo. construction; 1.5 mo. Installation
Premium for multiple stages: 0.0 months
Total FDD 6 months + 6 months = 12 months

Storyboard: 1) *Major design change occurs on day 60. Engine design is now D12-5, a very powerful but expensive engine, with a delivery of 8 + 3 months and a cost of 18 MYen.*

Expected Results: *First the PM-agent will transition to **Design-Change** Major Context where it will determine the feasibility of the new design. The PM-agent will determine that the new engine will carry the rocket to the required altitude, however, it will cause the rocket to go out of spec.*

Cost of new rocket = 32.5 MYen + 18 MYen = 52.5 MYen

52.5 MYen > 36.0 MYen Max Budget

Schedule of new rocket = [60 days (current day) + 240 days (8 months)] (time for all parts to be delivered) + 180 days (6 months installation) = 480 days

480 days > 460 days FDD

*Since the cost is greater than the budget by more than 10%, the PM-agent will transition to **Bad-Design** Major Context. Here the PM-agent will request a new design is requested from the DA. On the next day, we will have the DA provide a new design of B6-4.*

*This will cause the PM-agent to transition back to **Design-Change** Major Context where it will evaluate the new design.*

Cost of new rocket = 32.5 MYen + 4 MYen (B6-4) = 36.5 MYen

36.5 MYen > 36.0 MYen FDD

Schedule of new Rocket = [61 days (current day) + 120 days (4 months)] (time for all parts to be

delivered) + 150 days (5 months installation) = 331 days

331 days < 460 days FDD

*The new engine will allow the rocket to reach the required altitude. However, the cost will still not be within the specified budget. Nonetheless, since the new cost is within 10% of the max budget, the PM-agent will transition to **Specification-Violation** Major Context instead of **Bad-Design** Major Context as it did before. Here, AA-4 will be activated to evaluate what corrective actions it can take. AA-4 will use the cost reduction operator for the first time, lowering the cost by 10%.*

*36.5 MYen * 10% = 3.65 MYen*

36.5 MYen – 3.65 MYen = 32.85 MYen

*This places the rocket back within spec. allowing the PM-agent to transition to **Normal** Major Context where it will stay the reset of the mission.*

Notes: For this test, the description had to be changed slightly. In the new system, if a design change causes the system to be out of spec. by more than 10%, it will transition to the **Bad-Design** Major Context. Because of this, the system will not become over-constrained and be required to transition to **Impasse** Major Context. For this reason, we removed this condition from the test description. Instead we state that the system will require assistance from DA in order to fix the situation being experienced by the PM-agent.

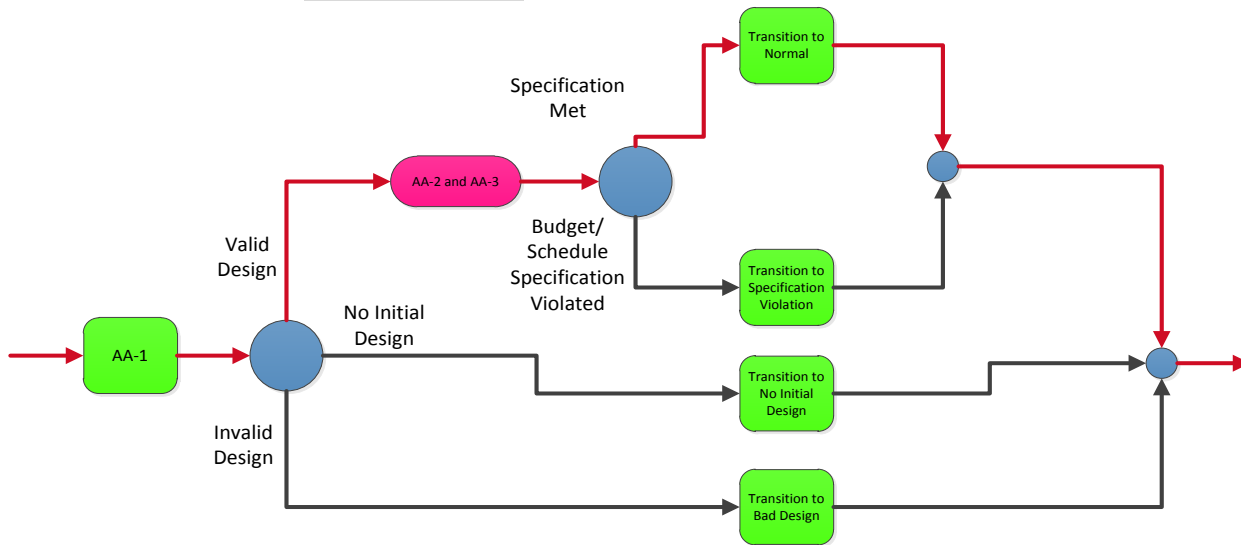
Test Results:

✧ Day 0:

○ Event 1:

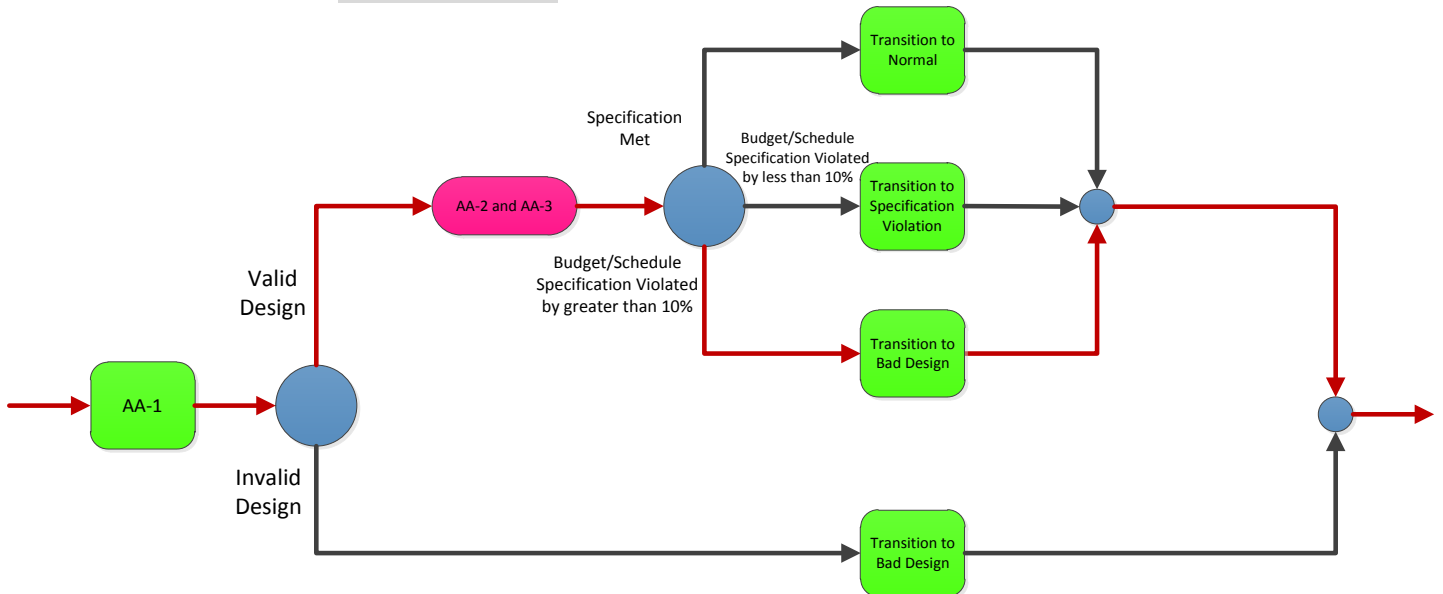
- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated

- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated
- ✧ Day 1 – 59
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 60
 - Event 1:
 - Design Change Given To PM
 - Event 2:
 - Design Change Announced
 - Transition to **Design-Change** Major Context → True

- CxG-2 Activated



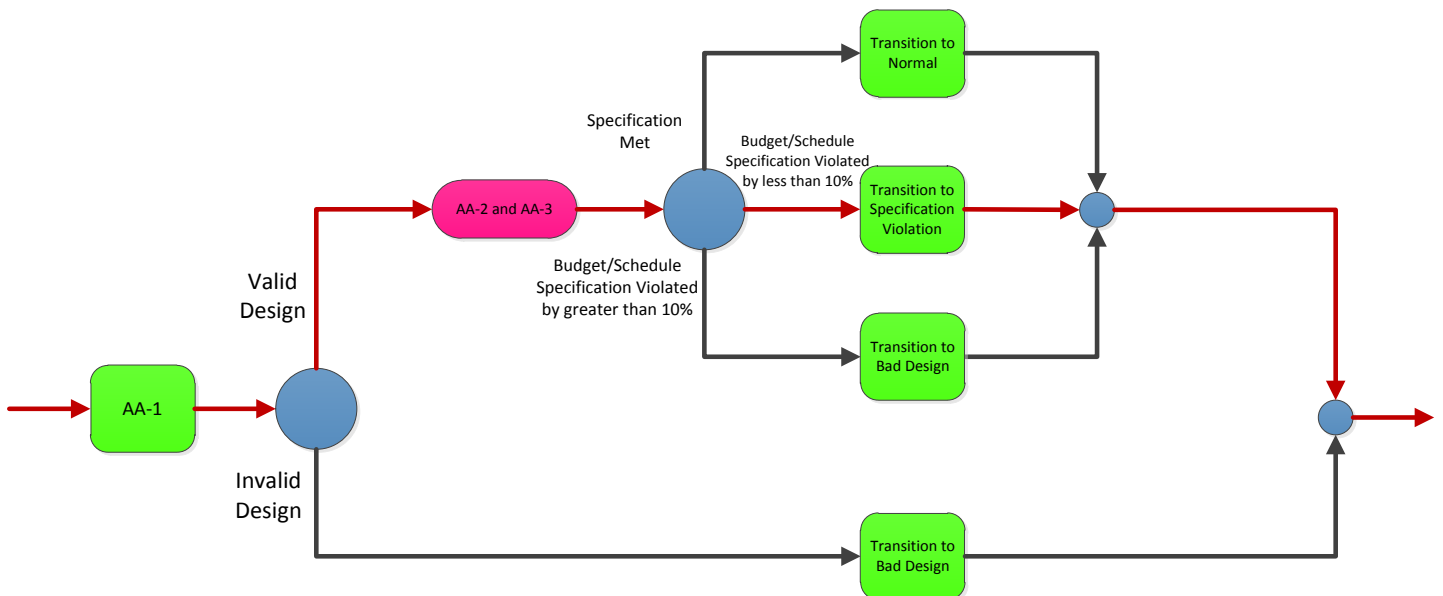
✧ Day 61

- Event 1:

- Transition to **Bad-Design** Major Context → True
- Major Context = **Bad-Design**
- Action Rule Request New Design – Activated
- DA Gives New Design

- Event 2:

- Transition to **Design-Change** Major Context → True
- Major Context = **Design-Change**
- CxG-2 Activated



- Event 3:

- Transition to **Specification-Violation** Major Context → True
- Major Context = **Specification-Violation**
- AA-4 Activated
- Overhead Reduction – First Time
- 10% Cost Reduction = 3.65 MYen
- Project Returned to Spec.
- Transition to Normal Declared
- Event 4:
 - Transition to **Normal** Major Context → True
 - Major Context = **Normal**
 - Normal Action Function Activated
- Event 5:
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 62 - 331
 - Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 2.5: This test is designed to verify that the design change evaluation and recovery mechanism works as designed. The initial rocket design parameters are the same as Tests 2.1, 2.2, 2.3 and 2.4. It is repeated here for the sake of completeness. The differences from Test 2.1 are written in italic font for ease of detection. This test introduces a design change to the body dimensions and payload that makes the rocket unable to lift the payload to the required altitude.

Initial Design:

Number of stages: One
Engine selected: C6-3
Fin material: 1/8" Balsa
Nose Cone material: Balsa
Body Length: 10 cm
Nose Cone Length: 4.0 cm
Fin Width: 2.0 cm
Body Diameter: 2.5 cm
Fin Length: 4.0 cm
Fin Height: 0.0 cm
Payload Weight: 20 g
Parachute Diameter: 1ft (default)
Control System: ACS-300

Feasibility of Design: The design characteristics are such that they are acceptable to lift the required payload to the required altitude.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	3.5 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	32.5 MYen

Predicted schedule:

Engine:	6 month construction; 3 month installation
Control System:	6 months
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. Installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 6 months = 12 months

Storyboard: 1) *Major design change occurs on day 60.*

Body Length: 30 cm

Nose Cone Length: 15.0 cm

Fin Width: 10.0 cm

Body Diameter: 2.5 cm

Fin Length: 10.0 cm

Payload Weight: 150 g

No change in cost or delivery time will occur.

Expected Results: *On day 60, the PM-agent will transition to **Design-Change** Major Context upon receiving notification of the design change. Here, the PM-agent will recognize that the rocket is not able to obtain the altitude designated by specification. Therefore, the PM-agent will transition to **Bad-Design** Major Context where it will request a new design from the DA. Since for this mission it has been determined that we do not want to change cost or schedule and reduction of the payload is acceptable, the DA will lower the payload to 50g which is still greater than the original 20g. This will allow the rocket to reach the specified altitude. The PM-agent will then transition back to **Design-Change** Major Context where it will be determined that the rocket is back within spec. This will thereby cause the PM-agent to transition back to **Normal** Major Context where it will stay the rest of the mission.*

Notes: In the original Test 2.5, the payload was increased to 50g. However, with a payload of 50g along with the other changes made to the design, the rocket simulator still allowed the rocket to reach the altitude required by spec. For this reason, we have increased the payload to 150g which will cause the rocket to fail to reach the specified altitude as was the idea of the test. Then, when the design is reject and the PM-agent transitions to **Bad-Design** Major Context, we will have the DA reduce the payload down to 50g which will allow the proper altitude to be reached again. In addition, although the test stated there would be no change in cost or delivery time that would occur, the test stated that there was a cost increase of 3MYen. For the reason that the test

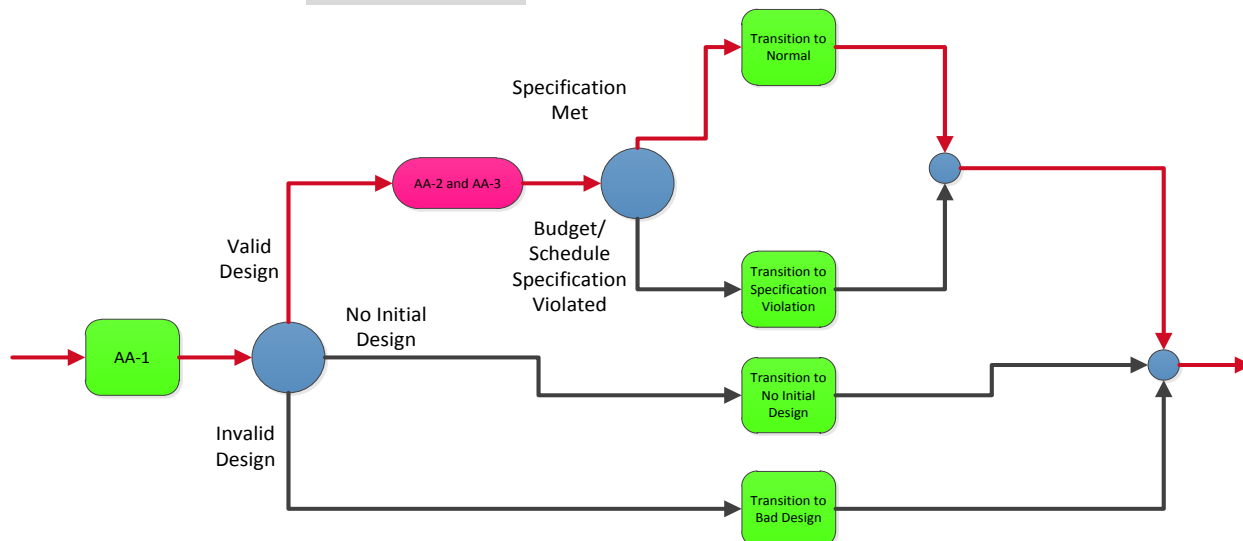
design stated that there would not be a change in cost or schedule, we will ignore the addition of an extra 3MYen. Thereby, although we are not going to be running the test as was previously done, we will now be running the test as described in the test's specification.

Test Results:

✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated

✧ Day 1 – 59

- Major Context = **Normal**
- Normal Action Function Activated

✧ Day 60

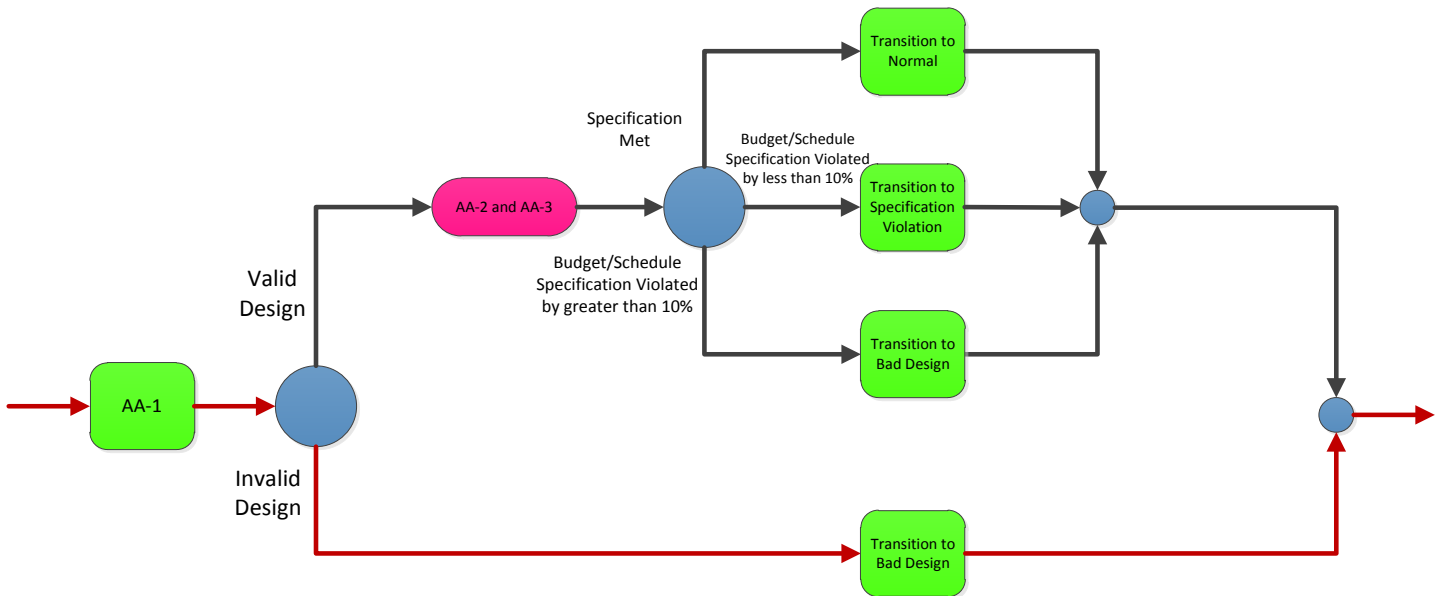
○ Event 1:

- Design Change Given To PM

○ Event 2:

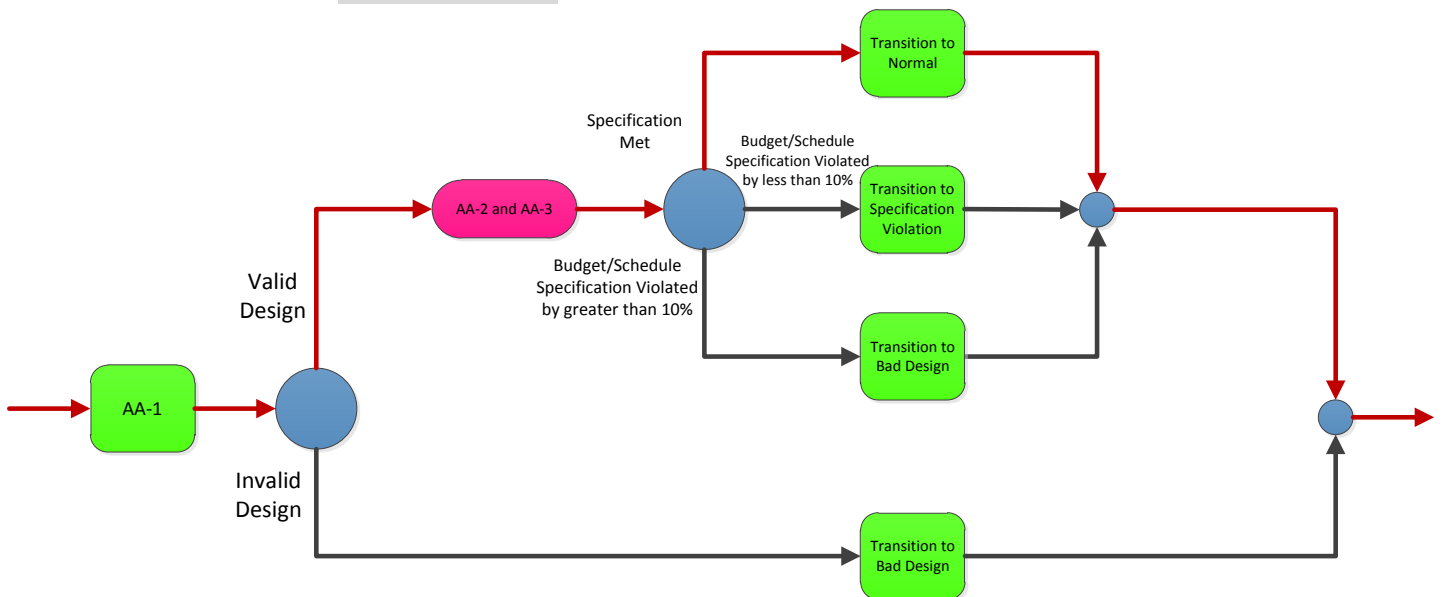
- Design Change Announced
- Transition to **Design-Change** Major Context → True
-

- CxG-2 Activated



Day 61

- Event 1:
 - Transition to **Bad-Design** Major Context → True
 - Rule Request New Design → True
 - DA Provides New Design
 - Transition to **Design-Change** Major Context Announced
- Event 2:
 - Transition to **Design-Change** Major Context → True
 - CxG-2 Activated



- Event 3:

- Transition to **Normal** Major Context → True
- Major Context = **Normal**
- Normal Action Function Activated
- Event 4:
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 62 - 360
 - Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

6.1.2.1 Test Set 2 Conclusions

Test Set 2 was designed by Gonzalez et al. to test their system's ability to handle design changes as well as external events occurring over the lifetime of a project. Just like Gonzalez et al.'s system, ours is able to handle these test cases. However, unlike Gonzalez et al.'s system, our system provides more effective solutions for some of the tests. For example, in Test 2.4, when the **Design-Change** Major Context is activated, CxG-2 verifies whether violations to specification that occurred exceed 10% before trying to transition to the **Specification-Violation** Major Context, where in turn, AA-4 would have been called to try to solve the problems with the specification. Because of the new contextualization added by CxG-2, the PM-agent instead transitioned to **Bad-Design** Major Context, a new major context introduced in our system, to handle to specification problem, thereby taking some of the responsibility off of AA-4 and providing a more effective solution. We will now continue on to Test Set 3 where we will demonstrate the abilities of our system to handle the situations that the original system was incapable of handling.

6.1.3 Test Set 3

For the previous system, Test Set 3 was designed to contain test cases that the system was thought to be incapable of handling. However, with our new, more contextualized approach, we were able to accommodate for these two situations easily. For Test Set 3, we had the following initial conditions: First, the maximum budget must be no greater than 40 MYen. In the original system, since there was an increase of 2 MYen in Test Set 3 from Test Set 2, we increased the budget to reflect this change just as we did in Test Set 2. Again, the same also goes for the schedule. The previous system decreased the FDD by 3 months from the values of Test

Set 2. However, we determined that 13 months is too short for a multi-stage rocket. Therefore, we decided to leave this value the same as that used for Test Set 2: 16 months. For the minimum altitude to be reached, we continued to use the value of 350m, since the original test had Test Set 3 use the same value as Test Set 2 for this parameter. Lastly, the payload of the rocket will be 35g. Again, just as in Test Set 2, this value is identical to original test case and did not need to be modified. From here on until Section 6.1.3.1, the text has been quoted from [Gonzalez, 2007] using the formatting specified at the beginning of this chapter. Again, changes made to tests are listed in **text with a shaded background**. Explanations of the changes made are added in a “notes section” listed under each test.

Test 3.1: This test introduces another engine design as the initial design offered by the DAs. It includes a heavier payload and three-stage rocket. The initial design is not workable from a cost and schedule standpoint. The initial design is not the same as the prior tests, so it is all shown in italics. This test case uses Mission #3, defined above. The heavier payload of Mission #3 requires multiple stage rockets, which complicate the decision-making but provide several alternatives.

Initial Design:

Number of stages: 1
Engines selected: Stage 1 - C6-0, Stage 2 - C6-0, Stage - C6-3
Fin Material: 1/8" Balsa Wood
Cone Material: Balsa Wood
Body Length: 40 cm
Nose Cone Length: 20.0 cm
Fin Width: 4.0 cm
Body Diameter: 1.9 cm
Fin Length: 24.0 cm
Fin Height: 0.0 cm
Payload Weight: 35 g
Parachute Diameter: 1ft (default)
Control System: ACS-300

Feasibility of Design: The design characteristics are such that they are acceptable to lift the required payload to the required altitude.

Predicted total cost of project:

Engine:	29 MYen
Control System:	4.5 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	5 MYen
Total Cost:	58.5 MYen

Predicted schedule:

Engine:	6 month construction; 9 month installation
Control System:	6 months
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. Installation
Premium for multiple stages:	3.0 months installation

Total FDD

6 months + 15 months = 21 months

Storyboard: *None*

Expected Results: *The rocket design is unacceptable at the start of the simulation. The cost is too high and the FDD is not within the specified schedule. First the system will begin in **Initialize** Major Context . From here, the PM-agent will recognize that although the initial design is able to meet the required altitude, it does not meet the rest of the mission's specification. This will cause the PM-agent to transition to **Specification-Violation** Major Context. Here, the PM-agent will call upon AA-4 to try to resolve the conflict. AA-4 first looks at cost. It will look to see if an overhead reduction could solve the budget conflict. Nonetheless, AA-4 will realize this solution is not a viable one as not enough overhead is allowed to be cut in order to maintain the budget specified. This will cause the PM-agent to then transition to the **Impasse** Major Context. Here, we will have the DA choose a new design. Since the original design never actually went through, all schedule and cost calculations will be are calculated from scratch. We will have the DA move to a 2-stage design using engine B6-0 for Stage 1 and B6-6 for Stage 2. This will decrease the cost and FDD of the rocket while still hitting the specified altitude.*

New Rocket Cost:

Engine:	10 MYen
Control System:	4 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	1 MYen

Total Cost:	35 MYen
--------------------	---------

New Rocket Schedule:

Engine:	4 month construction; 4 month installation
Control System:	6 months

Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. installation
Premium for multiple stages:	1.0 months installation
Total FDD	6 months + 8 months = 14 months

35 MYen < 40 MYen Budget

14 months < 16 Months FDD

The PM-agent is now expected to transition back to Normal where it will stay for the rest of the mission.

Notes: In the original system, some of the values chosen were inconsistent with what was allowed for the NASA rocket simulator to work. Therefore, in order to run the test, some changes had to be made in order for the test to be able to run.

First, the engines selected had to be modified. In the original test, the following engines were used: Stage 1 – A8-3, Stage 2 – 1/2A6-2, Stage 3 – B6-4. However, this combination of engines is impossible. Stage 1 and 2 engines must be booster engines; however, these engines are clearly not. (A booster engine has the following format: LetterNumber-0. Without getting into technical detail about model rocket engines, one need just understand the 0 signifies a booster rocket.) In addition, for no specified reason, in the test results section of the original system, the author began with engine C6-3 and then had a design change to the engines just specified. The test description specified that the engine combination above should have been the original engines chosen for the rocket. Therefore, in order to run a test that accurately reflected what was actually intended by the test description, we modified the original engines to those provided in our initial design section. Second, for fin and nose cone material, the description was listed as “light.” However, since this didn’t actually describe a material, we inferred this to mean the use of balsa wood since this is described as a light material in the original test paper. Lastly, the fin length had

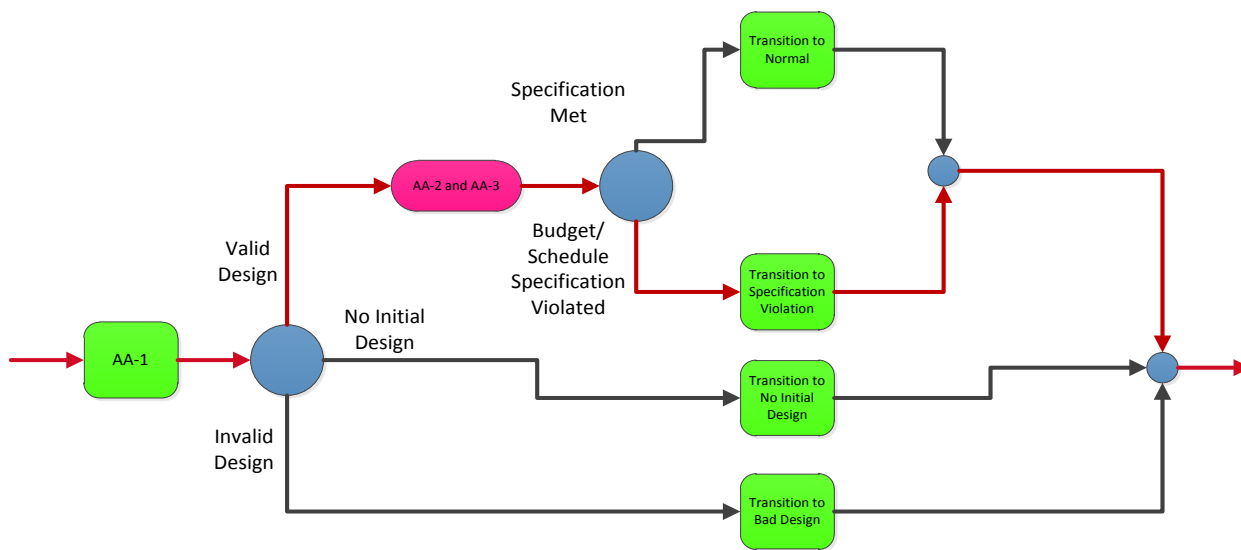
to be increase from the original 15". The simulator requires a minimum of 18". Therefore, in order for the simulator to run, we had to increase this value. We decided to use the simulators default value of 24".

Test Results:

✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



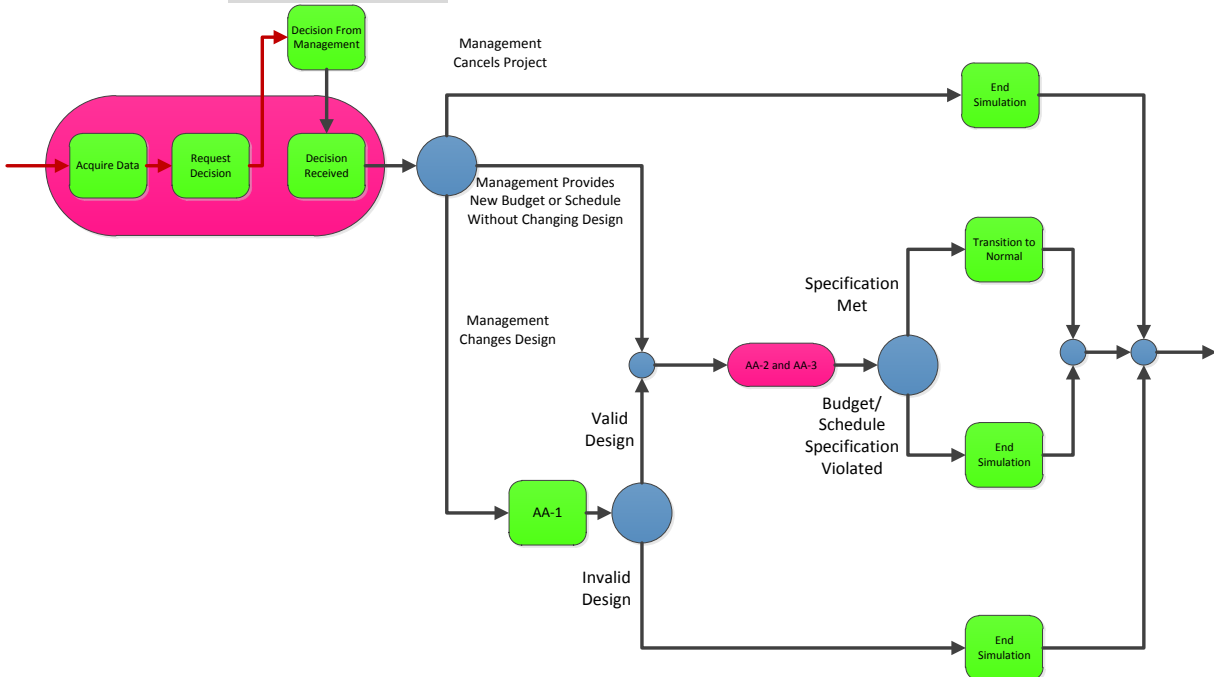
○ Event 2:

- Transition to **Specification-Violation** Major Context → True
- Rule Conflict Resolution → True
- AA-4 Activated
 - Overhead Reduction Needed Is Not Allowed
 - Impasse Declared

○ Event 3:

- Transition to **Impasse** Major Context → True
- Impasse Mission Context Activated

- CxG-5 Activated

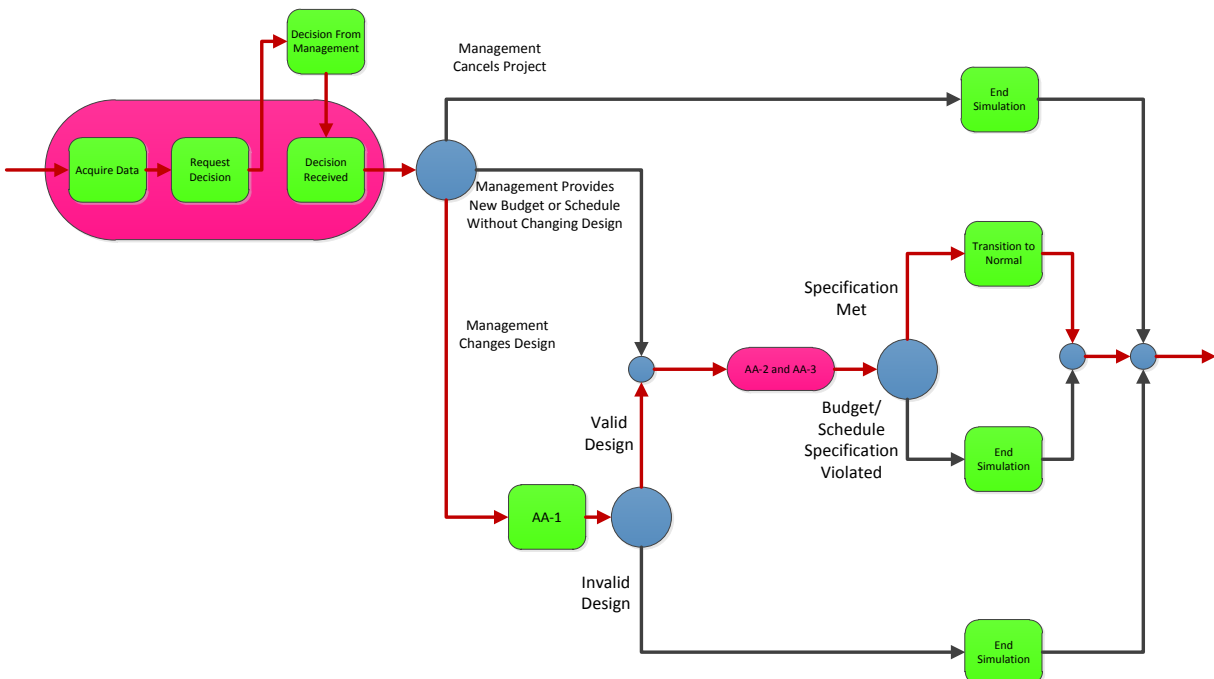


- Changes Requested from management to solve conflict

- Management Submits Design Change

- Event 4:

- CxG-5 Activated to Continue



- Event 5:

- Transition to **Normal** Major Context → True

- Major Context = **Normal**
 - Normal Action Function Activated
 - Event 6:
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 1 – 420
- Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 3.2: This test pushes the limits of the prototype by introducing late designs, causing uncertainty. The original prototype has not been designed with lateness of designs in mind. In this test, the rocket design selection lags behind schedule. The selection (design) is not submitted by the DA until Day 88. The control system cannot be ordered because unless the engine is selected. Alternatively, CS-400 could be ordered, but it is more expensive and has long lead times.

Initial Design:

Number of stages: Not available at start
Engine selected: Not available at start
Fin Material: Not available at start
Cone Material: Not available at start
Body Length: Not available at start
Nose Cone Length: Not available at start
Fin Width: Not available at start
Body Diameter: Not available at start
Fin Length: Not available at start
Fin Height: Not available at start
Payload Weight: 35 g
Parachute Diameter: Not available at start
Control System: Not available at start

Feasibility of Design: Unknown

Predicted total cost of project:

Engine	Unknown at start
Control System	Unknown at start
Fin and Nosecone Material:	Unknown at start
Premium for multiple stages:	Unknown at start
Total Cost:	Unknown at start

Predicted schedule:

Engine:	Unknown at start
Control System:	Unknown at start
Fin and Nosecone Material:	Unknown at start
Premium for multiple stages:	Unknown at start
Total FDD	Unknown at start

Storyboard: *None*

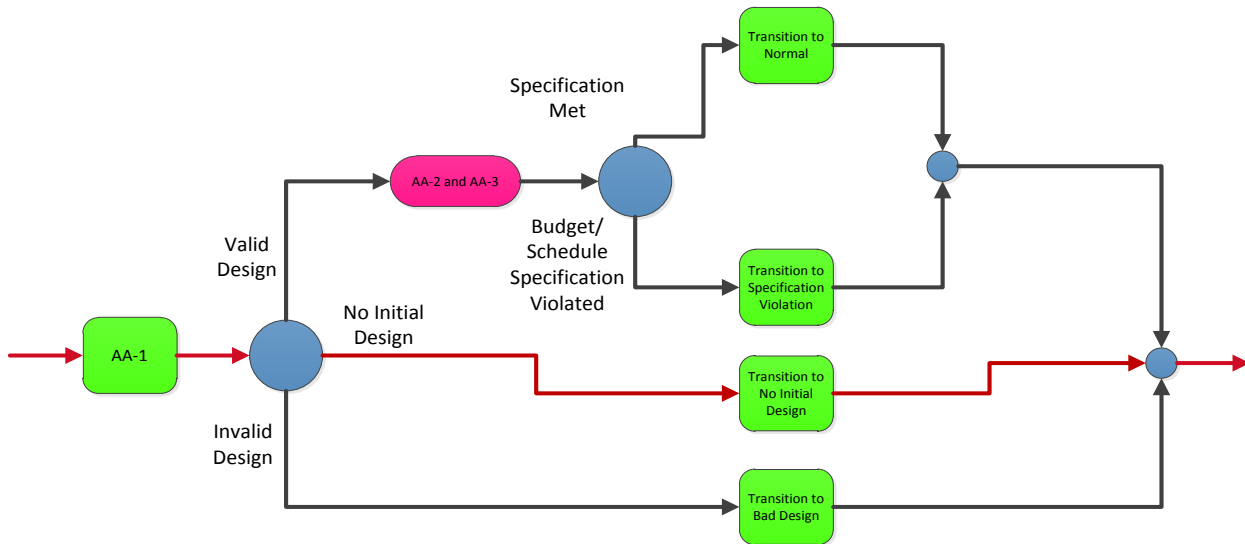
Expected Results: *The simulator starts with no initial rocket design. This will cause the following occurrences: First, the PM-agent will start in the **Initialize** Major Context . Here, the PM-agent will realize that an initial design has not been submitted. This will cause the PM-agent to transition to **No-Initial-Design** Major Context. The PM-agent will give the DA 14 days to submit a design before impasse is declared. Since by day 14, no design has been submitted, the PM-agent will transition to **Impasse** Major Context. Here, upper management is given a chance to intervene and submit a design before the project is cancelled. Since upper management does not have a design for the PM-agent, the project is cancelled. Day 88 will now never occur because the DA was too late with coming up with a design to meet specification.*

Notes: In the original system, initial values were chosen for the fins, nose cone, body length, nose cone length, fin width, body diameter, fin length, fin height, parachute diameter, and control system. However, since we are using NASA's rocket simulator to select the parts for our rocket's design, we were not able to create a rocket which was just missing engines. Therefore, the mission was slightly modified to show a situation where no initial design was available in order to maintain the spirit of the original test.

Test Results:

- ✧ Day 0:
 - Event 1:
 - Initial Specification Given to PM-agent Without a Design
 - **Initialize** Major Context Activated

- CxG-1 Activated



- Event 2:

- Transition to **No-Initial-Design** Major Context → True
- Activate No Initial Design Action Function → True
- No Initial Design Action Function Activated

- Event 3:

- No Initial Design Action Function Activated

- ✧ Day 1 – 13

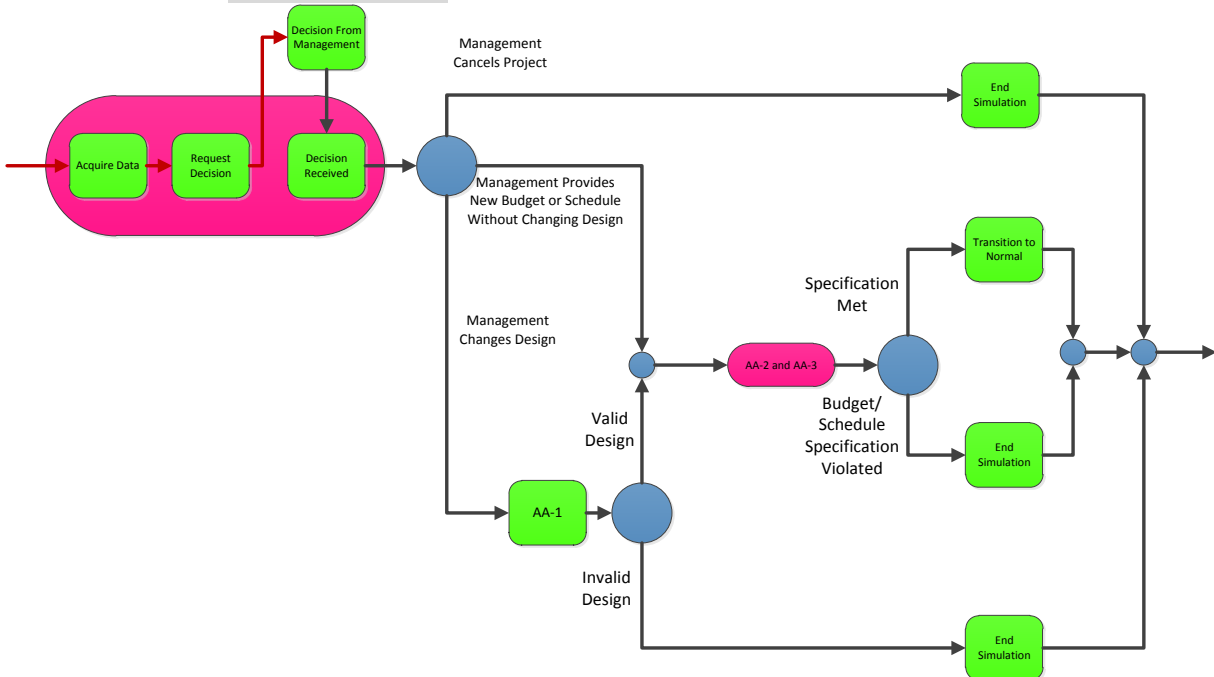
- No Initial Design Action Function Activated

- ✧ Day 15

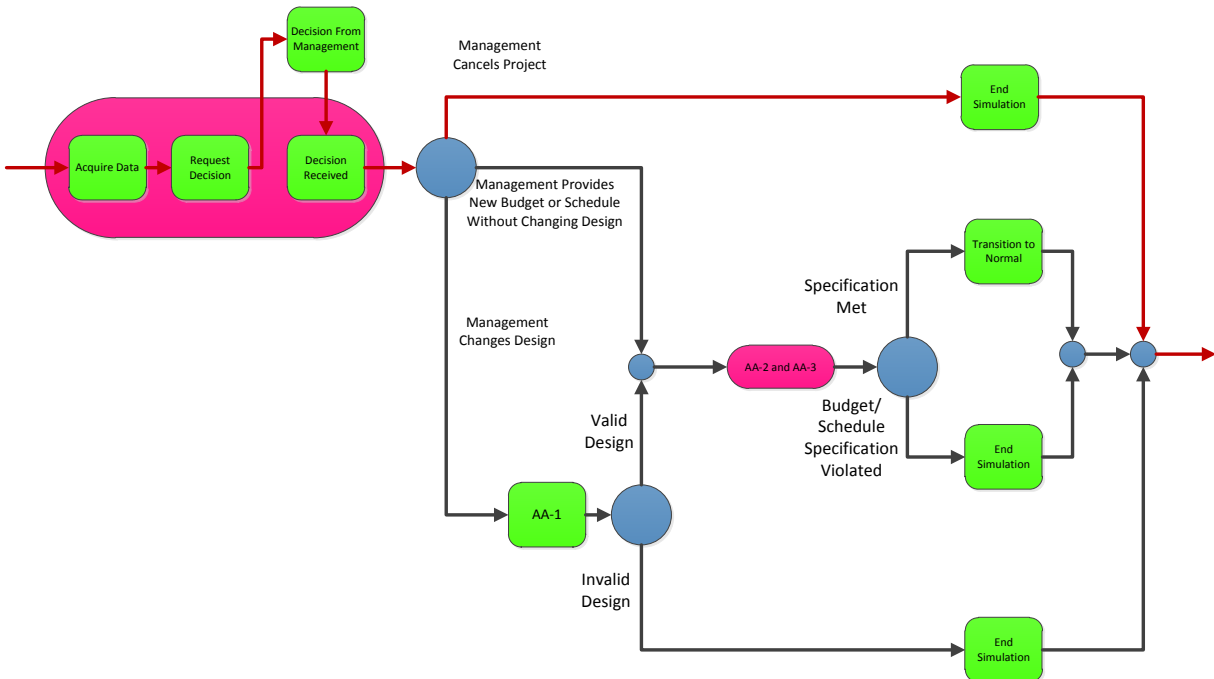
- Event 1:

- Transition To **Impasse** Major Context → True
- Major Context = **Impasse**

- CxG-5 Activated



- Changes Requested from management to solve conflict
- Management Submits No Changes Implies a Cancellation of the Mission
- CxG-5 Activated to Continue



- Event 2:
 - Mission Canceled

Conclusion(s): Results in agreement with the expected results. Test successful.

6.1.3.1 Test Set 3 Conclusions

Now that all the tests cases in the original system have been run, it can be confirmed that the new, more contextualized system can successfully handle all of the previous tests. By increasing the contextualization of the PM-agent, the handling of these situations became a simple matter. Test 3.1 and 3.2 can be handled successfully because of the contextualization added by **Initialize** Major Context, **No-Initial-Design** Major Context, and the new CxG-1. With the increased contextualization of the PM-agent added by these components, we can now handle a multitude of startup conditions that could only be done easily with the use of contextualization. Now that the all the previous cases have been handled successfully, we move on to Section 6.2 where we test our system with some new scenarios that the previous system was not designed to handle.

6.2 The New Test Cases

We now put our system to the test by providing even more complex scenarios than the previous tests. For this set of tests, first we repeat Test 3.2 with the expectation that an adequate design will be provided within the two week waiting period. Afterwards, we will be testing situations including multiple simultaneous events, with and without a design change, and situations involving a bad initial design caused by an inadequate design, in which the rocket could not reach the specified altitude. Unlike the original test cases, we decided all specifications for the rocket will be listed in the test description. Additionally, in order to maintain consistency with Gonzalez et al., we continue to follow his format. Explanations of the changes made are listed in a “notes section” listed under each test.

6.2.1 Test Case with Late Design Provided Within 2 Weeks of Start

Test 3.3: This test reworks Test 3.2 by introducing late design which is provided within 2 weeks of the start date. The new system now has the ability to handle late designs and this test is designed to show this capability. The new design will be submitted by the DA on day 10. We will use the design used in Test 1.1. Also, for our initial specification, it will also be the same as used by Test 1.1. This gives us a max budget of 30 MYen, a max FDD of 10 months, a payload of 49g, and a minimum altitude to reach of 0m. Just as in Test 1.1, a minimum altitude of 0m satisfies our requirements since no altitude specification is needed for this test. Just as done in [Gonzalez, 2007], we have indicated changes from Test 3.2 by using italics.

Initial Design:

Number of stages: Not available at start
Engine selected: Not available at start
Fin Material: Not available at start
Cone Material: Not available at start
Body Length: Not available at start
Nose Cone Length: Not available at start
Fin Width: Not available at start
Body Diameter: Not available at start
Fin Length: Not available at start
Fin Height: Not available at start
Payload Weight: *49 g*
Parachute Diameter: Not available at start
Control System: Not available at start

Feasibility of Design: Unknown

Predicted total cost of project:

Engine	Unknown at start
Control System	Unknown at start
Fin and Nosecone Material:	Unknown at start
Premium for multiple stages:	Unknown at start
Total Cost:	Unknown at start

Predicted schedule:

Engine:	Unknown at start
Control System:	Unknown at start
Fin and Nosecone Material:	Unknown at start
Premium for multiple stages:	Unknown at start
Total FDD	Unknown at start

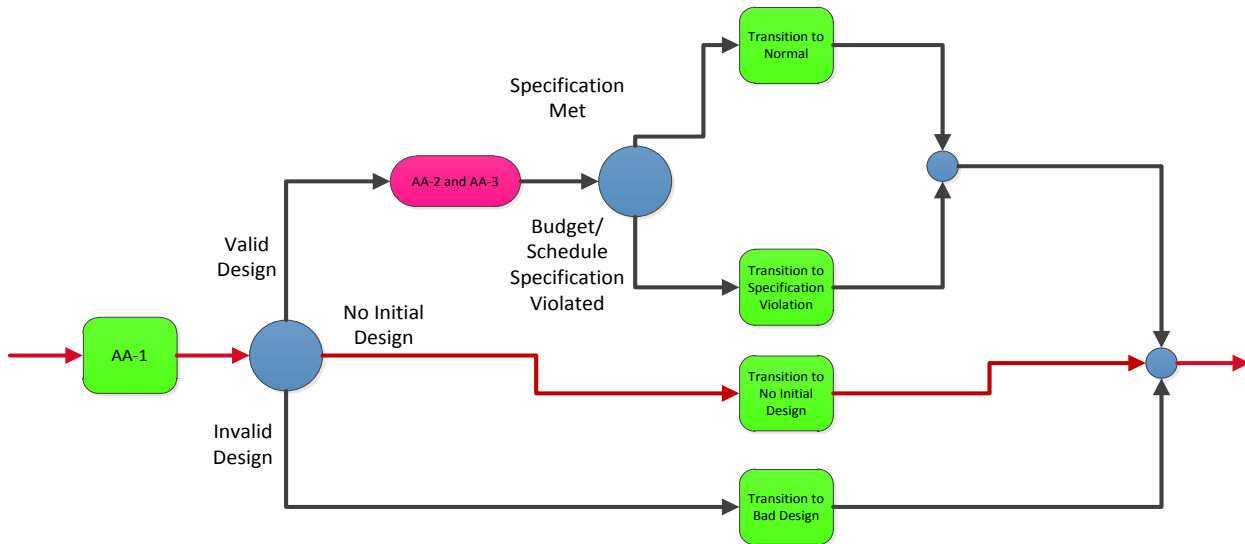
Storyboard: None

Expected Results: The simulator starts with no initial rocket design. This will cause the following occurrences. First, the PM-agent will start in the **Initialize** Major Context. Here, the PM-agent will realize that an initial design has not been submitted. This will cause the PM-agent to transition to **No-Initial-Design** Major Context. The PM-agent will give the DA 14 days to submit a design before impasse is declared. *On day 10, the DA will submit the design used in Test 1.1. This will cause the PM-agent to transition to **Initial Design** Major Context. From here the PM-agent will check the design feasibility. The PM-agent will recognize the design is able to meet the specified altitude however, because of the 10 day delay in receiving the design, it will not meet the FDD. This will cause the PM-agent to transition to **Specification-Violation** Major Context where AA-4 will be activated to take corrective action. Paying overtime for 10 days, the budget will be returned to spec. The PM-agent will then transition to **Normal** Major Context where the system will remain for the end of the simulation.*

Test Results:

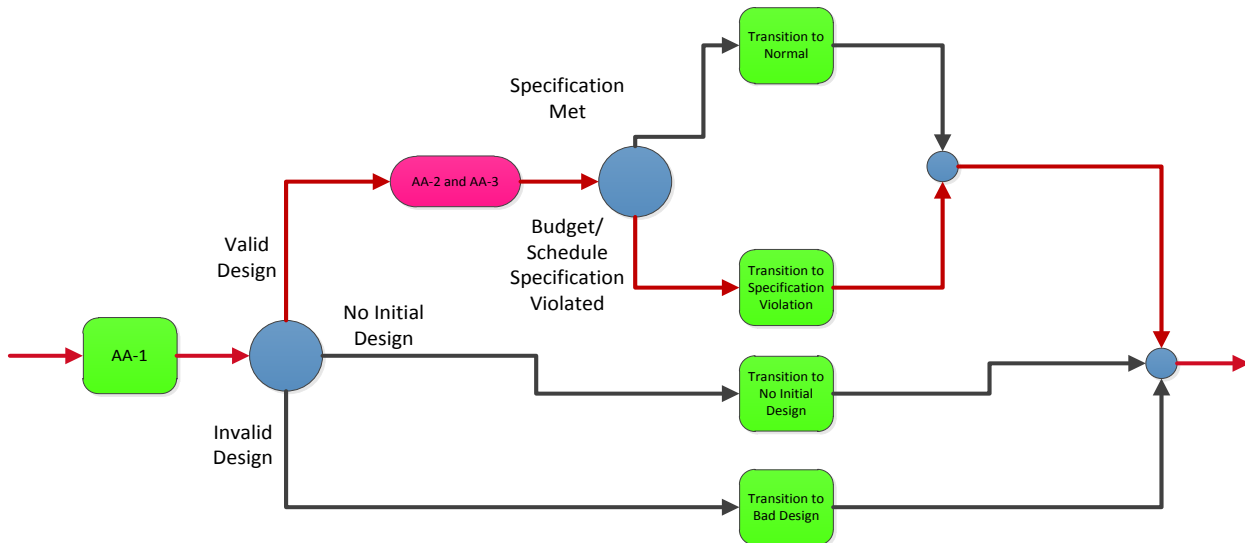
- ✧ Day 0:
 - Event 1:
 - Initial Specification Given to PM-agent Without a Design
 - **Initialize** Major Context Activated

- CxG-1 Activated



- Event 2:
 - Transition to **No-Initial-Design** Major Context → True
 - Activate No Initial Design Action Function → True
 - No Initial Design Action Function Activated
- Event 3:
 - No Initial Design Action Function Activated
- ✧ Day 1 – 9
 - No Initial Design Action Function Activated
- ✧ Day 10
 - Event 1:
 - No Initial Design Action Function Activated
 - New Design Submitted by DA
 - Event 2:
 - Transition to **Initialize** Major Context → True
 - Major Context = **Initialize**

- CxG-1 Activated



✧ Day 11

- Event 1:
 - Transition to **Specification-Violation** Major Context → True
 - Major Context = **Specification-Violation**
 - AA-4 Activated
 - Overtime Paid for 10 Days
 - Cost Increased by 0.5 MYen
 - Project Returned to Spec.
 - Transition to Normal Declared
- Event 2:
 - Transition to **Normal** Major Context → True
 - Major Context = **Normal**
 - Normal Action Function Activated
- Event 3:
 - Major Context = **Normal**
 - Normal Action Function Activated
- Day 12 - 300
 - Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

6.2.2 Text Case with Late Design Provided Within 2 Weeks of Start Conclusions

As can be seen by Test 3.3, the higher contextualization provides the means to handle various situations in which the original system could not handle. For this last scenario, we are able to show that the new system with the no initial design capabilities is not just limited to allowing the project to be cancelled. With the higher contextualized system, we can now have a DA submit a design within 2 weeks of starting a project as was done on day 10 of Test 3.3. Test 3.3 shows that by using the CxBR paradigm, the PM-agent was able to handle such situations easily and efficiently which the original, less contextualized system could not.

6.2.3 Test Cases Involving Multiple Simultaneous Events

Test 4.1: This test case was originally dubbed Test 1.6 in the documentation of the original system. However, since the original system should have been incapable of handling this situation, we decided it best to move it to Test 4.1 as this is really a new test case. For this test, we evaluated the PM's ability to handle both schedule and cost increase external events at the same time. This test was a scenario in which the system does not become over-constrained. In addition, for this test we used the following values for the test specification: Max Budget was set at 30 MYen and the FDD was set at 10 months. No attitude specification was required since altitude is not a concern of this test; therefore, we set the minimum required altitude to 0 m.

Initial Design:

Number of stages: One
Engine selected: A8-3
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 33 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 10.0 cm
Fin Height: 0.0 cm
Payload Weight: 49 g
Parachute Diameter: 1ft (default)
Control System: ACS-100

Feasibility of Design: The design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	1.0 MYen
Control System:	3.5 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	29.5 MYen

Predicted Schedule:

Engine:	2 mo. construction; 1 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 4 months = 10 months

Storyboard: On day 40, the engine manufacturer informs the PM-agent that the schedule for the engine has been delayed 130 days beyond the original delivery date. 2) On day 100, the control system manufacturer announces a price increase of 1 MYen. 3) On day 120 the control system manufacturer announces 2 simultaneous external events: a delay of 10 days in their delivery plus a cost increase of 3MYen.

Expected Result: After **external event #1** is detected, the PM-agent is expected to act just as it did in Test 1.5 and 1.6. The PM-agent will transition to **External-Event** Major Context upon detection of external event. Upon evaluating the situation, it determines the specification has been violated. The PM-agent will then transition to **Specification-Violation** Major Context where AA-4, the corrective action Auxiliary Agent, will be activated to apply the schedule reduction operator which uses overtime payments to shorten delivery.

Revised Engine schedule: 2 months construction + 130 days delay = 6.33 months delivery time

Schedule reduction operator: $50 \text{ KYen} \times 10 \text{ days overtime} = 0.5 \text{ MYen}$; $29.5 \text{ MYen} + 0.5 \text{ MYen} = 30 \text{ MYen}$; $30 \text{ MYen} \leq 30 \text{ MYen (Max Budget)} \rightarrow \text{OK}$

Schedule reduced by 10 days = $190 \text{ days} - 10 \text{ days} = 180 \text{ days}$; $6 \text{ months} \leq 6 \text{ months (Max FDD)} \rightarrow \text{OK}$

The PM-agent should now realize it is back in spec. and should transition to the **Normal** Major Context.

External event #2 introduces a subsequent cost increase that causes the PM-agent to transition to **External-Event** Major Context. Here it is determined that the budget has been exceeded by 1 MYen.

$$30 \text{ MYen} + 1 \text{ MYen} = 31 \text{ MYen}, > 30 \text{ MYen limit.}$$

Upon this realization, the PM-agent will transition to the **Specification-Violation** Major Context where it will call upon AA-4, the corrective action Auxiliary Agent. By applying the operator that reduces cost by 10% by reducing overhead staff:

$$31 \text{ MYen} \times 0.10 = 3.1 \text{ MYen.}$$

$$31 \text{ MYen} - 3.1 \text{ MYen} = 27.9 \text{ MYen,} \rightarrow \text{OK.}$$

After corrective action is taken, PM-agent should realize that all is once again within spec. and transition back to **Normal** Major Context.

New schedule = 300 days; New cost = 27.9 MYen.

External event #3 and External Event #4 together introduces another schedule delay of 10 days plus a cost increase of 3MYen. This will cause the PM-agent to transition to **Multiple-Simultaneous-Events** Major Context. Here, the PM-agent will decide to transition to **External-Event** Major Context since none of the events that occurred were design changes. Here, the PM-agent determines there is no schedule conflict; however, there is a cost conflict. AA-4 is called to resolve the issue.

Revised schedule: The control system will now have a delivery date of 190 days. However, since this is equal to the time it will take for the delivery of the engine with its delay, there will be no change in the time it will take for the completion of the rocket.

Revised Cost: 27.9 MYen + 3 MYen = 30.9 MYen

Now apply the cost reduction operator for the second time to reduce cost by an additional 5% through further reductions in overhead staff.

$30.9 \text{ MYen} \times 0.05 = 1.545 \text{ MYen}$.

$30.9 \text{ MYen} - 1.545 \text{ MYen} = 29.355 \text{ MYen} (< 30 \text{ MYen} \rightarrow \text{OK})$.

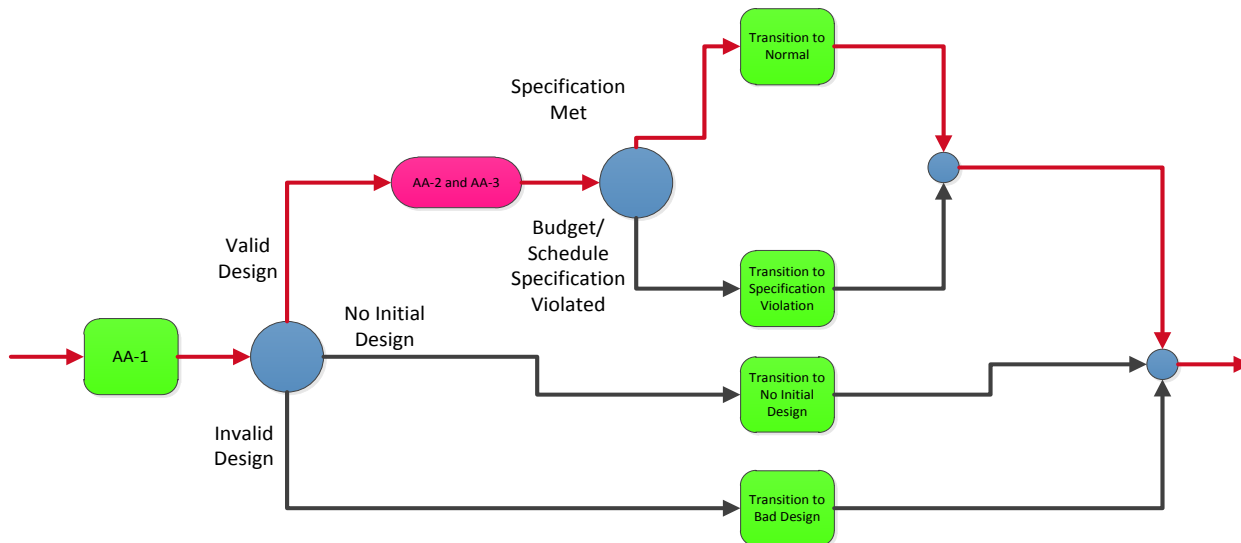
After realizing that all is within spec, PM-agent should transition back to **Normal** Major Context for the rest of the simulation.

Test Results:

✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



○ Event 2: Transition To **Normal** Major Context → True

○ Event 3: Major Context **Normal** Activated

✧ Day 1 – 39

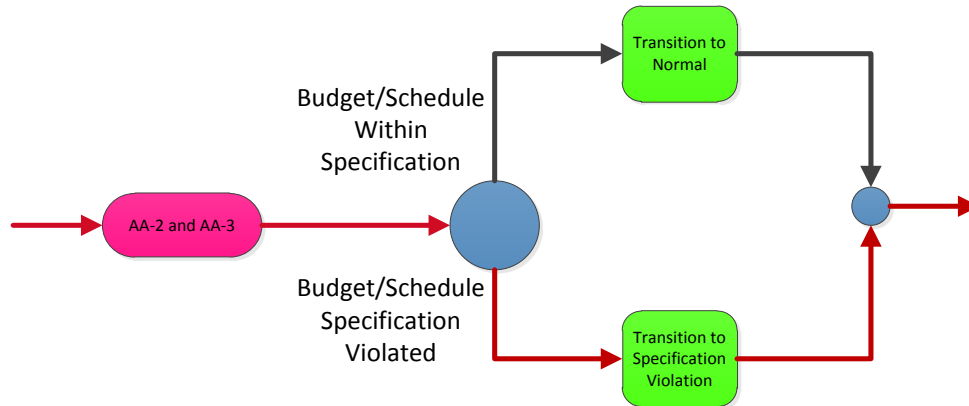
○ Major Context = **Normal**

○ Normal Action Function Activated

✧ Day 40

○ Event 1:

- External Event Announced – 130 Day Delay to Engine Delivery
- Transition to **External-Event** Major Context → True
- CxG-3 Activated



✧ Day 41

○ Event 1:

- Transition to **Specification-Violation** Major Context → True
- AA-4 Activated
 - Overtime Paid – 10 days
 - Cost Increase of 0.5 MYen
 - Project Returned to Spec.
 - Transition to Normal Declared

○ Event 2: Transition To **Normal** Major Context → True

○ Event 3: Major Context **Normal** Activated

✧ Day 42 - 99

○ Major Context = **Normal**

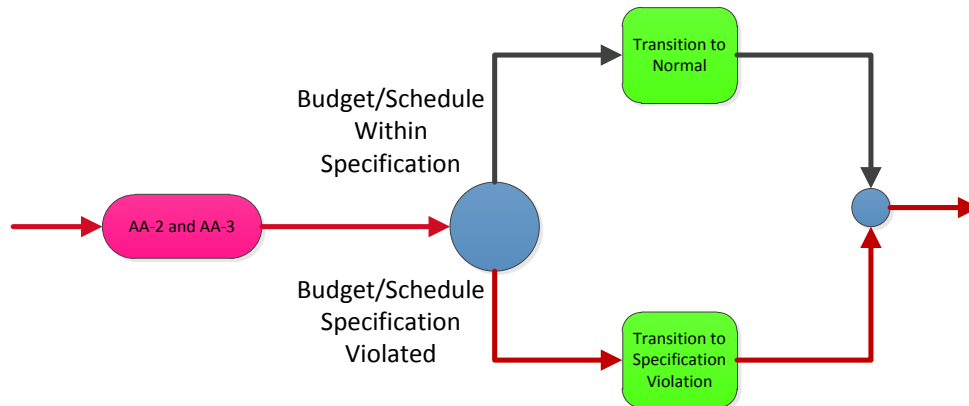
○ Normal Action Function Activated

✧ Day 100

○ Event 1:

- External Event Announced – 1 MYen Cost Increase
- Transition to **External-Event** Major Context → True
- Major Context = **External-Event**

- CxG-3 Activated



✧ Day 101

○ Event 1:

- Transition to **Specification-Violation** Major Context → True
- Major Context = **Specification-Violation**
- AA-4 Activated
- Overhead Reduced – First Time
- 10% Cost Reduction = 3.1 MYen
- Project Returned to Spec.
- Transition to Normal Declared

○ Event 2: Transition To **Normal** Major Context → True

○ Event 3: Major Context **Normal** Activated

✧ Day 102 – 119

○ Major Context = **Normal**

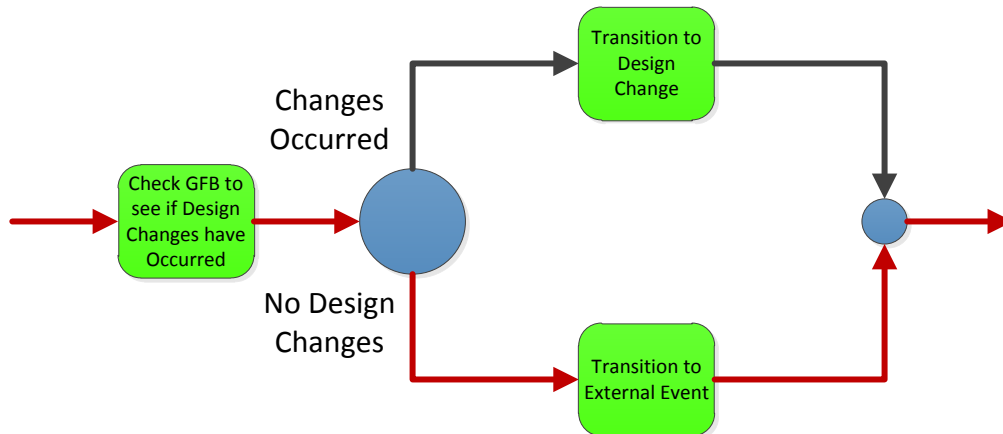
○ Normal Action Function Activated

✧ Day 120

○ Event 1:

- External Event Occurred – Control System Delivery Delay of 10 Days
- External Event Occurred – Cost Increase of 3 MYen
- Transition to **Multiple-Simultaneous-Events** Major Context → True
- Major Context = **Multiple-Simultaneous-Events**

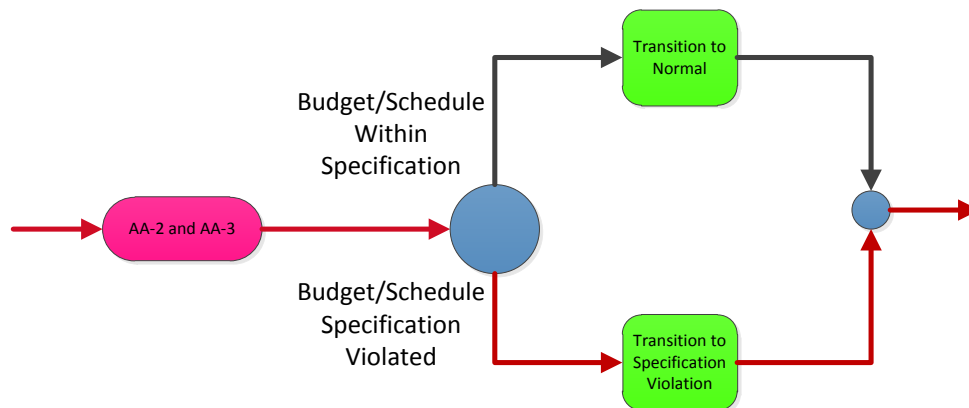
- CxG-4 Activated



✧ Day 121

- Event 1:

- Transition to **External Event** Major Context → True
- Major Context = **External-Event**
- CxG-3 Activated



- Event 2:

- Transition to **Specification Violation** Major Context → True
- Major Context = **Specification-Violation**
- AA-4 Activated
- Overhead Reduced – Second Time
- 5% Cost Reduction = 1.545 MYen
- Project Returned to Spec.
- Transition to Normal Declared

- Event 3:

- Transition to **Normal** Major Context → True
- Major Context = **Normal**
- Normal Action Function Activated

- Event 4:

- Major Context = **Normal**

- Normal Action Function Activated
- ✧ Day 122 - 300
 - Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 4.2: Again, just as in Test 4.1, this test case originally had another title. This test was originally dubbed Test 1.6 in the documentation of the original system. However, just as before, since the original system should have been incapable of handling this situation, we decided it best to move it to Test 4.2 as another new test case. This test is designed to evaluate the prototype's reaction to schedule changes and cost increases in the same test; however, the situation is now over-constrained, leading to an impasse. The initial design, schedule and costs are the same as those in Test 4.1. We have repeated them again below for completeness as was done in [Gonzalez, 2007].

Initial Design:

Number of stages: One
Engine selected: A8-3
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 33 cm
Nose Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 10.0 cm
Fin Height: 0.0 cm
Payload Weight: 49 g
Parachute Diameter: 1 ft (default)
Control System: ACS-100

Feasibility of Design: As in Test 4.1, the design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	1.0 MYen
Control System:	3.5 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	0.0 MYen
Total Cost:	29.5 MYen

Predicted Schedule:

Engine:	2 mo. construction; 1 mo. installation
----------------	--

Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	0.0 months
Total FDD	6 months + 4 months = 10 months

Storyboard: *On day 40, the engine manufacturer informs the PM-agent that the schedule for the engine has been delayed 130 days beyond the original delivery date. 2) On day 100, the control system manufacturer announces a price increase of 1 MYen. 3) On day 120 manufacturer announces 2 simultaneous external events: a delay of 20 days in their delivery plus a cost increase of 2MYen.*

Expected Result: *The first two external events do not cause an over-constrained situation. They are the same as in Test 4.1. However, external event #3 presents an unequivocal over-constraining element that leads to an impasse.*

Expected Result: After **external event #1** is detected, PM-agent is expected to act just as it did in Test 1.5, 1.6, and 4.1. The PM-agent will transition to **External-Event** Major Context upon detection of external event. Upon evaluating the situation, it determines the specification has been violated. The PM-agent will then transition to **Specification-Violation** Major Context where AA-4, the corrective action Auxiliary Agent, will be activated to apply the schedule reduction operator which uses overtime payments to shorten delivery.

Revised Engine schedule: 2 months construction + 130 days delay = 6.33 months delivery time

Schedule reduction operator: 50 KYen x 10 days overtime = 0.5 MYen; 29.5MYen + 0.5MYen = 30MYen; 30MYen ≤ 30MYen (Max Budget) → OK

Schedule reduced by 10 days = 190 days – 10 days = 180 days; 6 months ≤ 6 months (Max FDD) → OK

The PM-agent should now realize it is back in spec. and should transition to the **Normal** Major Context.

External event #2 introduces a subsequent cost increase that causes the PM-agent to transition to **External-Event** Major Context. Here it is determined that the budget has been exceeded by 1 MYen.

$$30 \text{ MYen} + 1 \text{ MYen} = 31 \text{ MYen}, > 30 \text{ MYen limit.}$$

Upon this realization, the PM-agent will transition to the **Specification-Violation** Major Context where it will call upon AA-4, the corrective action Auxiliary Agent. By applying the operator that reduces cost by 10% by reducing overhead staff:

$$31 \text{ MYen} \times 0.10 = 3.1 \text{ MYen.}$$

$$31 \text{ MYen} - 3.1 \text{ MYen} = 27.9 \text{ MYen,} \rightarrow \text{OK.}$$

After corrective action is taken, PM-agent should realize that all is once again within spec. and transition back to **Normal** Major Context.

New schedule = 300 days; New cost = 27.9 MYen.

***External event #3 and External Event #4** together introduces another schedule delay of 20 days plus a cost increase of 3MYen. This will cause the PM-agent to transition to **Multiple-Simultaneous-Events** Major Context. Here, the PM-agent will decide to transition to **External-Event** Major Context since none of the events that occurred were design changes. Here, the PM-agent determines there a schedule conflict but there is no cost conflict. AA-4 is called to resolve the issue.*

Revised schedule: The control system will now have a delivery date of 200 days. This is now greater than the time it will take for the delivery of the engine with its delay. Therefore, the FDD

is now exceeded by 10 days.

Revised Schedule: 300 days + 10 days = 310 days

Revised Cost: 27.9 MYen + 2 MYen = 29.9 MYen

AA-4 will now evaluate the situation. First it will look at how much overtime will need to be paid in order to resolve the schedule conflict. AA-4 will then realize that the budget will be exceeded by trying to apply the overtime needed to complete the mission. Impasse will be declared.

*PM-agent transitions to **Impasse** Major Context and externally requests for either a change in the projects specification or the design.*

Upper management will extend the FDD to 320 days.

*Entering acceptable values resolves the over-constraint and permits PM-agent to transition back to **Normal** Major Context.*

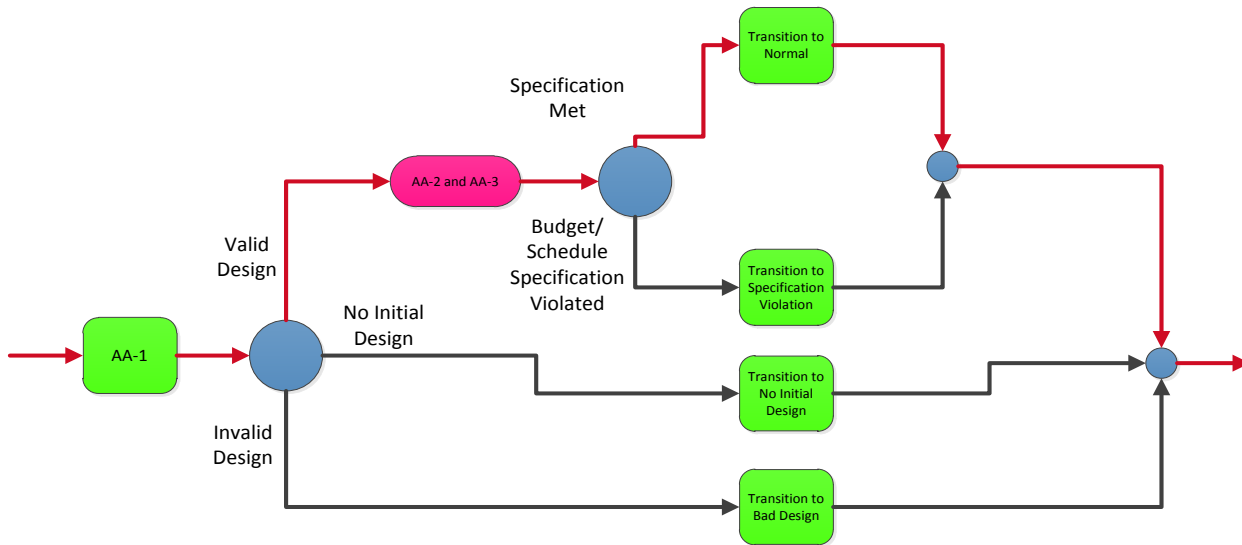
Test Results:

✧ Day 0:

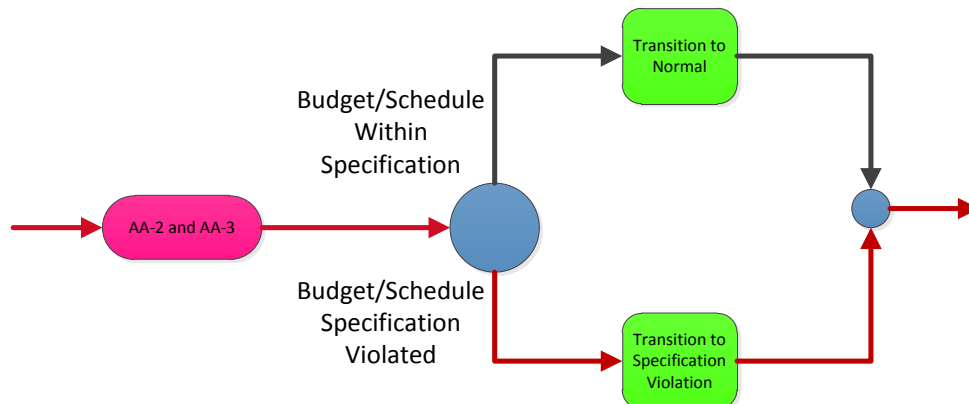
○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated

- CxG-1 Activated

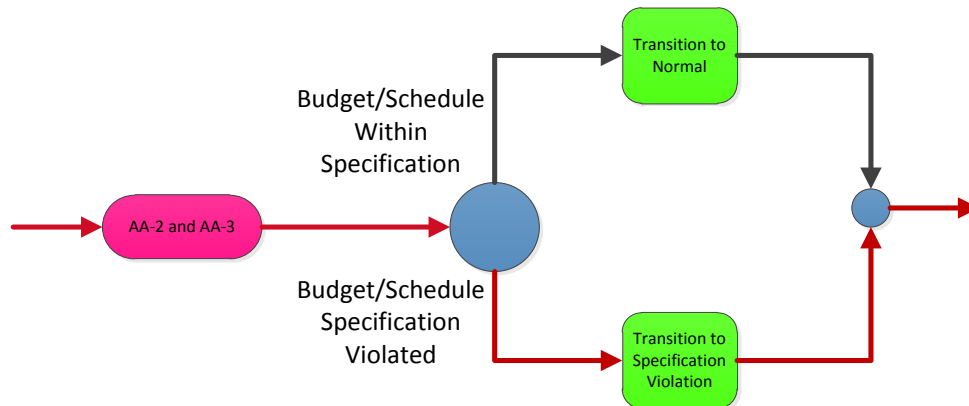


- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated
- ✧ Day 1 – 39
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 40
 - Event 1:
 - External Event Announced – 130 Day Delay to Engine Delivery
 - Transition to **External-Event** Major Context → True
 - CxG-3 Activated



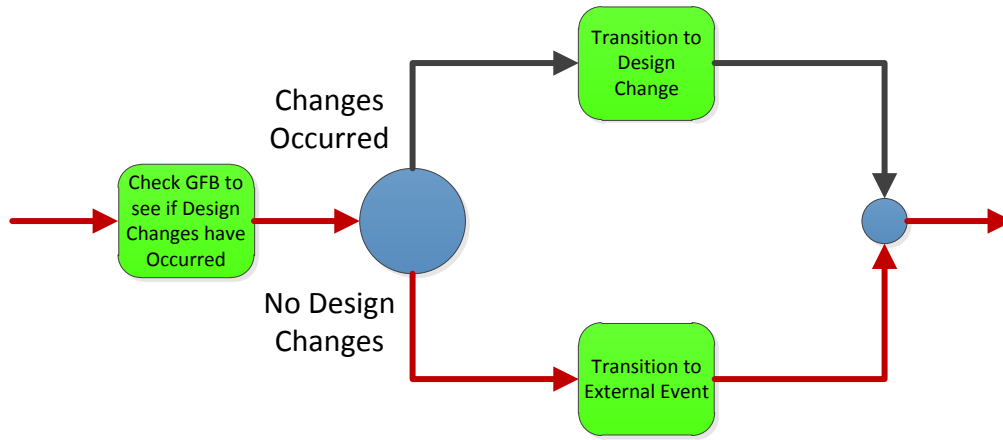
- ✧ Day 41
 - Event 1:
 - Transition to **Specification-Violation** Major Context → True
 - AA-4 Activated

- Overtime Paid – 10 days
 - Cost Increase of 0.5 MYen
 - Project Returned to Spec.
 - Transition to Normal Declared
- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated
- ✧ Day 42 - 99
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 100
 - Event 1:
 - External Event Announced – 1 MYen Cost Increase
 - Transition to **External-Event** Major Context → True
 - Major Context = **External-Event**
 - CxG-3 Activated



- ✧ Day 101
 - Event 1:
 - Transition to **Specification-Violation** Major Context → True
 - Major Context = **Specification-Violation**
 - AA-4 Activated
 - Overhead Reduced – First Time
 - 10% Cost Reduction = 3.1 MYen
 - Project Returned to Spec.
 - Transition to Normal Declared
 - Event 2: Transition To **Normal** Major Context → True
 - Event 3: Major Context **Normal** Activated
- ✧ Day 102 – 119
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 120
 - Event 1:
 - External Event Occurred – Control System Delivery Delay of 20 Days
 - External Event Occurred – Cost Increase of 2 MYen

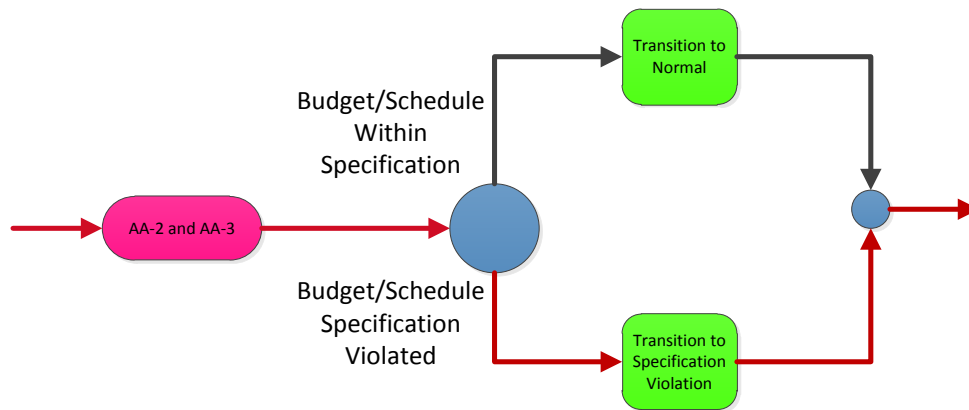
- Transition to **Multiple-Simultaneous-Events** Major Context → True
- Major Context = **Multiple-Simultaneous-Events**
- CxG-4 Activated



✧ Day 121

○ Event 1:

- Transition to **External-Event** Major Context → True
- Major Context = **External-Event**
- CxG-3 Activated



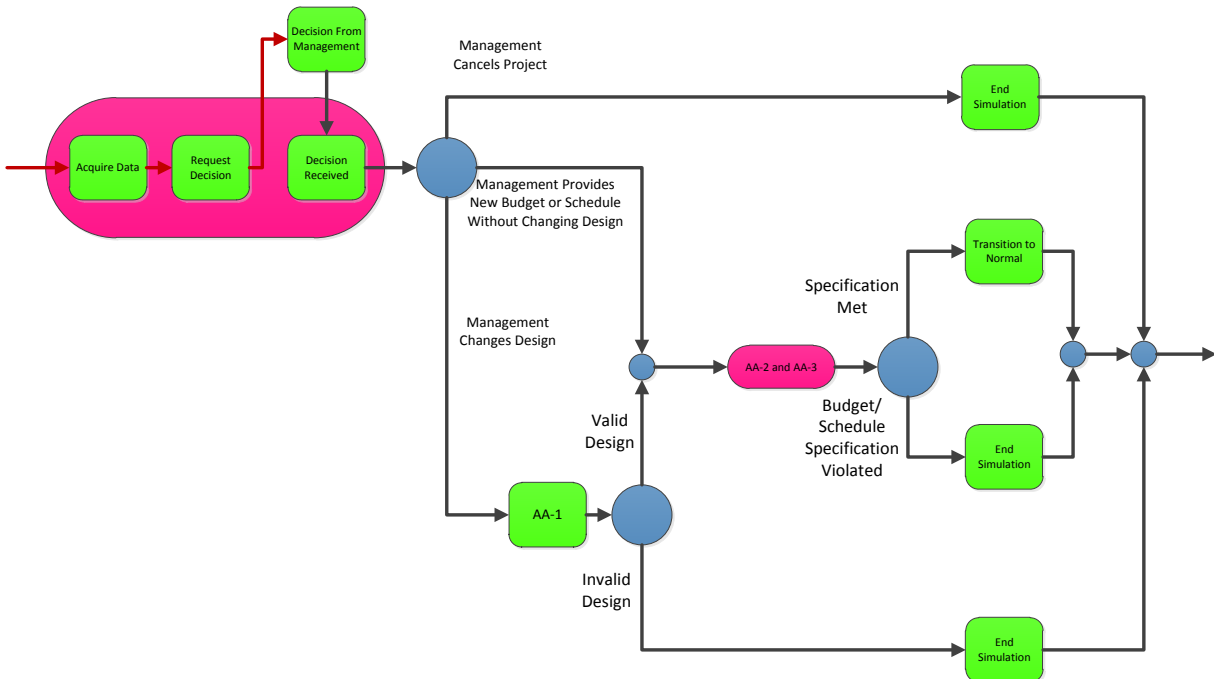
○ Event 2:

- Transition to **Specification-Violation** Major Context → True
- Major Context = **Specification-Violation**
- AA-4 Activated
- Overtime Needed Will Cause a Violation of the Budget
- Transition to Impasse Declared

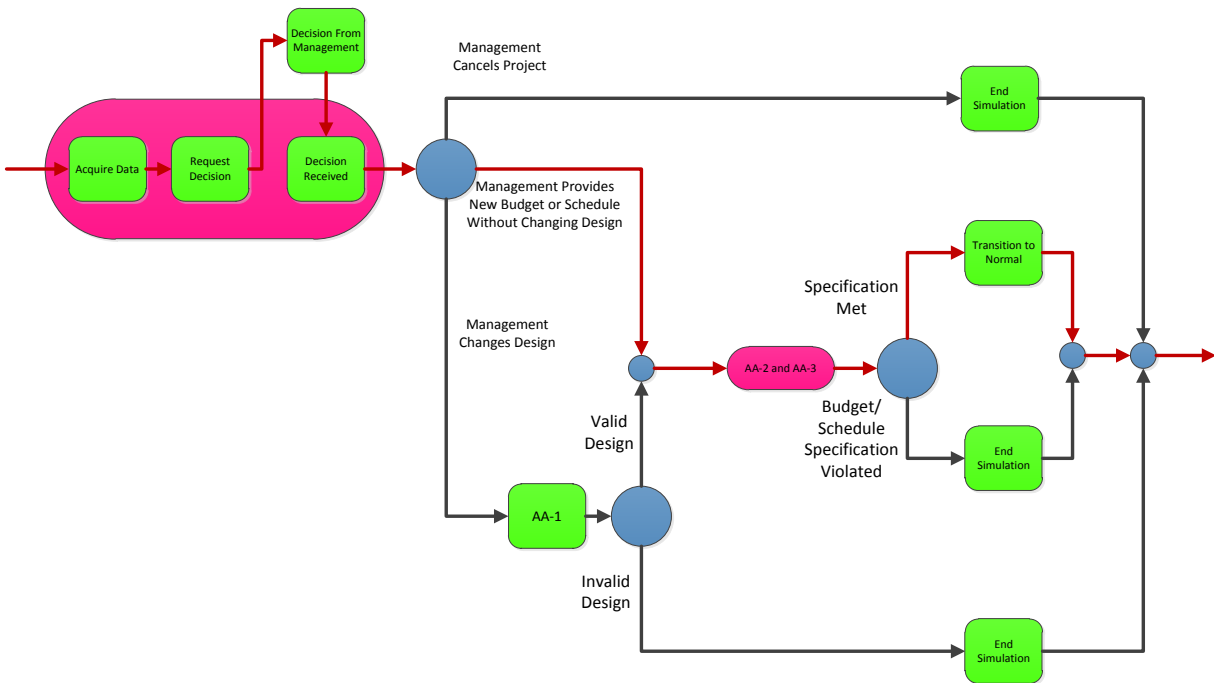
○ Event 3:

- Transition to **Impasse** Major Context → True

- Major Context = **Impasse**
- CxG-5 Activated



- Changes Requested from management to solve conflict
- Management Extends FDD to 310 days.
- Event 4:
 - CxG-5 Activated to Continue



- Event 5:
 - Transition to **Normal** Major Context → True
 - Major Context = **Normal**
 - Normal Action Function Activated
 - Event 6:
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 122 - 310
- Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 4.3: This test case expands upon the abilities of our PM-agent by demonstrating a new situation that goes beyond just the implementation multiple external events. In this test, we will show the PM's ability to now handle multiple external events as well as a design change simultaneously. In this test there will be a design change, a cost increase, and a delay which is independent of any specific part (a new feature, not part of the original system). This test will be a scenario in which the system does not become over-constrained. In addition, for this test we will use the following values for the test specification: Max Budget will be set at 45 MYen and the FDD will be set at 24 months. The customer requires a minimum altitude of 300 m while carrying a payload of 60g.

Initial Design:

Number of stages: Two
Engine selected: Stage 1 – B6-0, Stage 2 – B6-4
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 40 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 17.0 cm
Fin Height: 0.0 cm
Payload Weight: 60 g
Parachute Diameter: 1ft (default)
Control System: ACS-200

Feasibility of Design: The design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	4.0 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	1.0 MYen
Total Cost:	39.0 MYen

Predicted Schedule:

Engine:	4 mo. construction; 4 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	1.0 month installation
Total FDD	6 months + 8 months = 14 months

Storyboard: On day 40, the DA decides he want the rocket to be one stage instead of two. The DA proposes the use of engine C6-5 which will obtain the appropriate altitude. At the same time, in order to prevent a strike, a pay raise is given to the workers costing the company 10 MYen. In addition, a shortage of steel causes an overall delay to the project of 2 months.

Expected Result: External event #1, External Event #2 and Design Change #1 together will cause the PM-agent to transition to **Multiple-Simultaneous-Events** Major Context. Here, the PM-agent will recognize that a design change is part of one of the multiple events and will transition to **Design-Change** Major Context. In **Design-Change** Major Context, CxG-2 will be activated to determine the implications of the events that have occurred. The following will be reported:

The new design will reach the specified altitude.

New Engine Schedule:

1.33 months (today) + 6 months construction = delivery date of 7.33 months
3 months for installation

New Control System Schedule:

1.33 months (today) + 6 months construction = delivery date of 7.33 months
0 months for installation

New Predicted Schedule:

Engine:	7.33 mo. construction; 3 mo. installation
----------------	---

Control System:	7.33 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	0.0 month installation
Other Delay:	2 months
Total FDD	7.33 months + 6 months + 2 months = 15.33 months

15.33 Months \leq 24 Months Max FDD

New Total Cost:

Previous Cost of 39 MYen + 10 MYen for Strike Prevention + 10 MYen for New Engine + 3.5 MYen for New Control System = 62.5 MYen

62.5 MYen \leq 75 MYen Max Budget

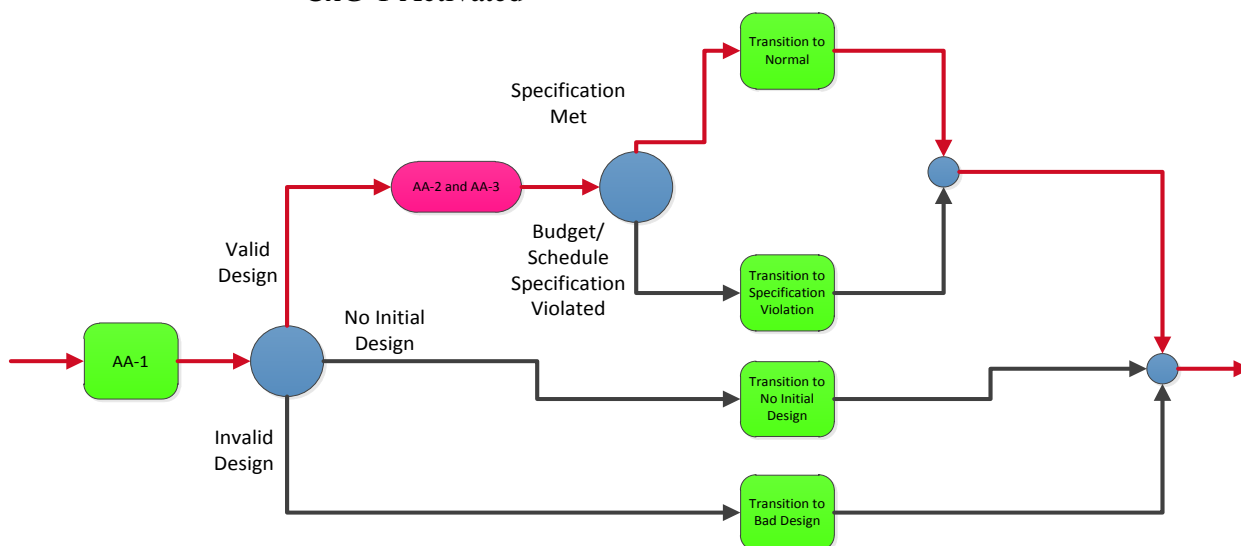
This will cause the PM-agent to recognize the Events did not put the project out of spec. and we can transition to normal. The project will then remain here until the rocket is completed on day 460.

Test Results:

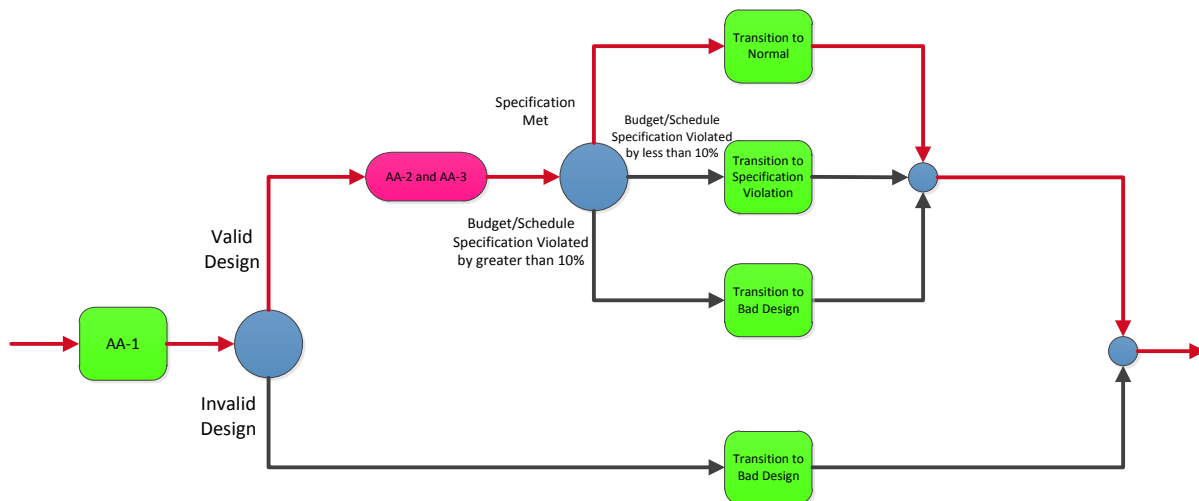
✧ Day 0:

○ Event 1:

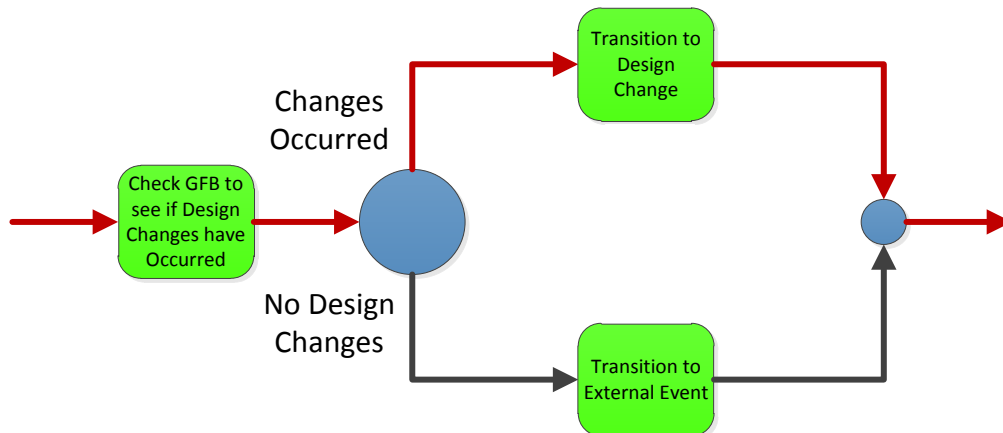
- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True
- Event 3: Major Context **Normal** Activated
- ✧ Day 1 – 39
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 40
 - Event 1:
 - New Design Submitted to PM
 - External Event Announced – Cost Increased By 10 MYen
 - External Event Announced – Other Delay of 60 Days
 - Transition to **Multiple-Simultaneous-Events** Major Context → True
 - Major Context = **Multiple-Simultaneous-Events**
 - CxG-4 Activated



- Event 2:
 - Transition to **Design-Change** Major Context → True
 - Major Context = Design-Change
 - CxG-2 Activated



- Event 3:
 - Transition to **Normal** Major Context → True
 - Major Context = **Normal**
 - Normal Action Function Activated
- Event 4:
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 41 - 460
 - Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

Test 4.4: This test case will take Test 4.2 not just a step, but a leap further. Here the PM-agent will go beyond handling just three events at the same time to handling nine events at the same time. The new higher contextualized PM-agent can handle infinite events at the same time, but, we deemed nine simultaneous events were sufficient for what a real PM might expect to see. In this test there will be a design change, four cost increases, and four delays. One delay will be on the control system. One delay will be on the engine. Two delays will be independent of any part. This test will be a scenario in which the system does not become over-constrained. In addition, we will use the following values for the test specification: Max Budget will be set at 45 MYen, the FDD will be set at 24 months, and a minimum altitude of 300 m while carrying a payload of 60 g will be required as in Test 4.2.

Initial Design:

Number of stages: Two
Engine selected: Stage 1 – B6-0, Stage 2 – B6-4
Fin material: 1/8" Plastic
Nose Cone material: Hollow Plastic
Body Length: 40 cm
Nose Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 17.0 cm
Fin Height: 0.0 cm
Payload Weight: 60 g
Parachute Diameter: 1ft (default)
Control System: ACS-200

Feasibility of Design: The design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	4.0 MYen
Fin Material:	5.0 MYen
Nosecone Material:	20.0 MYen
Premium for multiple stages:	1.0 MYen

Total Cost: 39.0 MYen

Predicted Schedule:

Engine:	4 mo. construction; 4 mo. installation
Control System:	6 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	1.0 month installation
Total FDD	6 months + 8 months = 14 months

Storyboard: On day 40, upper management decides they want their rocket to be able to carry larger payloads although they are not sure of how large. A **design change** occurs with the DA proposes Stage 2 to be change to D12-7 which will provide higher thrust enabling the rocket to support higher payloads. At the same time, four external events cause an increase in price and four external events cause delays extending the FDD. **External Event 1** is a payout to employees of 10 MYen to prevent a strike. Also, the time taken to work out the strike caused an overall delay of 1 month in which no work was able to be done. This will be defined as **External Event 2**. **External Event 3 and 4** are cost increases caused by an increased cost of steel because of steel shortages which will affect both the manufacturing of the engine and the control system. The two events cause increases of 5 MYen and 1 MYen respectively. **External Event 5** occurs because the fin manufacturer claimed they increased their cost by 1.5 MYen. In addition, the shortage of steel causes a delay to the construction of the Engines (**External Event 6**) and Control System (**External Event 7**) of 3 months and 2 months respectively. **External Event 8** causes an overall delay of 0.5 months when the customer and company have to go into negotiations to discuss the changes to the cost and schedule that occurred because of the previous events that happened.

Expected Result: The Nine Simultaneous Events together will cause the PM-agent to

transition to **Multiple-Simultaneous-Events** Major Context. Here, the PM-agent will recognize that a design change is part of one of the multiple events and will transition to **Design-Change** Major Context. In **Design-Change** Major Context, CxG-2 will be activated to determine the implications of the events that have occurred. The following will be reported:

The new design will reach the specified altitude.

New Engine Schedule:

1.33 months (today) + 8 months construction + 3 month delay = delivery date of 12.33 months
4 months for installation

New Control System Schedule:

1.33 months (today) + 6 months construction + 2 month delay = delivery date of 9.33 months
0 months for installation

Other Delays:

1 month + 0.5 months = 1.5 months

New Predicted Schedule:

Engine:	12.33 mo. construction; 6 mo. installation
Control System:	9.33 months delivery
Fin Material:	1 mo. construction; 1 mo. installation
Nosecone Material:	2 mo. construction; 2 mo. installation
Premium for multiple stages:	1.0 month installation
Other Delay:	1.5 months
Total FDD	12.33 months + 10.0 months + 1.5 months = 23.83 months

23.83 Months \leq 24 Months Max FDD

New Total Cost:

Previous Cost of 39 MYen + 12 MYen for New Engine + 10 MYen for Strike Prevention + 5 MYen for Engine Price Increase + 1 MYen for Control System Price Increase + 1.5 MYen for Fin Price Increase = 68.5 MYen

68.5 MYen \leq 75 MYen Max Budget

This will cause the PM-agent to recognize the Events did not cause the project to violate

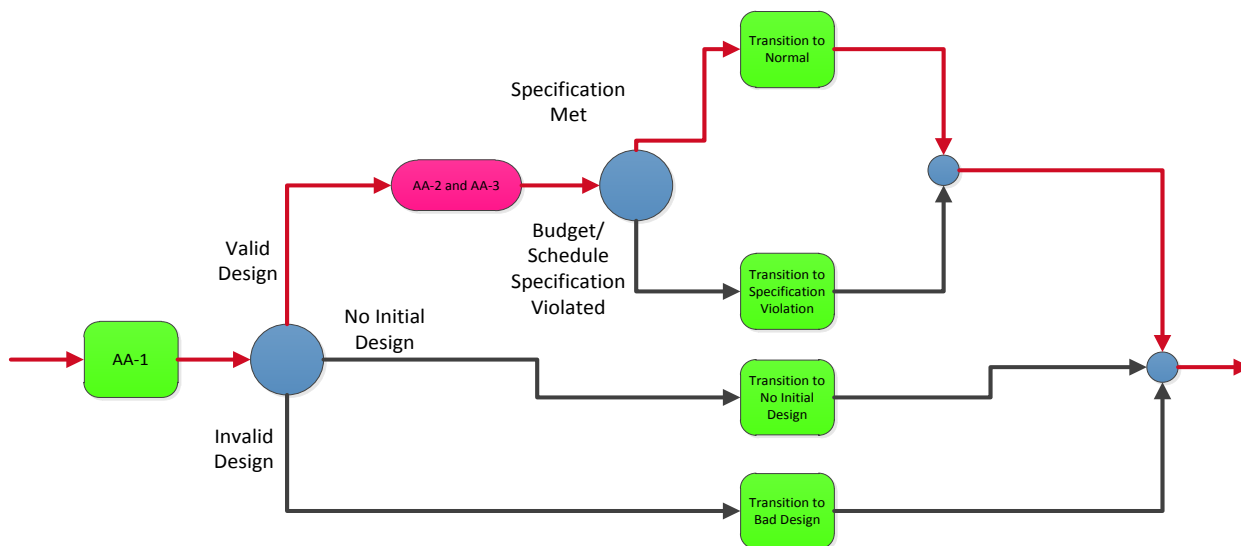
specification and it can transition to normal. The project will then remain in the **Normal** Major Context until the rocket is completed on day 460.

Test Results:

✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



- Event 2: Transition To **Normal** Major Context → True

- Event 3: Major Context **Normal** Activated

✧ Day 1 – 39

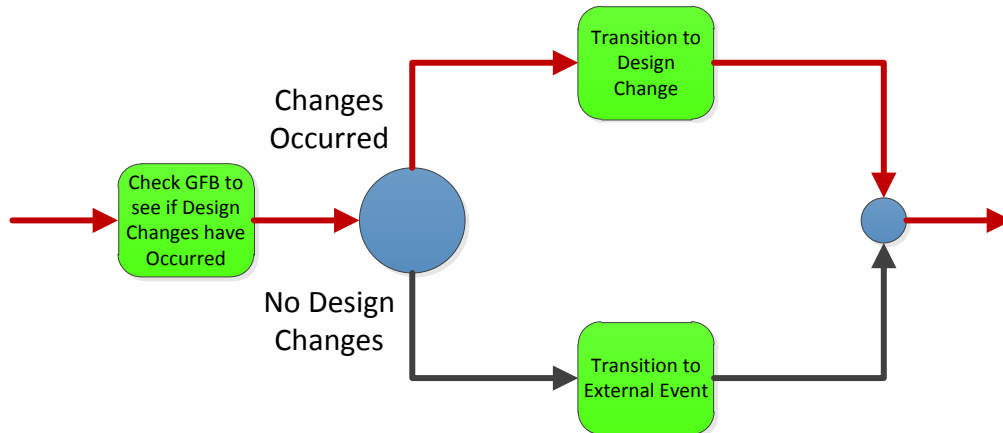
- Major Context = **Normal**
- Normal Action Function Activated

✧ Day 40

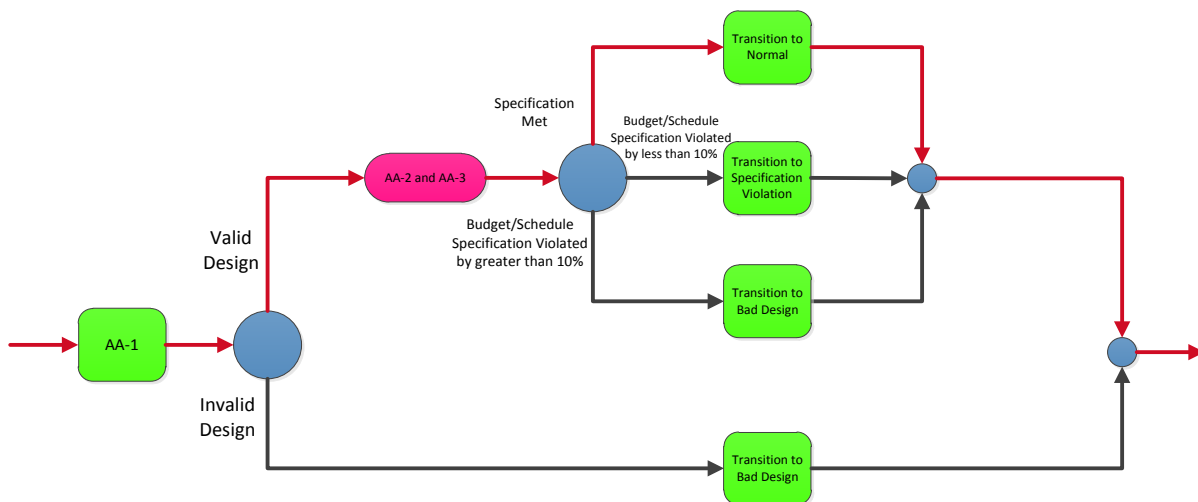
○ Event 1:

- New Design Submitted to PM
- External Event Announced – Cost Increased By 10 MYen
- External Event Announced – Other Delay of 30 days
- External Event Announced – Cost Increased By 5 MYen
- External Event Announced – Cost Increased By 1 MYen
- External Event Announced – Cost Increased By 1.5 MYen
- External Event Announced – Engine Construction Delay of 90 days
- External Event Announced – Control System Delivery Delay of 60 days

- External Event Announced – Other Delay of 15 days
- Transition to **Multiple-Simultaneous-Events** Major Context → True
- Major Context = **Multiple-Simultaneous-Events**
- CxG-4 Activated



- Event 2:
 - Transition to **Design-Change** Major Context → True
 - Major Context = **Design-Change**
 - CxG-2 Activated



- Event 3:
 - Transition to **Normal** Major Context → True
 - Major Context = **Normal**
 - Normal Action Function Activated
- Event 4:
 - Major Context = **Normal**
 - Normal Action Function Activated

✧ Day 41 - 715

- Major Context = **Normal**
- Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

6.2.4 Multiple Simultaneous Events Conclusions

As can be seen from Test 4.1, Test 4.2, Test 4.3, and Test 4.4, the higher contextualization of the system has expanded the capabilities of the PM-agent significantly. Test 4.1, 4.2 and 4.3 show us that the PM-agent can successfully handle multiple simultaneous events involving both cost increases and delays. Unlike the original system which was designed only to handle one event at a time, the higher contextualization enables the PM-agent to look at events in a less sequential manor. The addition of the **Multiple-Simultaneous-Events** Major Context and CxG-4 enables the PM-agent to determine whether it should handle the events as just pure external events or if it needs to take into account change to the design as well. This is important because when design changes occur, the situation needs to be handled differently, i.e. another context of our situation. After the PM-agent evaluates the situation, it can then determine which next context best suits the situation as is seen in the tests above. For Test 4.1 and 4.2, the PM-agent transitioned to **External-Event** Major Context since no design changed occurred. Nevertheless, the PM-agent could have gone to **Design-Change** Major Context which calls agents to look at the design as well as the budget and schedule, this is not efficient since we already know the previous design of the rocket is suitable for our needs. Therefore, in Test 4.1 and 4.2, the PM-agent transitions to **External-Event** Major Context where the PM-agent can now go under the context that external events dealing with the schedule and cost are its only concern. Of course, Test 4.3 is another matter. Here the CxG-4 tells the PM-agent to transition to **Design-Change** Major Context since the design does affect the situation. Now, since there is a necessity to reevaluate the design, the **Design-Change** Major Context has the PM-agent call upon AA-1 to determine the rocket's feasibility which was not a concern in Test 4.1 and 4.2. The breaking down of the situation simplifies the problem for the PM-agent, making for a simpler decision

making process.

Nonetheless, the abilities gained from higher contextualization don't stop there. Now, the system is also able to handle a high number of events occurring at the same time. Test 4.4 demonstrates handling a large amount of events simultaneously by introducing nine simultaneous events. As can be seen from Test 4.4, four events which cause budget increases, four events causing schedule delays, and a design change which affects both schedule and FDD to not cause the slightest bit of problems for the PM-agent. Just as a real PM does, our context-based PM-agent breaks down the situation and chooses a path suitable to the situation. As Test 4.3 shows, handling large amounts of events at the same time is no different than handling a couple of events. Here, the PM-agent acknowledges the multiple simultaneous event situations just as in Test 4.1, 4.2, and 4.3. It can be seen that because of the design of the contextualization of the system, no more work had to be done than was done in simpler situations such as Test 4.1, 4.2 and 4.3. Just as in Test 4.3, the PM-agent recognizes that a design change occurred via CxG-4 and transitions to **Design-Change** Major Context where it will continue just as Test 4.3 did. Again, we emphasize the fact that the new higher contextualized system empowers our PM-agent to handle a multitude of new situations in a simple and efficient manner that was not possible before.

6.2.5 Test Case Involving a Bad Initial Design

Test 4.5: This test case gives the PM-agent a situation that is unlike any of the previous tests. For this test, we will give the PM-agent a scenario where the initial design does not allow the rocket to reach the minimum specified altitude. Of course, this should never happen if we have a competent DA, however, in the real world, this isn't always the case. Therefore, we found this test a suitable the PM-agent will need to be able to handle. In addition, we will use the following values for the test specification: Max Budget will be set at 85 MYen, the FDD will be set at 27 months, and a minimum altitude of 1000 m while carrying a payload of 30 g will be required as was done in Test 4.2.

Initial Design:

Number of stages: Two
Engine selected: Stage 1 – B6-0, Stage 2 – B6-4
Fin material: 1/8" Balsa
Nose Cone material: Balsa
Body Length: 40 cm
Nose Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 17.0 cm
Fin Height: 0.0 cm
Payload Weight: 30 g
Parachute Diameter: 1ft (default)
Control System: ACS-200

Feasibility of Design: The design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	4.0 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	1.0 MYen
Total Cost:	33.0 MYen

Predicted Schedule:

Engine:	4 mo. construction; 4 mo. installation
Control System:	6 months delivery
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. installation
Premium for multiple stages:	1.0 month installation
Total FDD	6 months + 8 months = 14 months

Storyboard: None

Expected Result: On Day 0, during **Initialization** Major Context, the PM-agent will recognize that the design will not be suitable to reach the required altitude. It will immediately transition to **Bad-Design** Major Context where the PM-agent will request a new design from the DA. The DA will submit a design of a 3-stage rocket with the following engines: Stage 1 – D12-0, Stage 2 – D12-0 and Stage 3 – D12-7. Also the body length is lowered to that of the fin length at a size of 24 cm. This will cause the PM-agent to transition to **Design-Change** Major Context where CxG-2 will be activated to make sure the design is within specification. As a note, we want to make sure it is understood that these calculations were done as if this was the first design. This is because since the first design was never accepted, the PM-agent never actually purchased the materials.

Predicted total cost of project:

Engine:	52.0 MYen
Control System:	4.5 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	5.0 MYen
Total Cost:	81.5 MYen

Predicted Schedule:

Engine:	8 mo. construction; 12 mo. installation
Control System:	6 months delivery
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. installation
Premium for multiple stages:	3.0 month installation

Total FDD

8 months + 18 months = 26 months

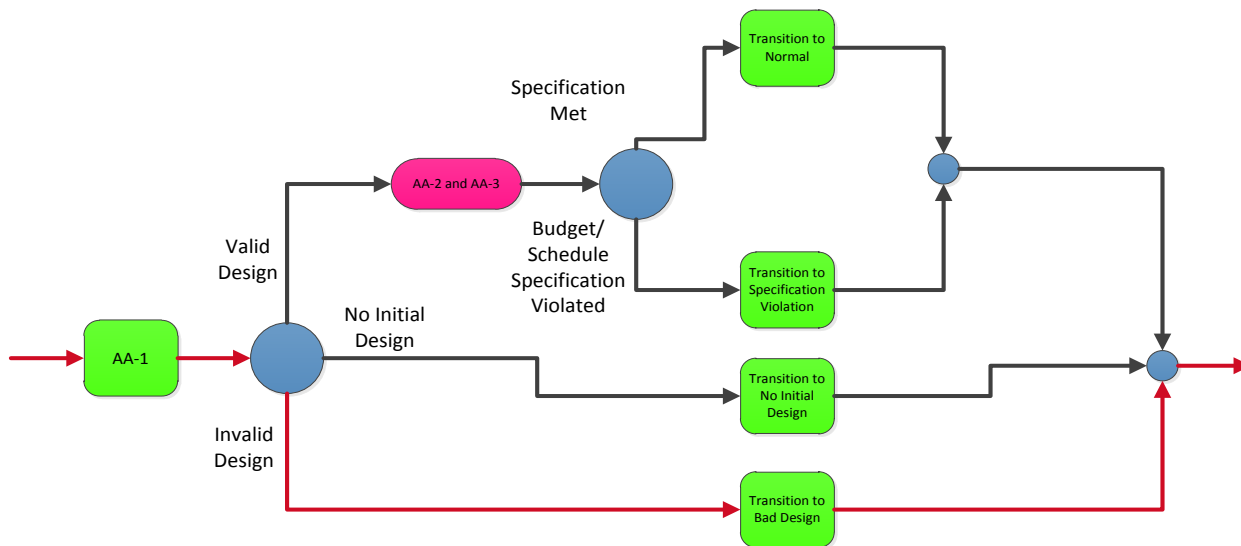
As can be seen, both the total cost and FDD are back still within specification. In addition, the rocket now reaches the required altitude. At this point, the PM-agent will transition to back to normal. The **Normal** Major Context will be activated and the PM-agent will remain here for the rest of the project.

Test Results:

✧ Day 0:

○ Event 1:

- Initial Specification and Design Given to PM
- **Initialize** Major Context Activated
- CxG-1 Activated



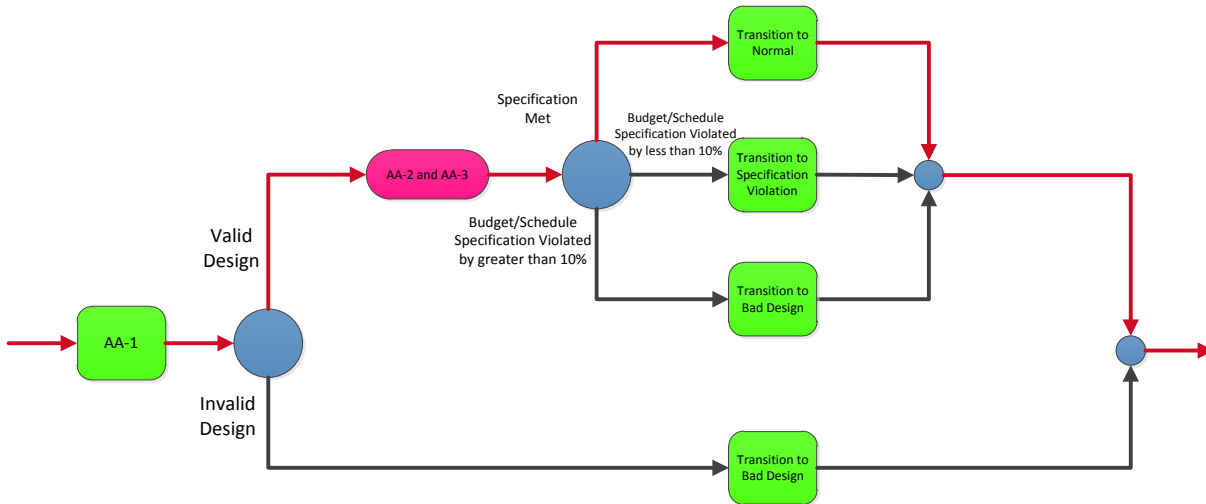
○ Event 2:

- Transition To **Bad-Design** Major Context → True
- Major Context = **Bad-Design**
- Rule – Request New Design Activated
- DA Submits New Design

○ Event 3:

- Transition to **Design-Change** Major Context → True
- Major Context = **Design-Change**

- CxG-2 Activated



- Event 4:
 - Transition to **Normal** Major Context → True
 - Major Context = **Normal**
 - Normal Action Function Activated
- Event 5:
 - Major Context = **Normal**
 - Normal Action Function Activated

✧ Day 1 - 780:

- Major Context = **Normal**
- Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test successful.

5.2.5 Test Case Involving a Bad Initial Design – Conclusions

Here we created a test that expands upon the design of Test 3.1. In this test, we are able to show that we can take the PM-agent to the limits by providing a scenario in which the initial design is invalid. By increasing the system's contextualization with the addition of **Initialization** Major Context and CxG-1, our new system is now capable of handling a situation any initial situation which does not meet specification of design, schedule and/or budget.

6.3 Ability to Easily Change the PM-Agent Because of CxBR

CxBR was designed so that by small contextual changes, the abilities of a system can be changed on the fly. In our system, it was recognized during testing that an infinite loop would occur when transitioning between the **Bad-Design** Major Context and the **Design-Change** Major Context. AA-5 had the potential to continuously provide bad designs which would create a Ping-Pong effect between these major contexts. However, because we were using CxBR, we were easily able to adapt the system to accommodate this circumstance. In the **Bad-Design** Major Context, an action function was added to check that a bad design is never received more than three times including the first bad design. If this occurs, a signal indicating a transition to impasse will be set, causing the PM-agent to transition to the **Impasse** Major Context where it will be left up to upper management to solve the problem. Below are Test 5.1 and Test 5.2. Test 5.1 demonstrates the occurrence of the infinite loop before the changes to the system were made. Test 5.2 demonstrates Test 5.1 again using the modified system. Below outline is shown of the modified **Bad-Design** Major Context:

Major Context Name: *Bad-Design*

Action Functions:

- Requests a new design to present to the PM-agent by activation of the appropriate

action rule.

- When 3 bad designs occur, announce a transition to impasse.

Transition Rules

- When new design received, activate ***Design-Change** Major Context*
- When transition to impasse received, activate ***Impasse** Major Context*

Action Rules:

- Activates AA-5 to request a new design which will be valid and meet specification.

Sub-Contexts

- None

Test 5.1: This test causes the PM-agent to enter an infinite loop it cannot escape. The test will begin with the PM-agent being provided a bad initial design. The PM-agent will request a new design from the DA and be provided with a bad design. The process will continue to repeat as the DA will continuously provide designs which do not meet the specified altitude given the specified payload. We will use the following values for the test specification: Max Budget will be set at 85 MYen, the FDD will be set at 27 months, and a minimum altitude of 900 m while carrying a payload of 30 g will be required.

Initial Design:

Number of stages: Two
Engine selected: Stage 1 – B6-0, Stage 2 – B6-4
Fin material: 1/8" Balsa
Nose Cone material: Balsa
Body Length: 40 cm
Node Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 17.0 cm
Fin Height: 0.0 cm
Payload Weight: 30 g
Parachute Diameter: 1ft (default)
Control System: ACS-200

Feasibility of Design: The design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	4.0 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	1.0 MYen
Total Cost:	33.0 MYen

Predicted Schedule:

Engine:	4 mo. construction; 4 mo. installation
Control System:	6 months delivery
Fin Material:	1.5 mo. construction; 1.5 mo. installation

Nosecone Material:	1.5 mo. construction; 1.5 mo. installation
Premium for multiple stages:	1.0 month installation
Total FDD	6 months + 8 months = 14 months

Storyboard: None

Expected Result: On Day 0, during **Initialization** Major Context, the PM-agent will recognize that the design will not be suitable to reach the required altitude. It will immediately transition to **Bad-Design** Major Context where the PM-agent will request a new design from AA-5. AA-5 will submit a design of a 2-stage rocket with the following engines: Stage 1 – D12-0, Stage 2 – D12-7. This will cause the PM-agent to transition to **Design-Change** Major Context where CxG-2 will be activated to make sure the design is within specification. Again, the PM-agent will recognize the rocket is not able to reach the required altitude. The PM-agent will transition to **Bad-Design** Major Context where the PM-agent will request a new design from AA-5. AA-5 will submit a design of a 2-stage rocket with the following engines: Stage 1 – C6-0, Stage 2 – C6-7. This will cause the PM-agent to transition to **Design-Change** Major Context where CxG-2 will be activated to make sure the design is within specification. Again, the PM-agent will recognize the rocket is not able to reach the required altitude. The PM-agent will transition to **Bad-Design** Major Context where the PM-agent will request a new design from AA-5. AA-5 will submit a design of a 2-stage rocket with the following engines: Stage 1 – C6-0, Stage 2 – C6-5. This will cause the PM-agent to transition to **Design-Change** Major Context where CxG-2 will be activated to make sure the design is within specification. Again, the PM-agent will recognize the rocket is not able to reach the required altitude. The PM-agent will transition to **Bad-Design** Major Context where the PM-agent will request a new design from AA-5. At this point, we will have AA-5 submit a design which meets the altitude specification so that we may leave the infinite loop. AA-5 will submit a design of a 3-stage rocket with the following engines:

Stage 1 – D12-0, Stage 2 – D12-0 and Stage 3 – D12-7. Also the body length is lowered to that of the fin length at a size of 24 cm so that the rocket maintains stability with these powerful engines. This will cause the PM-agent to transition to **Design-Change** Major Context where CxG-2 will be activated to make sure the design is within specification.

Predicted total cost of project:

Engine:	52.0 MYen
Control System:	4.5 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	5.0 MYen
Total Cost:	81.5 MYen

Predicted Schedule:

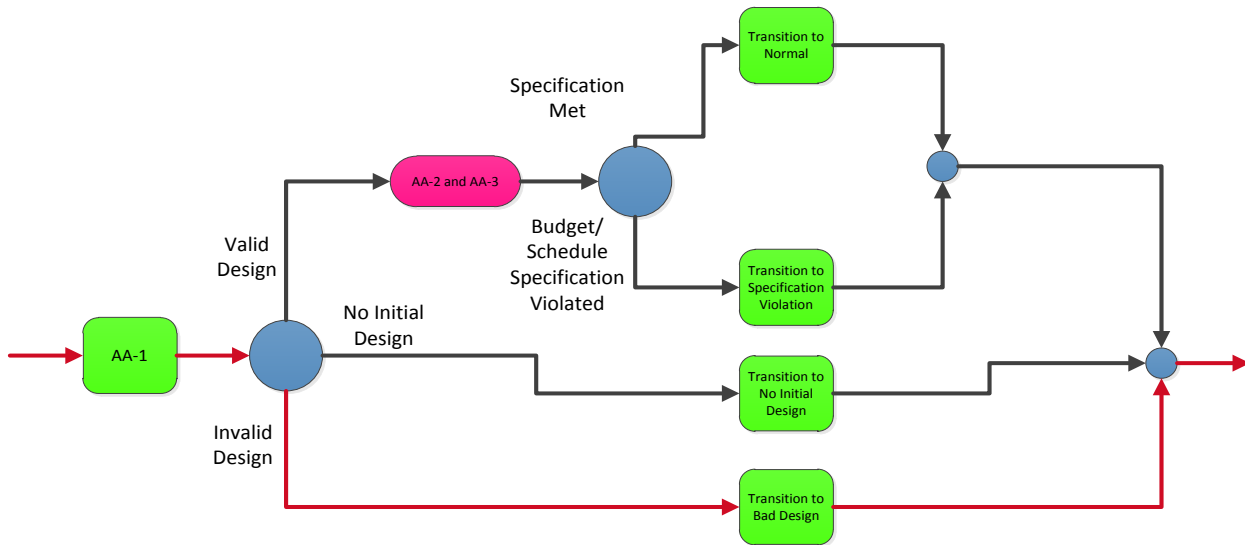
Engine:	8 mo. construction; 12 mo. installation
Control System:	6 months delivery
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. installation
Premium for multiple stages:	3.0 month installation
Total FDD	8 months + 18 months = 26 months

This time, the PM-agent will recognize the design meets specification and the rocket is able to meet the specified altitude while carrying the specified payload. Nonetheless, the PM-agent would have stayed in the infinite loop had we not submitted a valid design. The PM-agent will then transition to the Normal Major where it will stay for the duration of the mission.

Test Results:

- ✧ Day 0:
 - Event 1:
 - Initial Specification and Design Given to PM
 - **Initialize** Major Context Activated

- CxG-1 Activated

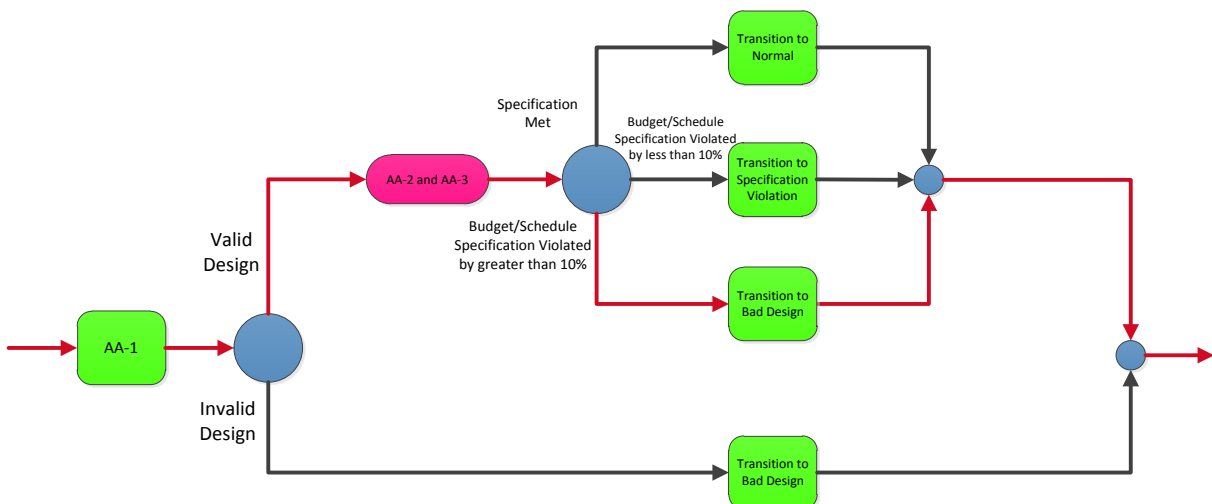


- Event 2:

- Transition To **Bad-Design** Major Context → True
- Major Context = **Bad-Design**
- Rule – Request New Design Activated
- DA Submits New Design

- Event 3:

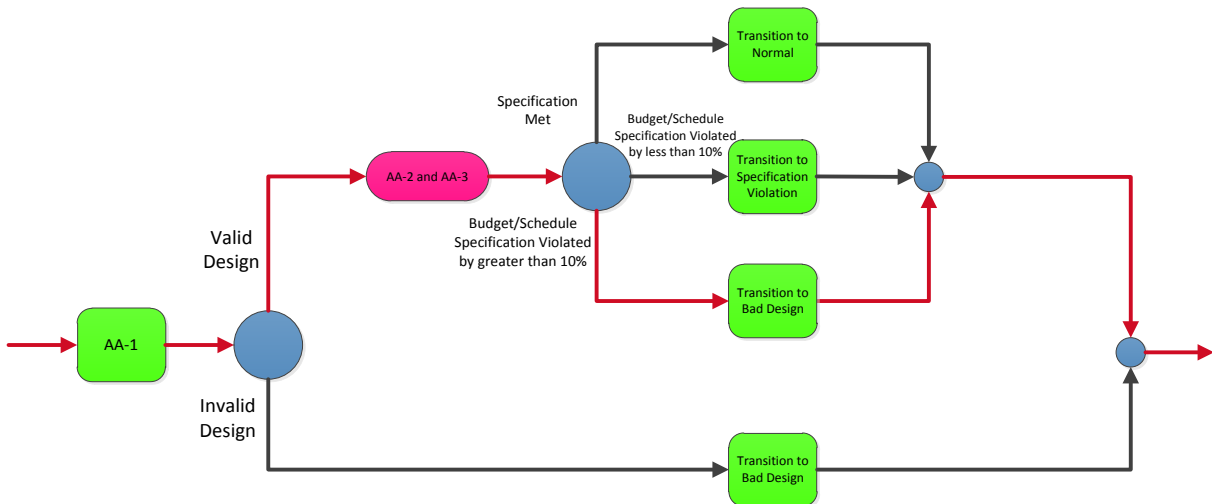
- Transition to **Design-Change** Major Context → True
- Major Context = **Design-Change**
- CxG-2 Activated



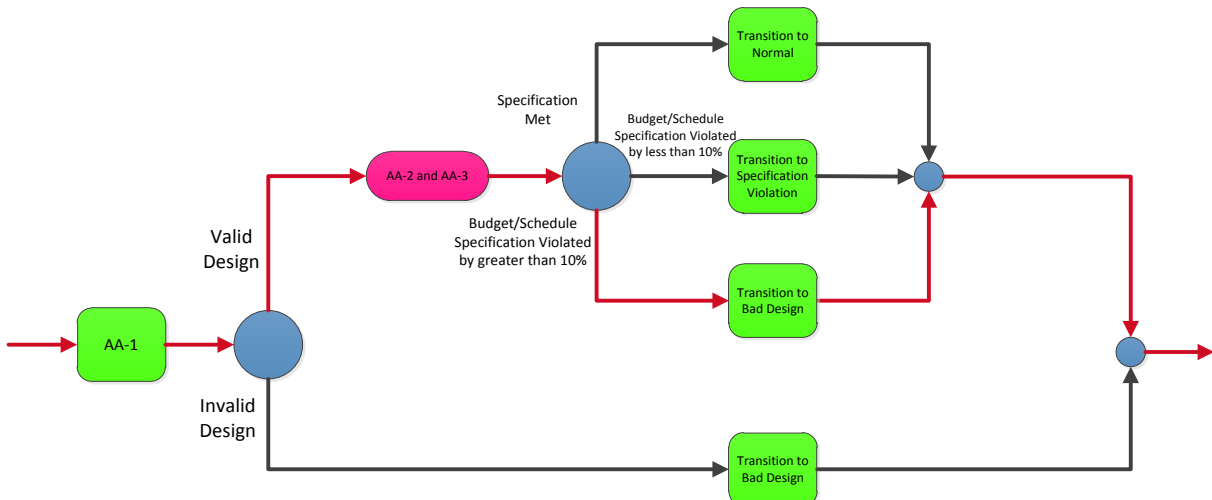
- Event 4:

- Transition To **Bad-Design** Major Context → True

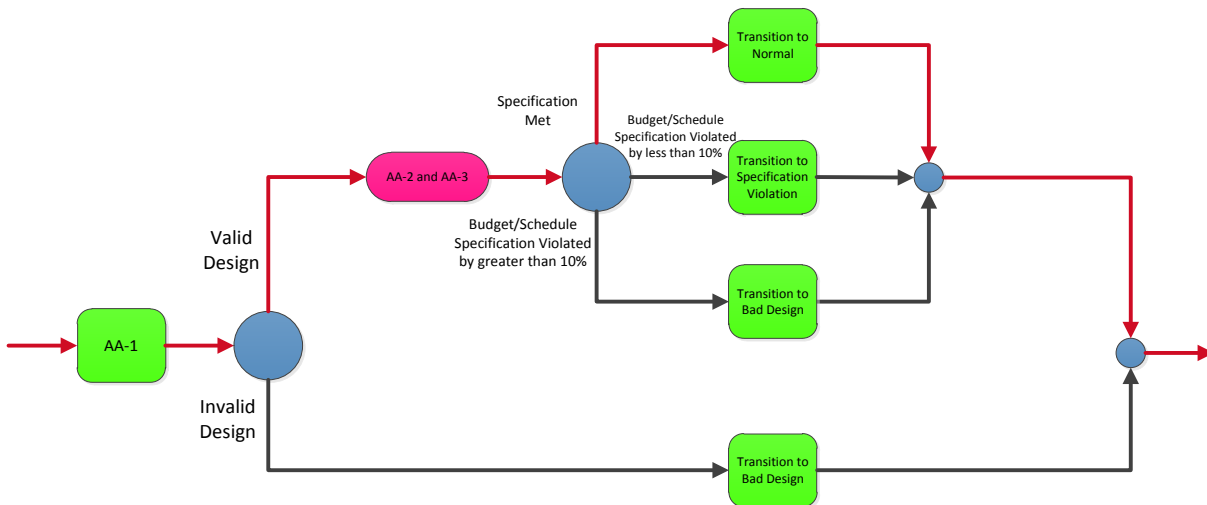
- Major Context = **Bad-Design**
- Rule – Request New Design Activated
- DA Submits New Design
- Event 5:
 - Transition to **Design-Change** Major Context → True
 - Major Context = **Design-Change**
 - CxG-2 Activated



- Event 6:
 - Transition To **Bad-Design** Major Context → True
 - Major Context = **Bad-Design**
 - Rule – Request New Design Activated
 - DA Submits New Design
- Event 7:
 - Transition to **Design-Change** Major Context → True
 - Major Context = **Design-Change**
 - CxG-2 Activated



- Event 8:
 - Transition To **Bad- Design** Major Context → True
 - Major Context = **Bad-Design**
 - Rule – Request New Design Activated
 - DA Submits New Design
- Event 9:
 - Transition to **Design-Change** Major Context → True
 - Major Context = **Design-Change**
 - CxG-2 Activated



- Event 10:
 - Transition to **Normal** Major Context → True
 - Major Context = **Normal**
 - Normal Action Function Activated
 - Event 11:
 - Major Context = **Normal**
 - Normal Action Function Activated
- ✧ Day 1 - 780:
- Major Context = **Normal**
 - Normal Action Function Activated

Conclusion(s): Results in agreement with the expected results. Test became stuck in infinite loop and PM-agent had to be given valid design to leave it.

Test 5.2: This test reruns Test 5.1 with the modified **Bad-Design** Major Context. We will use the following values for the test specification: Max Budget will be set at 85 MYen, the FDD will be set at 27 months, and a minimum altitude of 900 m while carrying a payload of 30 g will be required.

Initial Design:

Number of stages: Two
Engine selected: Stage 1 – B6-0, Stage 2 – B6-4
Fin material: 1/8" Balsa
Nose Cone material: Balsa
Body Length: 40 cm
Nose Cone Length: 8.0 cm
Fin Width: 4.0 cm
Body Diameter: 2.5 cm
Fin Length: 17.0 cm
Fin Height: 0.0 cm
Payload Weight: 30 g
Parachute Diameter: 1ft (default)
Control System: ACS-200

Feasibility of Design: The design characteristics are such that they allow the rocket to fulfill its required task.

Predicted total cost of project:

Engine:	9.0 MYen
Control System:	4.0 MYen
Fin Material:	10.0 MYen
Nosecone Material:	10.0 MYen
Premium for multiple stages:	1.0 MYen
Total Cost:	33.0 MYen

Predicted Schedule:

Engine:	4 mo. construction; 4 mo. installation
Control System:	6 months delivery
Fin Material:	1.5 mo. construction; 1.5 mo. installation
Nosecone Material:	1.5 mo. construction; 1.5 mo. installation
Premium for multiple stages:	1.0 month installation
Total FDD	6 months + 8 months = 14 months

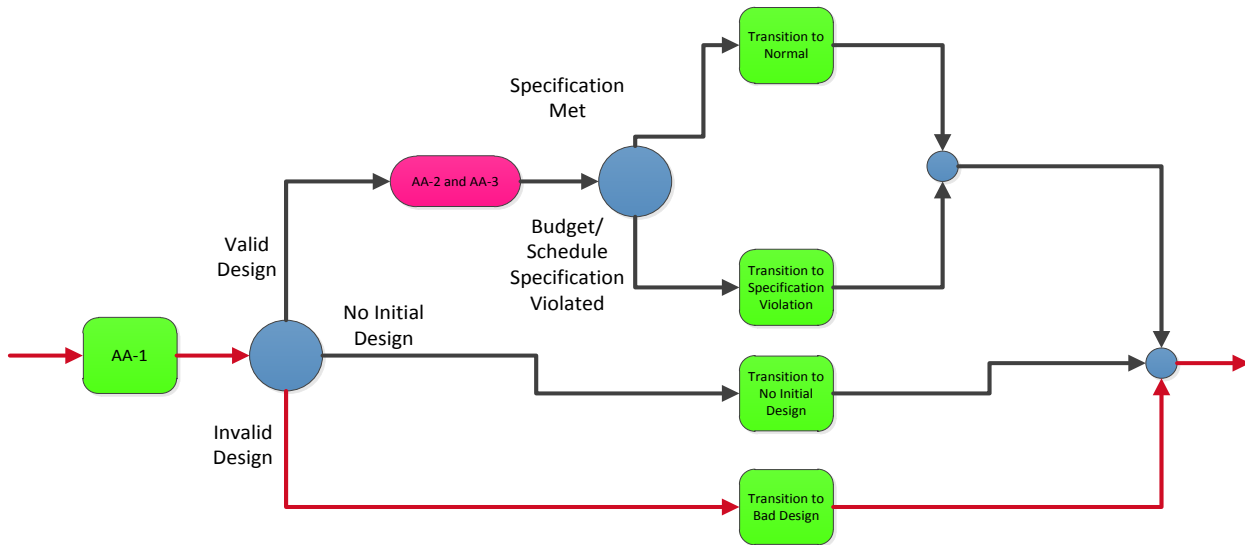
Storyboard: None

Expected Result: On Day 0, during **Initialization** Major Context, the PM-agent will recognize that the design will not be suitable to reach the required altitude. It will immediately transition to **Bad-Design** Major Context where the PM-agent will request a new design from AA-5. AA-5 will submit a design of a 2-stage rocket with the following engines: Stage 1 – D12-0, Stage 2 – D12-7. This will cause the PM-agent to transition to **Design-Change** Major Context where CxG-2 will be activated to make sure the design is within specification. Again, the PM-agent will recognize the rocket is not able to reach the required altitude. The PM-agent will transition to **Bad-Design** Major Context where the PM-agent will request a new design from AA-5. AA-5 will submit a design of a 2-stage rocket with the following engines: Stage 1 – C6-0, Stage 2 – C6-7. This will cause the PM-agent to transition to **Design-Change** Major Context where CxG-2 will be activated to make sure the design is within specification. Again, the PM-agent will recognize the rocket is not able to reach the required altitude. The PM-agent will transition to **Bad-Design** Major Context. However, since this is the third bad design, the PM-agent will transition to **Impasse** Major Context. CxG-5 will then be activated and a request will be sent to upper management to correct the issues with the system. For this test, we will have upper management will just cancel the test. The PM-agent will then cancel the mission.

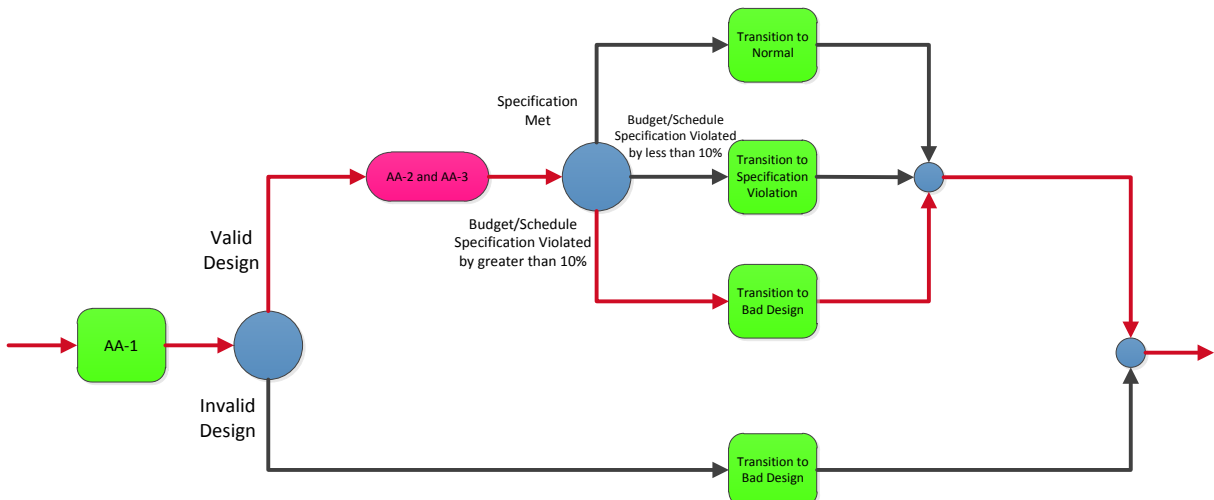
Test Results:

- ✧ Day 0:
 - Event 1:
 - Initial Specification and Design Given to PM
 - **Initialize** Major Context Activated

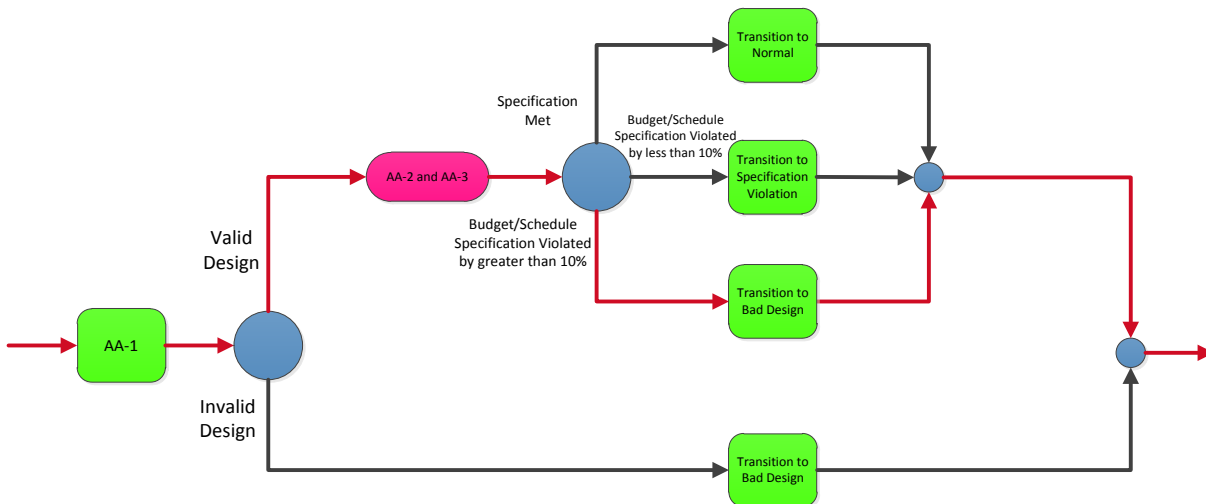
- CxG-1 Activated



- Event 2:
 - Transition To **Bad-Design** Major Context → True
 - Major Context = **Bad-Design**
 - Activated Bad Design Action Function → True
 - Bad Design Action Function Activated
- Event 3:
 - Rule – Request New Design Activated
 - DA Submits New Design
- Event 4:
 - Transition to **Design-Change** Major Context → True
 - Major Context = **Design-Change**
 - CxG-2 Activated

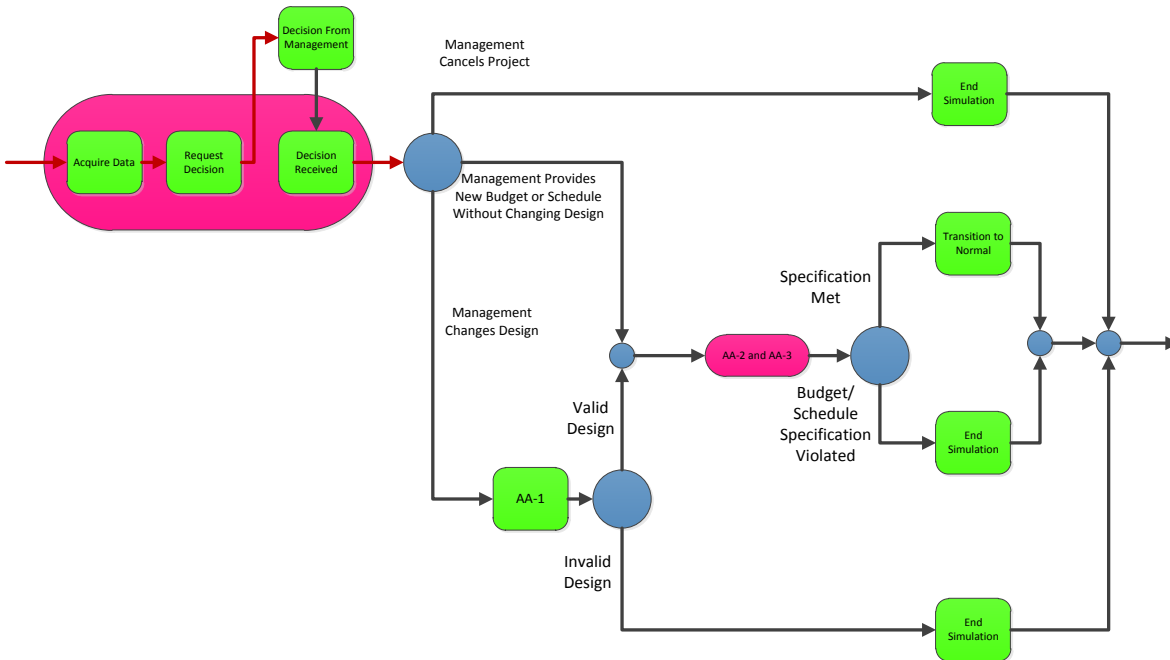


- Event 5:
 - Transition To **Bad-Design** Major Context → True
 - Major Context = **Bad-Design**
 - Activated Bad Design Action Function → True
 - Bad Design Action Function Activated
- Event 6:
 - Rule – Request New Design Activated
 - DA Submits New Design
- Event 7:
 - Transition to **Design-Change** Major Context → True
 - Major Context = **Design-Change**
 - CxG-2 Activated

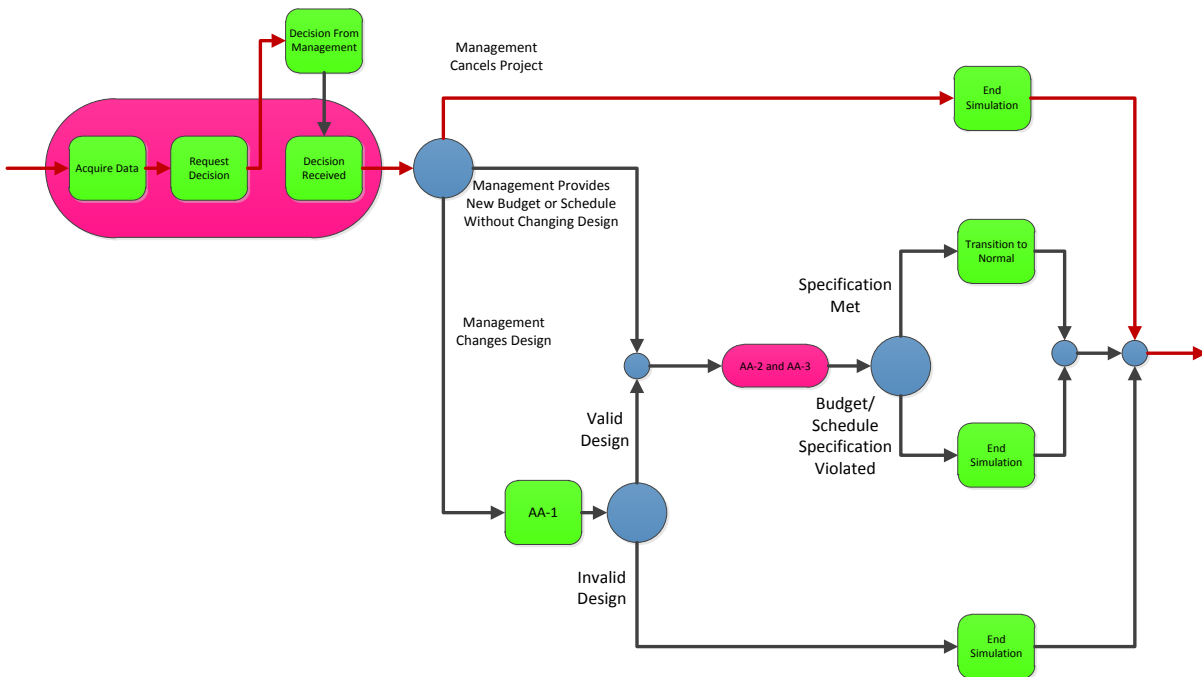


- Event 8:
 - Transition To **Bad-Design** Major Context → True
 - Major Context = **Bad-Design**
 - Activated Bad Design Action Function → True
 - Bad Design Action Function Activated
- Event 9:
 - Major Context = **Bad-Design**
 - Transition To **Impasse** Major Context → True
 - Major Context = **Impasse**
 - Rule – Activate CxG-5 → True

- CxG-5 Activated



- CxG-5 Activated to Continue



- Event 10:

- PM-agent reports “The Mission Has Been Canceled”

Conclusion(s): Results in agreement with the expected results. Test did not cause PM-agent to get stuck in infinite loop.

6.3.1 Ease of Change Conclusions

Test 5.1 proved that a problem existed with our design. In Test 5.1, the PM-agent entered an infinite loop from which it could not escape until a proper design that met specification was given. However, there might not always be a proper design and the best option might just be to cancel the project. Because of the ease of modification of the PM-agent provided by the CxBR infrastructure, with 25 lines of new code for the new Design-Change Action Function and 8 lines of XML script, for the new transition rule and new action function, we were able to correct this problem. In all, the correction took less than one hour to implement with testing.

With the new action function and transition rule implemented, Test 5.2 was run. As can be seen above, the test ran seamlessly. After the third time that a bad design was reported, the PM-agent automatically transitioned to **Impasse** Major Context thereby leaving it up to upper management to make the call on what to do about the problem. This is a much better solution than calling upon AA-5 to provide a new design which it proved itself clearly not able to do.

6.4 Summary and Conclusion of Test Results

Table 9 below summarizes the tests done in this Chapter. In the table, a key is used in order to condense the results to fit in the table. In the table, the following keys are used for the major contexts: “In” is used to represent the **Initialize** Major Context, “No” is used to represent the **No-Initial-Design** Major Context, “BD” is used to represent the **Bad-Design** Major Context, “SV” is used to represent the **Specification-Violation** Major Context, “N” is used to represent the Normal Major Context, “DC” is used to represent the **Design-Change** Major Context, “EE” is used to represent the **External-Event** Major Context , “MSE” is used to represent the **Multiple-Simultaneous-Events** Major Context, and “I” is used to represent the **Impasse** Major

Context. The keys for the sub-contexts are as follows, C1 represents CxG-1, C2 represents CxG-2, C3 represents CxG-3, C4 represents CxG-4, and C5 represents CxG-5. When a design is rejected, an “r” will be used and when a design is accepted, an “a” will be used. Corrective Action Operator 1 and 2 will be represented by CA1 and CA2 respectively. Lastly, Can will be used to show a mission is cancelled.

Test Number	Expected Result	Actual Result	Success?
1.1	In-C1-a-N	In-C1-a-N	Yes
1.2	In-C1-a-N-EE-C3-N	In-C1-a-N-EE-C3-N	Yes
1.3	In-C1-a-N-EE-C3-SV-CA2-N	In-C1-a-N-EE-C3-SV-CA2-N	Yes
1.4	In-C1-a-N-EE-C3-N	In-C1-a-N-EE-C3-N	Yes
1.5	In-C1-a-N-EE-C3-SV-CA1-N	In-C1-a-N-EE-C3-SV-CA1-N	Yes
1.6	In-C1-a-N-EE-C3-SV-CA1-N-EE-C3-SV-VA-CA2-N	In-C1-a-N-EE-C3-SV-CA1-N-EE-C3-SV-VA-CA2-N	Yes
2.1	In-C1-a-N-DC-C2-a-N	In-C1-a-N-DC-C2-a-N	Yes
2.2	In-C1-a-N-DC-C2-r-BD-DC-C2-a-N	In-C1-a-N-DC-C2-r-BD-DC-C2-a-N	Yes
2.3	In-C1-a-N-DC-C2-a-SV-CA2-N	In-C1-a-N-DC-C2-a-SV-CA2-N	Yes
2.4	In-C1-a-N-DC-C2-a-BD-DC-C2-a-SV-CA2-N	In-C1-a-N-DC-C2-a-BD-DC-C2-a-SV-CA2-N	Yes
2.5	In-C1-a-N-DC-C2-r-BD-DC-C2-a-N	In-C1-a-N-DC-C2-r-BD-DC-C2-a-N	Yes
3.1	In-C1-a-SV-I-C5-N	In-C1-a-SV-I-C5-N	Yes
3.2	In-C1-No-I-C5-Can	In-C1-No-I-C5-Can	Yes
3.3	In-C1-No-In-C1-a-SV-CA1-N	In-C1-No-In-C1-a-SV-CA1-N	Yes
4.1	In-C1-a-N-EE-C3-SV-CA1-N-EE-C3-SV-CA2-N-MSE-C4-EE-C3-SV-CA2-N	In-C1-a-N-EE-C3-SV-CA1-N-EE-C3-SV-CA2-N-MSE-C4-EE-C3-SV-CA2-N	Yes
4.2	In-C1-a-N-EE-C3-SV-CA1-N-EE-C3-SV-CA2-N-MSE-C4-EE-C3-SV-I-C5-N	In-C1-a-N-EE-C3-SV-CA1-N-EE-C3-SV-CA2-N-MSE-C4-EE-C3-SV-I-C5-N	Yes
4.3	In-C1-a-N-MSE-C4-DC-C2-a-N	In-C1-a-N-MSE-C4-DC-C2-a-N	Yes
4.4	In-C1-a-N-MSE-C4-DC-C2-a-N	In-C1-a-N-MSE-C4-DC-C2-a-N	Yes
4.5	In-C1-r-BD-DC-C2-a-N	In-C1-r-BD-DC-C2-a-N	Yes
5.1	In-C1-r-BD-DC-C2-r-BD-DC-C2-r-BD-DC-C2-r-BD-DC-C2-a-N	In-C1-r-BD-DC-C2-r-BD-DC-C2-r-BD-DC-C2-r-BD-DC-C2-a-N	Yes
5.2	In-C1-r-BD-DC-C2-r-BD-DC-C2-r-I-C5-Can	In-C1-r-BD-DC-C2-r-BD-DC-C2-r-I-C5-Can	Yes

As can be seen from the test summary provided by Table 9, every test was able to be completed successfully. Because of our more highly contextualization system, our system was able to handle all the situations introduced by the old system as well as a new series of situations presented by Test Set 4. In addition, we were able to show that because of the introduction of the **Bad-Design** Major Context and CxG-2, the PM-agent didn't rely as heavily on AA-4, as shown by Test 2.4. Lastly, we were able to show how quick and easy the CxBR architecture allows a system is to modify should new situations arise that are currently not handled.

CHAPTER 7 – SUMMARY, CONCLUSIONS AND FUTURE RESEARCH

The research succeeded in improving upon Gonzalez et al.'s PM-agent design. We were able to increase the situational awareness of the PM-agent. This is because the more highly contextualized system, the PM-agent is now able to handle more than just design changes and external events. The PM-agent now has the ability to handle multiple simultaneous events, projects that start off with either a bad design or a specification that is not met, and projects that do not have an initial design.

In addition, we have successfully removed some of the burden from AA-4. With the use of CxG-2 by the **Design-Change** Major Context, the PM-agent removes some of the burden from AA-4 by checking to see if the specification is exceeded by 10% in the schedule or budget before determining the next context to transition to. In turn, if the specification is violated as such, the PM-agent will transition to the new **Bad-Design** Major Context where the design will be rejected outright rather than transitioning to the **Specification-Violation** Major Context. The latter would be an attempt to have AA-4 try to solve problems it is most likely will not be able to handle because of the limitations of the corrective action operators. Lastly, our research showed the ease and speed with which one could change the functionality of the PM-agent because of the used of CxBR.

However, we did not succeed in making a true constraint-based AA-4 as we would have like to do. For the current responsibilities of AA-4, true constraint-based architecture was not necessary; in the future however, more constraints will likely be necessary as systems become more complex. This will require that AA-4 be a more robust problem solving agent. Therefore, for future research, a true constraint-based AA-4 will become necessary.

In conclusion, after the completion of this project, research on the benefits of the use of higher contextualization for the Rocket Development PM-agent have been shown. Although there is much more that could be done to improve the system, most of these improvements would not benefit from a context-based architecture. Therefore, for future research, we recommend investigating a more complex application that more rigorously stresses the CxBR architecture. This will require additional contextualization, but as we have shown, contextualization can help address the problem of complexity. Additionally, in more complex systems, the AA-4 problem solving agent will require significant enhancements, including but not limited to, a true constraint-based problem solver.

REFERENCES

- [Gonzalez et al., 2008a] Gonzalez, A. J., Nguyen, J., Tsuruta, S., Sakurai, Y., Takada, K., & Uchida, K. (2008). Using contexts to supervise a collaborative process. IEEE International Conference on Systems, Man and Cybernetics, 2008. SMC 2008. 2706-2711.
- [Gonzalez et al., 2008b] Gonzalez, A. J., Stensrud, B. S., & Barrett, G. (2008). Formalizing context-based reasoning: A modeling paradigm for representing tactical human behavior. *International Journal of Intelligent Systems*, 23(7), 822-847.
- [Gonzalez and Ahlers, 1998] Gonzalez, A. J., & Ahlers, R. (1998). Context-based representation of intelligent behavior in training simulations. *Transactions of the Society for Computer Simulation International*, 15(4), 153-166.
- [Borges et al., 2004] Borges, M. R. S., Brézillon, P., Pino, J. A. & Pomerol, J. C. (2004). Bringing context to CSCW. Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design, 2004. 2 161-166 Vol.2.
- [Hu and Liu, 2007] Hu Z., & Liu J. (2007). The research of ASW helicopter ACGF construction based on CXBR. Computational Intelligence and Security Workshops, 2007. CISW 2007. 132-135.
- [Kulkarni and Tripathi, 2010] Kulkarni, D., & Tripathi, A. (2010). A framework for programming robust context-aware applications. *IEEE Transactions on Software Engineering*, 36(2), 184-197.

- [Hur et al., 2008] Hur, H., Park, H., Kim, J., Chang, H., Lee, M., Cho, S., et al. (2008). Context model of user aspect in context-awareness. 10th International Conference on Advanced Communication Technology, 2008. ICACT 2008. 3 1647-1651.
- [Mostéfaoui and Brézillon, 2004] Mostéfaoui, G. K., & Brézillon, P. (2004). Modeling context-based security policies with contextual graphs. 2004. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 28-32.
- [Nguyen et al., 2008] Nguyen, T. V., Lim, W., Nguyen, H. A., & Deokjai Choi. (2008). Context awareness framework based on contextual graph. Wireless and Optical Communications Networks, 2008. WOCN '08. 5th IFIP International Conference on, 1-5.
- [Brézillon et al., 2002] Brézillon, P., Pasquier, L., & Pomerol, J. (2002). Reasoning with contextual graphs. *European Journal of Operational Research*, 136(2), 290-298.
- [Gonzalez, 2007] Gonzalez, A.J. (2007). Using Contexts to Control a Collaborative Process. Final Report, Tokyo Denki University, December 2007. Retrieved from <http://isl.ucf.edu>
- [Wikipedia, 2010] Computer supported cooperative work. (n.d.). In Wikipedia. Retrieved November 18, 2010, from <http://en.wikipedia.org/wiki/CSCW>
- [NASA, 2003] National Aeronautics and Space Administration. (2003). RocketModeler (Version 1.2) [Software]. Available from <http://microgravity.grc.nasa.gov/education/rocket/>