A Comparative Analysis Between Context-based Reasoning (cxbr) And Contextual Graphs (cxgs).

Peterson Marthen Lorins
University of Central Florida

A COMPARATIVE ANALYSIS
BETWEEN CONTEXT BASED REASONING (CXBR)
AND CONTEXTUAL GRAPHS (CXGS)

by

PETERSON M. LORINS
B.E. The City College of C.U.NY, 2000

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2005

ABSTRACT

Context-based Reasoning (CxBR) and Contextual Graphs (CxGs) involve the modeling of human behavior in autonomous and decision-support situations in which optimal human decision-making is of utmost importance. Both formalisms use the notion of contexts to allow the implementation of intelligent agents equipped with a context-sensitive knowledge base. However, CxBR uses a set of discrete contexts, implying that models created using CxBR operate within one context at a given time interval. CxGs use a continuous context-based representation for a given problem-solving scenario for decision-support processes. Both formalisms use contexts dynamically by continuously changing between necessary contexts as needed in appropriate instances. This thesis identifies a synergy between these two formalisms by looking into their similarities and differences. It became clear during the research that each paradigm was designed with a very specific family of problems in mind. Thus, CXBR best implements models of autonomous agents in environment, while CxGs is best implemented in a decision-support setting that requires the development of decision-making procedures. Cross applications were implemented on each and the results are discussed.

I would like to dedicate this thesis to my parents who instilled in me the value of integrity, and character, my best friend Arlene who never ceases to believe in me, my late brother Reginald Lorins who continues to live through everything that I do, my two amazing children Brandon and Trisha, and last but not least my advisors Dr. Avelino Gonzalez of the University of Central Florida and Dr. Patrick Brézillon, of LIP6, University Paris 6, without whom this wouldn't have been possible.

ACKNOWLEDGMENTS

I'd like to thank Dr. Avelino Gonzalez and Dr. Patrick Brézillon for being an inspiration to me. Dr. G., thank you for always pushing me towards excellence and reminding me of the bottom line. I am very humbled to have had the chance to work with men like you; it is something that I’ll carry with me for a lifetime.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AIP's	Autonomous Intelligent Platforms
CXBR	Context-Based Reasoning
CXGs	Contextual Graphs
CGFs	Computer Generated Forces
CHRIS	Common HBR Interchange System
CITKA	Context-based Intelligent Tactical Knowledge Acquisition
DMSO	Defense Modeling and Simulation Office
DOD	Department of Defense
GP	Genetic Programming
HBR	Human Behavior Representation
HVA	High Value Asset
KA PACKAGE	Knowledge Acquisition Package
KE	Knowledge Engineer
MOE's	Measures of Effectiveness (HBR Tactical Effectiveness)
MOP's	Measures of Performance (HBR Technical Performance)
MOV's	Measures of value (Overall Utility of the HBR Model)
NM	Nautical Miles
SAF's	Semi-Automated Forces
SME	Subject Matter Expert

CHAPTER ONE: INTRODUCTION

This thesis presents a comparative analysis of two major context-based formalisms, namely Context-based Reasoning, and Contextual Graphs. The goal was to eventually identify a potentially viable framework to synergistically compare them via their common characteristics while also considering their differences.

This type of comparison was facilitated through the use of similarity metrics, such as their similarity involving ‘context representation’, ‘context movement and/or contextual change’, “Environment: parallel between CxBR’s autonomous knowledge, and CxG’s procedures and practices, ‘time representation’, and ‘knowledge acquisition and learning’. The end result of this comparison could be represented in a tool named the ‘CxBR-CxG Synergy Exploration Tool’. This chapter provides a clear understanding of the two formalisms, in order to effectively compare them. CxBR will be covered first, and will be followed by a description of CxGs based on the literature.

1.1 Context-Based Reasoning (CxBR)

Context-based Reasoning is a human behavior representation paradigm that uses context as the basis of the representation. CxBR models are used to control autonomous agents in the performance of a autonomous mission. In CxBR, there are three kinds of contexts and they are organized hierarchically to represent contexts and they are defined as: (1) Major Context, (2) Major Context and (3) Sub-Context (Figure 1a). The Mission-Context defines the constraints and the Major Contexts that are to be used in the execution of the described mission (Saeki, and Gonzalez, 2000). In order to construct a CxBR model, one must first define a mission outlining an autonomous agent's goals and motives.

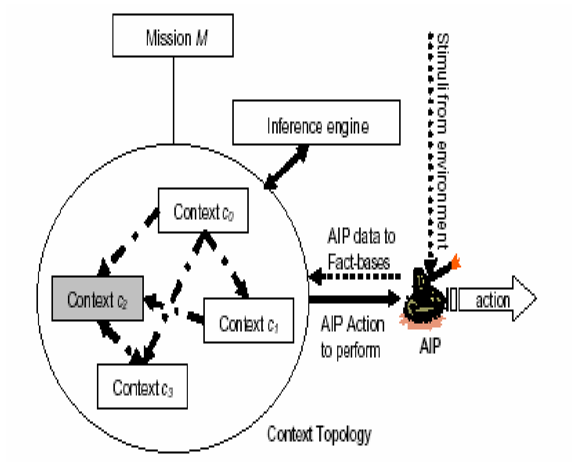


Figure 1: CxBR Model Block diagram (Stensrud, and Gonzalez, 2003)

The autonomous agent's knowledge of the autonomous situation in question is defined implicitly within the 'contexts' of a 'mission', where a context, or Major Context, is a set of environmental and physical conditions that suggest a specific behavior (Gonzalez and Ahlers, 1998). Each mission encompasses a set of mission requirements that have to be satisfied. In order to satisfy such requirements, a autonomous agent often has to operate in several of the included major contexts at different times during a mission execution. To detect the conditions under which a context switch or transition is required 'sentinel rules' that hold such triggering conditions are used appropriately. A mission uses a set of contexts and context-transition pairs to define the high-level behavior of the autonomous agent. When implemented within a CxBR model, a Context includes a set of Sentinel Rules and Context Logic. The context logic represents the conditions, rules, and functions that constitute the Autonomous Agent's behavior within a context (Stensrud, and Gonzalez, 2003).

On the other hand, Sub-contexts represent actions/functions that are less critical to the mission objectives. Sub-contexts are autonomous procedures that are not critical themselves in reaching the mission objectives. They are typically of temporarily short duration. Sub-contexts are at this time mutually exclusive with one another, but can be compatible, and thus be allowed to coexist within major contexts. (Saeki, and Gonzalez, 2000). Transitional Sentinel Rules look for the conditions in the environment, which call for transition to another major context. They are as important in representing the autonomous agent's behavior as the context logic itself. For instance, if the mission provides a context-transition pair for Major Context C1 to C3, a sentinel rule will be present within C1 monitoring for the conditions warranting a transition from C1 to C3. As expected, this sentinel rule will have an antecedent and a consequent part. If any transition-warranting conditions exist, the sentinel rule will fire and a transition will be initiated, activating the new context and de-activating the old one.

The use of a set of discrete contexts in CxBR, so that models created using it can operate within one context at a given time interval, is related to that of Giunchiglia [1993] where the following are considered: (1) the notion of a set of discrete contexts, (2) the relationships between them (via bridging rules for entering and leaving the contexts), and (3) a context that makes reasoning local. A comparison can be drawn with the realization that a "bridging rule" therein is quite similar to a sentinel rule (existing within a particular context) in CxBR. The latter simply warrants that a transition will occur from that specific context to another if the CxBR mission provides a context-transition pair that includes such a context.

CxBR is based on the idea that the execution of a autonomous mission is based on the expectation that the autonomous agent in question will encounter sequential situations. In order to successfully navigate a particular autonomous situation, certain skills and actions are required. One has to keep in mind that quite often situations evolve abruptly from one to the next. Thus, the successful completion of the mission is contingent upon the autonomous agent's ability to successfully navigate each of the autonomous situations, and recognize when a change in autonomous situation has occurred. Accordingly, Gonzalez et al (2002), emphasize three basic principles that are required by CxBR, namely, " (1) A autonomous agent calls for set of actions and procedures that properly address the current situation; (2) As a mission evolves, a transition to another set of actions and procedures may be required to address the new situation; (3) Things that are likely to happen while under the current situation are limited by the current situation itself."

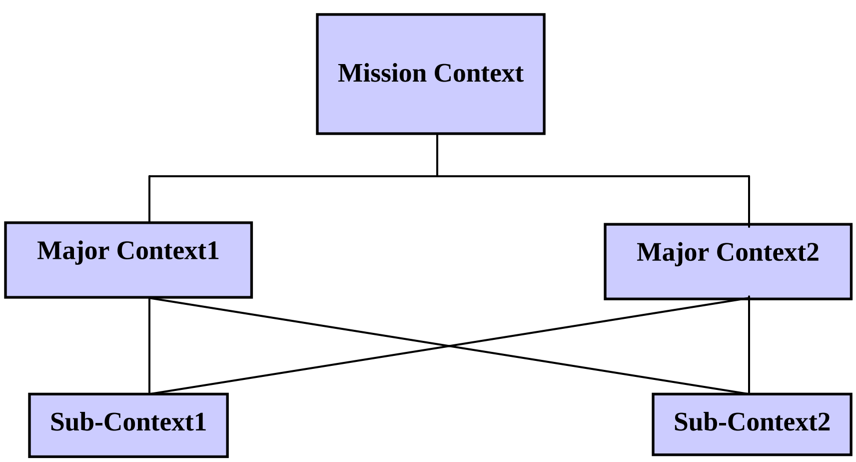


Figure 1a: Context Hierarchy (Gonzalez et al., 2002)

1.1.1 CxBR Implementation

The literature indicates that in order to create any model in CxBR, one must clearly identify the following (Gonzalez et al):

- " The Mission Context and a value set for its applicable attributes, including name of the mission, description, weather, lighting, conditions, location, constraints, and the mission objective
- The Subject Matter Expert (SME) must be used to identify the main contexts and sub-contexts, whether the context is part of the original plan or in reaction to unplanned, yet potentially expected situations.
- The procedures required for controlling a simulated entity while under each major context or sub-context.
- The specification and incorporation of all procedures into their appropriate contexts.
- The Context Transition Rules, and their specification and incorporation within an appropriate context.
- The mission Objects and the specification and definition of their capabilities (For example, if the mission for a platoon of M-1 tanks, the tank's maximum speed, turning radius, fuel capacity, weapons load, and other boundary conditions must be defined)
- The identification, specification and definition of "Helping functions"; where examples of Helping Functions are that of finding the distance

between two points."

In CITKA, Gonzalez et al (2002) accentuate the identification, specification and definition of "helping functions." Hence, it is demonstrated that it is highly intuitive for a subject matter expert to provide much of the latter (Gonzalez et al, 2002). Subsequently, the knowledge engineer or the programmer will have to write the functions in an appropriate computing language. There exists a clear advantage here and it is that most of such developed functions will be easily reusable among different applications, thereby reducing the burden on the knowledge engineer or programmer.

The initial part of incorporating a model within CxBR is the building of its *context-base*. The *context base* is built by: (1) defining the contexts; (2) defining the procedures or context actions; (3) defining the context transition rules, and (4) identifying/defining necessary objects in the entity's environment. Once the context base for the model has been built, it is then incorporated within the *CxBR framework* and linked to the simulation of choice to be executed. The CxBR framework is the engine that exercises the knowledge represented a context base to achieve desired behaviors or actions. Accordingly, in this Thesis, it is assumed that the CxBR framework already exists, thus one's task is to:

- Specify the context base, and,
- Develop the context base

There are certain context-base constraints that would warrant the knowledge engineer's attention; for example:

(1) All Major Contexts should be associated with the Mission Context. In other words, no

Major Context should be allowed to float in the name space

(2) Context-switching (transition) functions and action definitions can be shared among contexts.

(3) Deleting a Major Context may eliminate all of its sub-contexts unless they are shared by other major contexts.

(4) Deleting a Context will eliminate its Action Definitions and Transition Criteria.

Once these constraints are kept in mind, the model can be fully developed by the knowledge engineer via coding through the use of appropriate algorithms. For the most part, this includes the creation of classes in an object-oriented with respect attributes' assigned values, and the instantiation of their objects. Generally, the class attributes are as follows (Saeki, and Gonzalez, 2000):

- **Constraints**: lists all the constraints that are imposed on the autonomous agent during a given mission.
- **Avoid**: describes anything that must be avoided by the autonomous agent throughout the scenario
- **Mission Objectives**: what will indicate successful completion of the mission
- **Major-Contexts**: lists the Major Contexts applicable to the mission.

  A Major-Context is also defined into a class in an object-oriented environment and contains the following attributes:

  (1) **Initializer**: References the name of the initializing function executed whenever the Context/Sub-Context is activated

(2) **Objectives**: states what the objective of the Context/Sub-Context is. Generally, it references a function that has some variables that are the goal of the respective Context or Sub-Context.

(3) **Compatible-next-Major-Context**: lists all those Major Contexts to which a switch from a context is possible.

(4) **Necessary-Sub-Context**: lists all Sub-Contexts, which are necessary with the current context. It is good to remember that Sub-Contexts are lower level autonomous procedures and as the sentinel rules and the control functions.

There are also certain assumptions that help form the basis of CxBR, and as described by Gonzalez and Ahlers (1998), they are as follows: " (a) Life for a autonomous agent is a continuous sequence of contexts, which change as the situation changes. A context can be likened to a situation that has been recognized, and which has a prescribed set of procedures that must be carried out, either sequentially, or arbitrarily. The behavior of a autonomous agent in the simulation is controlled by the context that is active for it at the time (Gonzalez and Ahlers, 1998), (b) the active context may not be the same for all autonomous agents. This is a reasonable expectation, since each autonomous agent may have a different mission, different sensor inputs, and different capabilities, (c) Contexts are represented temporally as intervals of time rather than time points. They are considered to be progressive stages to reach a goal, (d) Goals can be time points, but only to serve as transitions to other contexts, and (e) only a limited number of things can take place in any single context. Hence, a situation, by its very nature, will limit the number of others situations that can take place."

1.2 Contextual Graphs

Brézillon and Pomerol (2002) states that a "Contextual Graph (CxG) is an acyclic directed graph with a unique input, a unique output, and a serial/parallel organization of nodes connected by oriented arcs. It allows one to have a context-based representation of a given problem-solving scenario for decision-support processes while considering the environment in question. Brézillon and Pomerol (2002) point out that there are three types of context, namely: (1) external context, (2) contextual knowledge, and (3) proceduralized context. These types of context were created with Incident Management in mind (Brézillon's Paris Metro incident analysis via the SART CxG system). They allow a knowledge engineer to model various parameter types and the amount of information required at each step of the incident resolution process. In CxGs, a context is considered to be the information that may limit the possible decision field without directly intervening in the incident resolution process itself. For example, the time that the incident occurred is not as relevant as the occurrence of the incident itself. Nevertheless, potential resolution strategies observed are different whether the incident occurs at rush hour or not. Another good definition of context is "a collection of relevant conditions and surrounding influences that make a situation unique and comprehensible" (Brézillon, et al. 2003.) They showed strong relationships between context and knowledge, and provided an example of the application of these ideas in the SART application in the monitoring of a subway line.

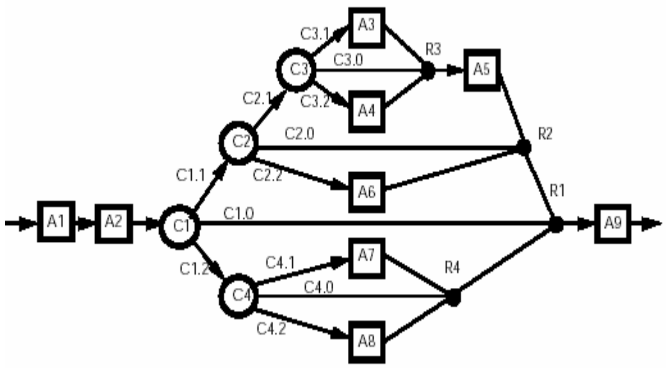


Figure 2: A pedagogical example of CxG (Brezillon, 2003a)

The initial 'procedure' established by a user company induces a CxG's initial structure. The CxG is then progressively enhanced by the practices used by the company's operators through the application of the company 'procedures' in different contexts. Figure 2 demonstrates the different types of nodes that are used to construct CxGs. In the graph, a square box represents an Action Node, where an action is an executable method. A pair composed of a contextual node and a recombination node represents a contextual element. A large white circle represents a Contextual Node by representing the explicit instantiation of the contextual element, while a Recombination Node is depicted by small black circle corresponding to the de-instantiated contextual element in question once the branch action has been accomplished.

Other graph components include, a sub-graph - an activity (not represented in Figure 2) that allows the modeling of operators' activities. An Activity (Human Operator Activity) is a complex action assembling different elements such as a CxG with a unique input and a unique output. Operators identify an activity as a recurring structure observed in different CxGs. It is an interesting identification because a change in a particular activity in a CxG appears automatically in all the other CxGs where the activity has been identified. Accordingly, it has to be noted that a sub-graph can be (see Figure 2): an action (e.g. A3), a sequence of actions (e.g. A1-A2), a pair of contextual and recombination nodes (e.g. C3-A3/A4-R3), or all the branches between a contextual and recombination node (e.g. the upper branch of C2 for the value of C2.1 with C3-A3/A4-R3-A5.) It represents a local reasoning (diagnosis/action structure) corresponding to intermediate goals. It is itself a CxG, directed, acylic, with one input and one output.

The ‘Action’ Context (e.g. A3, action 3) is composed of two parts, the Proceduralized context and the Contextual Knowledge. If we analyze ‘A3’ in Figure 2, we can define it in two parts, namely Action 3 (A3)’s Proceduralized Context {C1 with the value of C1.1, C2 with the value of C2.1, and C3 with the value of C3.1}, which can also be expressed as {(C1, C1.1), (C2, C2.1), (C3, C3.1)}, supposing that the actions A1 and A2 are realized. Action 3 (A3)’s Contextual Knowledge: is {C4}.

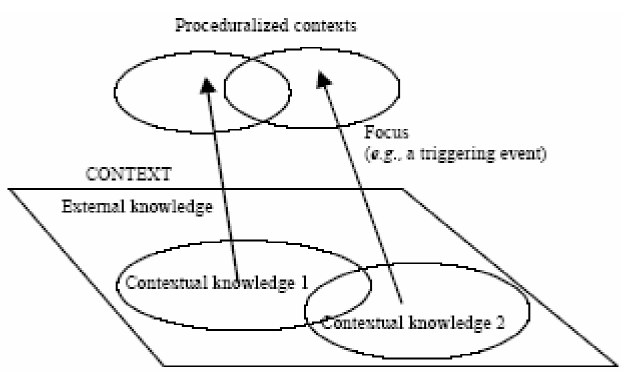


Figure 2a: The three types of context (Brézillon, 2003)

A Practice is provided by the operator as a sequence of actions on a corresponding path such as the following: {A1, A2, A3, A5, and A9}. The latter can be intertwined with contextual nodes or node values C1.1, C2.1, C3.1, and recombination nodes (R3, R2, and R1) to be more clearly represented as follows: {A1, A2, C1.1, C2.1, C3.1, A3, R3, A5, R2, R1, A9}. Contextual Nodes (C's) and Recombination Nodes (R's) are used to represent a 'contextual element'.

A contextual node corresponds to the explicit instantiation of the contextual element. A Contextual element is instantiated only between a contextual node and the corresponding recombination node. When a piece of contextual knowledge becomes instantiated at a contextual node, it enters the 'proceduralized context'. Thus, a contextual change corresponds to the movement or transition of a piece of contextual knowledge into the proceduralized context, or conversely from the proceduralized context into the contextual knowledge which is the respective two parts of an action context. The construction of a 'proceduralized context' involves the use of contextual knowledge from possibly various domains. CxGs are empowered with a "spindle" general structure by using contextual and recombination nodes identified by a divergence of branches at a contextual node initiated by a diagnosis, and a convergence of the branches at the recombination nodes in relation with the realized actions/activities (Brézillon, 2003b).

1.2.1 CxG Implementation

Brézillon and Pomerol (2002) explain that the SART system represents a CxG implementation. "Knowledge for that project is managed and engineered as follows:

(1) SART must deal with an explicit model of what context is for traffic

operators.

(2) SART must have access to the traffic computer in order to be able to detect the beginning of a traffic incident and alert a traffic operator accordingly

(3) SART should be able to gather a large number of contextual data that define the context in which the incident is solved.

(4) SART must first retrieve incidents that occur in the subway line in question, then on similar lines, and finally on all other lines."

As a measure of precaution, Brézillon and Pomerol (2002) argue that operators cannot be disturbed by SART during the decision making process. However, when they are off duty, they may opt to use SART for help in: " (a) Repetitive tasks (e.g., the writing of the report on the incident), part of which SART can help fill automatically, pointing out missing information, (b) The analysis of the incident that has just occurred, (c) Intelligently facilitating the interaction between two operators discussing an incident (e.g., for training purposes), (d) Providing the history of events that occurred prior to an operator's work shift, (e) Facilitating a presentation of exceptional events that could help operators in their decision making process, or enrich their experience and (f) Providing an ordered list of incidents as per some criteria."

As demonstrated in Figure 2a, a CxG involves three types of context, and is a representation of a given problem-solving scenario for decision-support processes by taking the environment into consideration. Thus, the initial CxG structure is a procedure established by a respective organization. The latter evolves as the procedure is applied in different contexts, hence adding new practices to the CxG. Thus, a practice is the application of a procedure in a particular context. In the case of applying the SART application framework to the Paris Subway line, most of the incidents have been well known for a long time (object on the track, lack of power supply, suicide, etc.), Thus the responsible company, RATP, has established procedures for incident solving on the basis of their experience. However, operators tend to develop individual, if not unique practices when it comes to incident solving with a given procedure.

Brézillon (2003) analyzes context proceduralization in a very interesting manner. ‘Context Proceduralization occurs with the passage from ‘Contextual Knowledge’ to ‘Proceduralized Knowledge’ and it is said to be task-oriented because of the following reasons:

- Depends on the *focus* on a task (e.g., a triggering event)
- Triggered by an event or activated by the recognition of a pattern
- Performed by individuals that transform contextual knowledge into some form of functional knowledge or ‘causal and consequential reasoning’ in order to anticipate the result of their own action

This type of context proceduralization occurs under the following two conditions:

(1) There’s a need for a consistent, explicative framework to anticipate the results of a decision or an action. This consistency is obtained by reasoning about situational causes and consequences.

(2) A conscious reasoning about causes and consequences.

The SART application framework was used to develop Brézillon’s (2003) ideas in a CxG representation of knowledge and reasoning. The CxG formalism itself can be used in any other framework consisting of practices that are developed by operators from the procedures imposed by a certain company.

As mentioned earlier, the building or construction of the proceduralized context represents some functional knowledge or causal and consequential reasoning, essentially, building a proceduralized context model. In that case, the context base includes the procedure established by the company because they are the ones that will help determine the External Knowledge. The decision-making or task-performing process emanates from that External Knowledge to warrant the creation of contextual knowledge that can be subsequently proceduralized based on the task at hand. In Figure 2b, the notion of External Knowledge is stated with regards to a decision-making, or task performing process. Thus, External Knowledge is the part of the decision-making or task-performing process that is not relevant at a point in the process, whereas Contextual Knowledge is the relevant part of the process at a point. Moreover, if this Contextual Knowledge is proceduralized, it is then called a proceduralized context.

Hence, the proceduralized context is the one that is immediately useful for the task at hand. This type of context construction is often a process of communication in a community of practice; even the members of that community come from different domains. In Brézillon et al., [1999, 2000], the dynamic of the environment is taken into consideration so that appropriate assumptions can be reached. In order to do so, one needs to take the evolution of physical factors (e.g., user location, request time), user and environment contextual knowledge into consideration.

1.3 Why should there be an effort to compare CxBR and CxGs?

A comparison of two modeling formalisms like CxBR and CxGs presents an opportunity for an unfamiliar reader to not only discover the essence of two well-established context-based formalisms, but also understand their applications of contexts, particularly in the field of artificial intelligence. Although, both formalisms have their own area of concentration, this comparison will allow future researcher to understand how the different aspects of the two formalisms may either be used synergistically or exchanged for independent use.

The idea that CxBR formalism is more suited for an autonomous environment, whereas that of CxGs is typically for a decision-support-centric one, is quite understandable. However, it should not limit the context-driven researcher in analyzing, or deriving innovating ideas that could possibly be by-products of both formalisms as they relate to autonomous environments (e.g. unmanned submarines, tank platforms, aircrafts) or decision support driven environments (e.g. subway operations, or workplace decisions.) For instance, an exposure to both formalisms may induce a researcher to argue that the Paris Metro environment, typically depicted to be decision-support in nature, is also a autonomous one. Hence, decisions taken by operators in the Paris Metro may then be considered to be comparable to those taken by warriors in an autonomous environment.

Accordingly, conducted simulations may allow future researchers to scrutinize how important aspects of both formalisms can be interchanged as both are either used synergistically or independently. For instance, an area of exploration could be the use of the CxBR framework to incorporate autonomy in CxGs, or alternatively that of using CxGs into the CxBR framework in other to provide a decision support feature to users.

1.4 Autonomy vs Decision Support

In today's world, computers continue to increasingly participate in decisions that affect human lives (e.g. Medical expert systems, automated pilots). Accordingly, the exploration of paradigms like CxBR, which was conceptualized to provide reasoning in an autonomous environment (e.g. HBR Challenge project, automated pilots), and CxGs, on the other hand, which was developed to work in systems that provide decision support (e.g. SART, and medical expert systems.)

Autonomous systems are very important when human presence is either not advisable due to hazards or simply too costly. On the other hand, decision-support systems are important when human-decision making needs to be aided or supported to promote accuracy or consistency. As previously introduced, CxBR is a context-based human behavior representation paradigm that promotes the simulation of human behavior in an autonomous environment. This type of autonomy is important for unmanned operations such as those that happen in the military or space exploration when the use of human beings is strictly prohibited or not reliable enough. However, when human presence is necessary, it is useful to use context based decision support formalisms like CxGs to promote accuracy and consistency (e.g. SART). For instance, in DMSO's HBR challenge project, CxBR was used for the context-based simulation of human behavior guiding submarine vessels. On the other hand, in the Paris Metro, CxGs was used via the SART application to provide decision support to train operators as a means of problem solving or task facilitation to promote consistency or accuracy. They will both be explored so that conclusive comparative insights can be reached either regarding synergistic use or the use of features of one into the other.

CHAPTER TWO: LITERATURE REVIEW

This chapter provides a review of the CxBR and CxGs literature in an effort to provide more insights to the reader about both modeling formalisms. It reviews their individual approach in addressing contexts and their representation through computational paradigms.

2.1 CxBR Literature Review

The literature amplifies that CxBR is an automated reasoning paradigm that provides a simple and effective way of simulating human autonomous behavior by using "an intuitive identifier called a Context" (Gonzalez and Saeki, 2000). Fernlund and Gonzalez (2002) asserted that CxBR is based on the concept that humans think and act in terms of contexts. Accordingly, the role of a Context is to provide a means of:

- Addressing all conditions in a current situation
- Controlling the behavior of an intelligent agent in a autonomous situation (Gonzalez and Saeki, 2000).

Therefore, in the event of a situational change, a context that is currently active proceeds to search for another context that better addresses the conditions of the new situation. Once such a context is found, the previously active context deactivates itself and activates the newly selected context. Thus, it is easy to come to the conclusion that context transitions can be used to intelligently control an intelligent agent through situation-based and continuous context transitioning.

2.1.1 CxBR and Competing Context

Gonzalez and Saeki (2000) emphasized the level of difficulty involved in "hard-coding" such situation-based context transitions. Although it may be possible to "hard code" context transitions for some situations (e.g. taking actions upon getting order from a superior), it is rather unrealistic to do so in all situations. Why one might ask? The system developer would have to either be able to predict all possible situations that an intelligent agent may face, or develop and "excessively large and complex set of Contexts" that are ready to be activated via a context transition when a specific condition in the simulation is met. Both of these ideas are quite unrealistic. Thus, Gonzalez and Saeki (2000) introduced the "competing context" approach, which was developed solely to address situations where several contexts seem to be acceptably equipped to address a situation. Thus, the goal of the Competing Context concept is to determine the context that is best equipped to deal with the new situation, and its immediate goal through the use of a "constraint-based technique and time warp simulation." They argue that besides "soft-coding the tactics, the Competing Context concept provides yet another benefit which can pave the way for easy online learning." Moreover, Gonzalez and Saeki (2000) argue that the competing context concept goes beyond being a means of "soft-coding the tactics". Essentially, it provides an additional benefit that can pave the way for easy online learning.

In Gonzalez and Ahlers (1998) an interesting description of CxBR is provided based on the following assumptions:

- "In the world of an Intelligent Agent, life is a continuous sequence of Contexts, which change as the situation changes. A Context may be like recognized situation with a prescribed set of procedures that must be carried out either sequentially or arbitrarily. Hence, in a simulation, an intelligent agent's behavior is controlled by the context that is activated for a particular situation in a set moment in time.

- The active context may not be the same for intelligent agents. The latter is reasonable expectation, since each may have a different mission, different sensor inputs, and different capabilities.
- Contexts are represented temporally as time intervals rather than time points. They are considered to be transitions to reach a goal.
- Goals can be time points, but only to serve as transitions to other contexts.
- Occurrences that can take place in a single context are limited. Hence, a situation will limit the number of situations that can take place.
- The presence of a new context will alter the present course of action and the applicable expectations to some degree" (Gonzalez and Ahlers, 1998.)

2.1.2 CxBR, Computer Generated Forces (CGFs) and CITKA

Gonzalez et al (2002) investigate "means to semi-automatically build" Computer Generated Forces (CGF). It is an interesting investigation because over the past decade research in CGFs has focused on better ways of representing autonomous human behavior for training and analysis. This investigation allowed them to discover that by using CxBR they can reduce the effort required to build the CGF models, as well as reduce any corresponding errors. CxBR is well-structured, and possesses a hierarchical organization, allowing it to facilitate knowledge acquisition through an automated query system that they called the Context-based Intelligent Tactical Knowledge Acquisition (CITKA) system. The approach they use in CITKA is based on an intelligent query session between the SME and the CITKA system, which uses its own knowledge base to intelligently compose the queries. CITKA selects the next question based on the SME's previous replies. In addition, it has a feature that allows a Knowledge Engineer (KE) to refine knowledge that's been previously entered by the KE.

In order to build a CGF model, one begins with the specification of the capabilities of the model. The specification is based on the specific mission, and as expected, task force assets differ from mission to mission regardless of their size. Once specified, the CGF model is developed by building the context base, that is, by defining the contexts, the procedures for context actions, the transition rules between contexts and the necessary objects (Gonzalez et al, 2002).

The CITKA system consists of four independent yet cooperating modules, which are as follows:

1. Knowledge Engineering (KE) Database Backend
2. Query Rule-base Backend
3. Knowledge Engineering (KE) Interface
4. Subject Matter Expert (SME) Interface

"The SME Interface module maps into the Query Rule-based Back end, while the KE Interface module maps into the KE Database Backend module. The latter is a data structure that holds the evolving context base, as it gradually become more developed, either by the KE or by the SME. On the other hand, the Query rule-base Backend is a rule-based system containing the rules for executing the intelligent dialog with the subject matter expert" (Gonzalez et al, 2002).

The effectiveness of the CITKA system was evaluated and two main issues were addressed: (1) Estimating the reduction in person-hour effort to develop a context-based model for a particular mission and (2) Estimating the percent of a context-based model that could automatically be developed through CITKA. (Gonzalez et al, 2002).

2.1.3 CxBR and Genetic Programming (GP)

The previous section indicates the significant progress that CxBR has made in the area of modeling CGFs in training simulations. However, this accomplishment was followed by a substantial problem with the time and effort that it takes to accurately collect meaningful knowledge from SME's for the development of intelligent (simulated) entities (i.e., models of autonomous human decision-making). This is a challenge that required the development of new tools and methodologies in order to automate the creation of such simulated entities. Fernlund and Gonzalez (2002) describe one way of accomplishing the development of such a tool through using CxBR in conjunction with Genetic Programming (GP).

GP allows computer programs to evolve into new and better programs by themselves. Fernlund and Gonzalez (2002) emphasize that it is inspired by Darwin's theory of evolution by creating a "population" of computer program using operations and selection mechanisms. The result of the genetic process is a computer program that will solve most types of predefined problems in almost any area, such as, classification, planning, mathematics, optimizing, and control" (Fernlund and Gonzalez (2002). Hence, in their collaboration, Fernlund and Gonzalez (2002) present a new approach to automatically create intelligent simulated entities. The automation of this creation process is catalyzed by learning from observation, a strategy that involves learning the behavior of an SME through a mere observation of his or her behavior. They knew that the need for such a tool or methodology had already been identified for some applications, such as after-action-review programs, modeling human behavior in battlefield simulations and in street traffic flow simulators. Accordingly, their developed tool automatically creates models of simulated entities.

2.1.4 CxBR and Intelligent Simulation

The modeling and simulation of human behavior or human-controlled entities is a very important research area. Norlander (1999) emphasized the importance of human behavior simulation in agents as a field in itself. This emphasis is substantiated by examples of applications in military battle simulations, cars in battle simulators, and games. Perhaps the best example that Norlander (1999) provided was that of populating a "battlefield in a war game in order to provide student training in a more realistic environment. In this case, agents can be tanks, aircrafts, submarines, or dismounted infantry, and the main purpose of simulating agents is to lower training cost by replacing human operators. Hence, Norlander (1999) presented two different types of agents:

1. Semi-Automated Forces (SAF), which are partly controlled by a human operator.

2. Autonomous Intelligent Platforms (AIP) which are self-operated.

In order to model autonomous agents, Norlander (1999) decided to use CxBR as his modeling paradigm since it is not only an effective and efficient tool to model human behavior, but also it is particularly well suited for modeling autonomous behavior. The focus of his thesis was to formalize CxBR by doing the following:

- Provide a definition of its usable functionalities
- Develop a tool (that he named the CxBR Framework) for easy development of agents using the CxBR paradigm.
- Identify concepts that are likely to be incorporated in CxBR in the future (e.g., Cooperating Agents and Temporal Reasoning)

In designing the CxBR Framework, Norlander (1999) envisioned a flexible tool that would allow future research and enhance execution compared to the previously used CLIPS-based system. His research yielded the development of two car simulations:

- The 'Rural Road' Simulation- models the behaviors of cars driving in a rural setting
- The 'CityTraffic' Simulation - A more extensive system that models the behavior of cars, pedestrians, traffic lights, and traffic control.

Implemented behaviors included, normal driving on urban roads, suburban roads, and highways, following and passing cars, stopping and yielding in intersections, and stopping for pedestrians crossing the street (Norlander, 1999). His work undoubtedly provided a great contribution to the CxBR paradigm not only by enhancing its execution speed, but also by creating an evolutionary framework upon which future research will flourish.

One of those future research attempts include that of Gerber (2001) in an attempt to provide synchronization of behavioral models of human-controlled vehicles with actual vehicles in order to greatly increase and sustain the accuracy of such behavioral models. In other to effectuate such a goal, Gerber (2001) used "a hierarchical, context-based representation, whereby the behavioral model, located on the other vehicles in an embedded simulation, performs the actions that are appropriate for the behavioral context and sub-contexts of the actual vehicle it represents. However, the model has to know what the current behavioral context of the human-controlled vehicle is in order to respond with the correct actions."

The focus was first on the recognition of the behavioral context in real time, and then the synchronization of the distributed behavioral models with the actions of the human-controlled vehicle. In order to facilitate behavior recognition, template-based reasoning is used where each template is a representation of each behavioral context and sub-context. Hence, the weight given to each template becomes very critical, since it helps in correctly selecting the template that identifies the "current behavior, and is based on weighted attributes of the vehicle's state and its surrounding environment" (Gerber, 2001).

Gerber (2001) work effectuated a research certainly developed and implemented a "novel" methodology for learning by observation through the use of fuzzy membership sets and neural networks to automate the setting of template attribute weights. Hence, allowing "significant discrimination between different categorized behavioral contexts or sub-contexts on a human-controlled vehicle” (Gerber, 2001).

2.2 CxGs Literature Review

As indicated in earlier chapters, Contextual Graphs (CxGs) is a context-based formalism, which had been initially developed in the SART application (French acronym for support system in traffic control) for the development of a support system in incident solving on a subway line (Brézillon, 2003.) Indeed, the SART project involved the design and development of an intelligent support system for subway line traffic regulators. The ultimate goal was to have a more efficient way of handling the difficult task of incident management especially considering the role that context plays in it. The first terminology that was used to represent the first phase of the formalism was *decision graphs* because decision graphs are reasoning models derived from decision trees (Pasquier et al., 1999.) They behave as their parents, but integrate the difference between contextual knowledge and proceduralized context and the dynamic switch of those knowledge states. In decision trees, branches diverge according to the possible choices and never converge even if the choice is no more relevant. This led to trees with identical action sequences on several branches. In such a case we decided to merge the common part of the branches. The structure is no more a tree but a directed graph, called decision graph.

The evolution from decision trees to decision graphs is based on the fact that a contextual piece of knowledge is proceduralised at the step where it intervenes in the choice. This choice generates several branches (representing the different possible action sequences). When the branches are merged, the proceduralised piece of context retrieves its contextual piece of knowledge state." (Brézillon et al, 1999.) Subsequently, based on the contextual nature of the formalism, it was renamed to Contextual Graphs (CxGs). The name is appropriate because in order for SART to be an efficient assistant, it had to deal with an explicit model of exactly what context is for operators. Indeed, their reasoning process in arriving to the final paradigm was a context-based representation of the domain knowledge and a context-based representation of operators' reasoning during an incident solving.

2.2.1 CxGs and Decision-support Processes

The idea of using CxGs involves a working environment where decision-support processes are already established. Thus, when a problem occurs, it needs to be solved as quickly as possible, which requires a modeling paradigm like CxGs in order to provide a context-based representation or model of the problem-solving process. Generally, a procedure established by an organization creates the initial structure of a CxG. Subsequently the operators that work for such an organization apply this procedure through their practices in different contexts, hence progressively enhancing the CxG (Brézillon, 2003.) In a CxG, a practice is represented by path on the graph. A practice is comprised of operators' actions, which are intertwined with the contextual elements as considered by a respective operator. Contextual elements use the differences in actions that are present in a practice to differentiate one practice from another. Hence, producing different instantiations for such practices.

2.2.2 CxGs and Incremental Knowledge Acquisition

The CxG formalism uses a graphical user interface in a manner similar to that used with CxBR by Gonzalez et al, (2002) in order to facilitate knowledge acquisition. This interface provides a graphical representation of the current state of the CxG, thus allowing the operator to interact with the system in order to identify which sequence of actions was used in a problem/incident solving session. Upon solving a problem, the operator instructs the system of the sequence of action that was used to solve the problem. Subsequently, the operator informs the system of which practice is closest to the entered action sequence (Brézillon, 2003.) The sequence of actions entered by the operator may be known or unknown. However, if it's an unknown sequence, the system will prompt the operator to provide a respective definition (contextual element), location (position of the contextual and recombination nodes on the path) and instantiations for the known and entered practice. Accordingly, it is said that the added contextual element came from external knowledge since its instantiation was previously irrelevant. However, it is then instantiated in a very specific way in the new practice, hence allowing movement from external knowledge to contextual knowledge, and right through its use in a proceduralized context. (Brézillon, 2003.)

In short, a CxG is capable of evolving simply by "assimilating and accommodating entered operators' practices in order to form a kind of corporate memory” (Brézillon, 2003.) Thus, once a new practice is created, the CxG in turn creates a corresponding contextual-recombination node pair for it.

2.2.3 CxGs and Explanation Generation

The goal of generating explanation depends on one’s knowledge of the domain in question. In the case of CxGs, it concerns the involved tasks at hand and actions - definition, input, and output (Brézillon, 2003.) However, according to Brézillon and Pomerol (1997), when there is a lack of consideration for the contextual aspect of the domain knowledge, explanations will only bring limited insights to the user. Mackie (1965) stresses the context-dependency of explanation as a process of making a distinction between some current situation and other class of situations. Thus, “(when it comes to deciding the quality of an explanation, context is quite relevant. Thus, using a theory of contextual influences may be able to determine which explanations are appropriate (Brézillon, 2003.) Another example is that of Leake (1992), which considers the relationships between explanations and context in the framework of case-based reasoning. An explanation is required when there is a conflict between an event and a model that we have of the place where the event occurs. Accordingly, Brézillon (2003) points out that the advent of CxGs consider the insights behind those two points thus, allows the following to take place:

- Explicit Context Representation
- Acquisition of knowledge in the right context
- Explanation generation from all items in a CxG, e.g., contextual elements,

actions, activities etc.

Accordingly, Brézillon (2003) indicates that the explanation generation of a practice simply depends on:

- The presentation of the different contextual elements intervening along the path.
- The order in which these different contextual elements intervene.
- “Temporary instantiations and the temporal chronology in which they have been incorporated in the CxG.”

On the other hand, in order to generate explanation of an action in a practice, one relies mainly on that action’s preceduralized context. In other words, the system is capable of tracing the reasoning approach taken prior to such an action in the practice, and thus presents the following:

- “The contextual elements explicitly used in the practice until the action,”
- “The instantiations of these contextual elements,”
- The order in which different contextual elements are instantiated
- The order in which (and the reasons why) the contextual elements have been introduced in the contextual graph.

Indeed, Brézillon (2003) emphasizes that the last two points encompass a way of allowing the explanation to take into account the context dynamics leading to the action that one is trying to explain. This allows the system (or one) to view such an explanation as a process in progress, rather than directly deriving it from known or static factors. Consequently, as the user (operator) continues to interact with the system on the same

CxGs, they can provide each other with relevant explanation to effectuate better dialogue. In fact, the system proceeds to enter a phase of incremental knowledge (practice) acquisition when the user inputs explanation into it. This explanation in turn enhances its reasoning capability.

Now that literature reviews of both modeling formalisms have been provided, the next section will provide an exploration of the problems/challenges that both formalisms have to be able to tackle in order to be efficient. Section 3 covers the HBR challenge project for CxBR, and the Paris Metro project for CxGs, hence giving the reader with an exposure to two research projects involving the successful use of the respective formalisms.

CHAPTER THREE: PROBLEM DEFINITION

This investigation pursues a comparison framework that will either allow strengths from one of the formalisms to be transferable into the other, or combine their strengths to achieve better results by allowing them to work hand in hand. However, this is a difficult task. A desirable goal would be to develop the aforementioned ‘CxG-CxBR Synergy Tool’ as a platform to test the effectiveness of either sharing or combining the strengths that exist in both formalisms (modeling techniques) in order to potentially determine how they can be customized for particular applications. On the other hand, if one considers the alternative of contrasting the two formalisms, a better understanding on what context is and the mechanisms by which context evolves can be reached. For example, the dynamics of context is linked to the entrance or departure of an item in the focus of attention. In CxG, this corresponds to the movement between the contextual knowledge and the proceduralized one, whereas in CxBR it is characterized by the transition between two contexts, as indicated by the context transition pair. Chapter 4 introduces metrics used for the comparison framework. However, this chapter focuses on introducing aspects of the two applications of both formalisms that will be used in Chapter 5 to perform the cross application analysis, namely, the CxG-based SART project, and the CxBR-based Human Behavior Representation (HBR) application and the other way around.

3.1 The SART Project for the Paris Metro

The goal of the SART project can be readily simplified into deriving a way of representing decision-support knowledge via an intelligent assistant system in order to assist operators in the Paris Metro in arriving at practical decisions. It is not an illogical goal because these operators have already been using existing general procedures, most of which have been in existence since the mid-1900's either for troubleshooting or for solving problems.

The goal of designing an intelligent assistant system whose aim is to support the decision of the subway operators in subway control involved the *modeling* of the operators' activities and knowledge. Accordingly, the CxG model was introduced as a simple solution to describe and manage decision-making. How do operators react when an incident occurs? Essentially, when an incident occurs, they have to react rapidly in order to: "(1) Devise a representation about the issue, (2) Collect information on the situation, (3) Analyze the incident and (4) Implement the corrective actions" (Brézillon et al, 2001.) There is, however, a problem with the use of such general procedures because they tend to not put into perspective the contextual elements of the issue in question. This is particularly important because modern organizations have to deal with increasingly contextual considerations, hence the need to represent and manage contextually dependent procedures. Thus, interaction between the operator and the system is of utmost importance, and for that reason Brézillon et al, (2001) tried to synchronize the system reasoning to that of the operators. Indeed, as required to fulfill the latter, they proceeded to analyze the decision-support knowledge used by operators and record it in such a way that it is adaptive enough to be easily understood by the operators and efficiently used by the computer (Brézillon, Pomerol and Pasquier, 2001.)

3.1.1 Paris Metro Decision Support Knowledge (Practices)

Metro Line Representations/viewpoints

There are several viewpoints that can be used to represent a subway line in the Paris Metro (Brézillon and Pomerol, 2001):

1. **Travelers' Viewpoint**: A succession of *stations* and *interstations*, where *interstations* are the rail track portions between two successive stations.
2. **Electrical (Power Supply) "sectioning/sub-sectioning" Viewpoint**: In order to power all the trains on the line, several power rectifier-substations are needed. This allows a track to be divided into several sections where each section is being powered by at least one power rectifier substation. Accordingly, each section is independent from the others. Moreover, to reduce the impact of an incident on the traffic, each section is sub-divided into subsections.
3. **Operation/Organization Regulatory Viewpoint**: In order to regulate a subway line, two main classes of operators are needed. The first class of operator is the "Local Control Point (LCP) agents" and their job constitutes the management of the trains and their departures times. The second type of operator is the "Centered Control Room agent" (PCC in French), whose jobs are essentially traffic supervision and incident solving. These two classes of operators work at different

places and use the telephone as a means of communication so that the Centered Control Room operators can also communicate with train drivers, station managers and the supervisor who is connected to the emergency services (Brézillon and Pomerol, 2001.)

Operators base their decisions on what the current principal issue of concern is and thus, use the above viewpoints in an appropriate manner. For instance, if the issue were more of a traveler concern, the first viewpoint would immediately apply. Likewise, if a technical problem has been located and has caused a long-lasting incident, operators would choose the second viewpoint in order to limit the impact that the issue may have on residual traffic. An example that applies to the third viewpoint is one in which the class-1 (LCP) operators (responsible for train management and departure time) communicate a train delay to the class-2 (PCC) operators (responsible for traffic supervision and incident solving.) In short, this exemplifies why the three viewpoints are needed in order to solve most incidents. Thus, in order to summarize all three viewpoints the following has to be kept in mind by all operators (Brézillon and Pomerol, 2001):

1. The security of travelers is an important contextual factor that constrains incident solving.
2. An incident should have as few repercussions as possible on general traffic.
3. Appropriate actions (e.g. redirection) have to be taken in order to maintain traffic regularity, even if it interferes with the Center Room Control agents' work.

3.1.2 Paris Metro control organization

Each subway line has two endpoints (a.k.a. a terminus.) There is a principal endpoint and one or more secondary endpoints, and each of them has a local control point to control the train departures. The LCP operators also have an additional responsibility besides those mentioned earlier. They are also responsible for traffic around the endpoints in order to "choose which train will start to order the departure according to the theoretical timetable and adapt it to the actual conditions (Brézillon, and Pomerol, 2001.) One of the idiosyncrasies of the Parisian subway line is that the PCCs of all the lines (except a new line called METEOR that is entirely automatic) are in the same room. This allows operators that are responsible for different lines to work hand in hand to solve different types of incidents. In addition, team formations and turnover rate definitions are used to further organize operators. Thus, such a level of organization allows each operator to rapidly share each new experience. Hence, they form a community of practice (Brown and Duguid, 1991), and a community of interaction (Nanoka, 1994.) This is important to explain the construction of a shared knowledge and a collective resolution of the problem, which is why from this level on the problems are considered in the viewpoint of the operators (Brézillon and Pomerol, 2001.)

Each subway line operator is assigned to a schedule to cover part of the day, which corresponds to a particular *control console* that allows him/her to:

- Cut the power in any section
- Stop trains at each section

- Use high frequency telephones to communicate with train drivers
- Use automatic telephones to communicate with endpoint (terminus) operators, station operators, local operators or exploitation supervisor.

In order to provide the operator with a good representation of the line, a large *synoptic display* (called TCO in French) is used. It also provides information about line sectioning and sub-sectioning, the stations and the train position. In addition, the TCO also controls some commands such as energy commands and switching commands (Brézillon and Pomerol, 2001.)

In a nutshell, the job function of each operator involves actions that are mainly concerned with:

- Train regulation (delaying)
- Train redirection
- Section and sub-section power-cutting and power supply
- Coordination of event or action from local or external agents in order to gather appropriate information.

The role that the operators play is very important because it allows them to act as a two way communication channel in order to dispatch information from local agents (drivers, station agents, endpoint agents, or locally-situated executives), exploitation and line executives." (Brézillon and Pomerol, 2001.) The operators relay all that information and consequently they act as the coordinators of all the people solving the incident including the managers of the necessary resources. This system of coordination works because in the event of an incident on a particular line, the operator who is responsible for such a line automatically becomes the *incident manager*. However, there are other operators that should be in a position to assist the incident manager in assessing the incident, and they are appropriately called the *assessors*.

Thus, on one hand, the incident manager is stationed at the control console in order to respond to phone calls, control the trains and make appropriate decisions. On the other hand, the role of the assessors come into play and mainly consists of providing help in the following (Brézillon and Pomerol, 2001.)

- For all incidents that require line power control, an assessor stays at the TCO to either cut or re-establish power on sections or sub-sections and for possible train redirections.
- For more important incidents, a second assessor observes the activity, provides advise to the incident manager as per the final assessment, and takes notes on the steps and procedure taken to resolve the incident (e.g. time of the actions and events, train number, location of train redirections, etc.)

In the end, when the incident is deemed resolved, the incident manager writes a report containing the description of the incident and the corresponding actions that were taken in order to effectively resolve it.

3.1.3 Paris Metro line decision-support knowledge (practices) representation – and underlying hurdles

Brézillon and Pomerol (2001) put particular emphasis on the difficulty that one is bound to face in regards to **modeling** decision-support practices. Essentially, there are three main hurdles that constitute this difficulty:

1. The first hurdle lies in the fact that there are many decision-support practices.
2. The second hurdle accentuates the fact that those decision-support practices are often implicit within the community of practice (Brown and Duguid, 1991) and are strongly linked to one another.
3. The understanding that the context in which these decision-support practices are applied. This also involves the understanding that such practices are all too "often dynamically constrained in sequences of actions." (Brézillon and Pomerol, 2001)

Therefore, in order to gather and study decision-support practices, Brézillon and Pomerol (2001) had to record the subway line incidents as a set of characteristics, which includes context description and the action sequence applied to get them resolved. Subsequently, the gathered data was used to construct an adapted representation of these decision-support practices, which in turn is used to collect and organize this knowledge type, or for reuse purposes.

3.1.4 Decision Support knowledge (practices) representation modeling evolution– From contextual/decision trees to contextual graphs

Pomerol and Brézillon (1999) were determined to understand and model the role of context in reasoning, "for the sake of engineering applications." Accordingly, they decided to work on the control of one of the Paris Metro lines (Pasquier, Brézillon, Pomerol, 1999.) These observations were what allowed them to define contextual knowledge as part of the context (where the complementary part of the context is called the *e*xternal context) that is relevant in a given situation for a given operator (Brézillon and Pomerol, 1998). Thus, the contextual knowledge can be perceived as subset of the context. Therefore, the operator can use it to find every chunk of knowledge for:

- Reasoning about a situation
- Interpreting a situation
- Explaining a situation

In addition, Brézillon and Pomerol also defined the notion of a proceduralized context, which is the proceduralized part of the contextual knowledge, which is considered explicitly with causal and consequential links at a given step of the problem solving or incident resolution process (Brézillon and Pomerol, 1998.) When an incident occurs, the focus of the operators remains on the proceduralized context part of the contextual knowledge at each step of the problem solving process (see Figure 2a.) They accentuated that the proceduralized status of a chunk of context from the contextual knowledge is not permanent because the proceduralization of a piece of contextual knowledge only happens when a particular operator focuses on it. Thus, when the operator ceases to focus on it, it returns back to the contextual knowledge form. Hence, it is then no longer active in the reasoning. Thus, Brézillon et al, (2001) started out with using a decision tree modeling/representation approach to represent an official procedure (e.g. a lack of train power incident.) the decision tree representation was made of two types of elements, namely the actions (directives to do an action - rectangular boxes) and the contextual nodes (select a branch depending on the knowledge about the current context – circles.) There was a problem with this type of modeling approach because it spanned in the unacceptable way. In fact, this particular tree structure expanded significantly when it came to the representation of highly contextual decision-making in complex applications (Brézillon and Pomerol, 2001.)

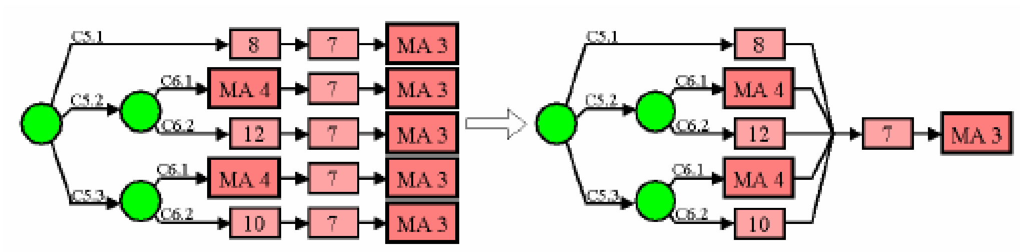


Figure 3: From Tree to Graph (Brézillon and Pomerol, 2001)

Brézillon and Pomerol (2001) indicate how the need to arrive at a better representation is what allowed them to understand what they called a “scarcity principle” that induces operators to use well-known procedures as soon as possible when they encounter a problem. Therefore, this reasoning had a great impact on arriving to the model approach, which took more of a form of a graph, thus named a contextual graph (CxG) because with this reasoning (Brézillon and Pomerol, 2001):

1. The representation/model is oriented without any circuits with exactly one root and one goal (Figure 3). This is because the operators have only one goal, which is that of resolving the incident and return to a normal operation. It also transitions from using a sequence of actions to one micro-action. The branches were

established to represent different strategies depending on the context used to achieve a particular goal. Moreover, the graph structure allowed them to extend the representation

2. They now had a way of keeping the size of the structure under control. Although the introduction of a new contextual element will add some elements in the graph, it will not drastically increase its size.
3. The transition from a decision tree model to a contextual graph (CxG) model introduced a dynamics that is essentially comparable to the dynamics of a transition between proceduralized context and contextual knowledge. In a CxG model, when two branches are merged, one knows that "actions that have been undertaken have led to a common situation from different contexts." (Brézillon and Pomerol, 2001.) Moreover, the contextual elements that are attached to the different branches are proceduralized at the diverging node, and because they intervene in the branch decisions, they don't have to change for different action sequences (hence no expansion.) Last but no least, they are deproceduralized when the branches are merged. Accordingly, the life duration of the contextual elements are expressed explicitly (Figure 4.)

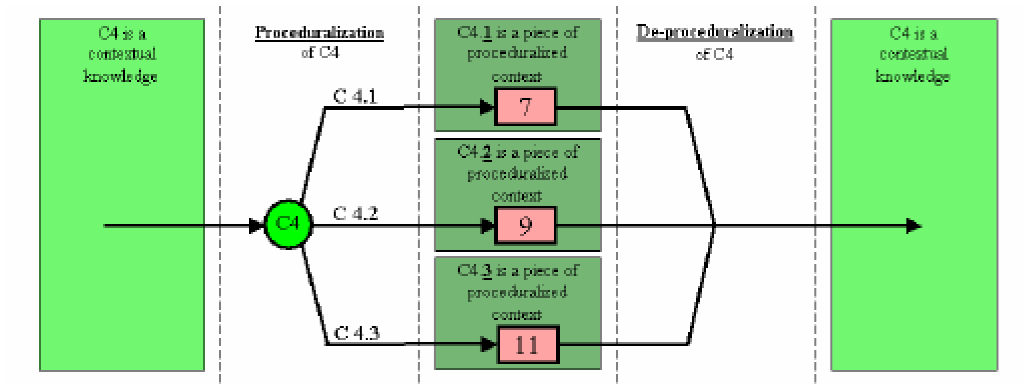


Figure 4: Proceduralization and de-proceduralization (Brézillon and Pomerol, 2001)

In Figure 4, Brézillon and Pomerol (2001) provide an example of two trains on the same line and have to unload passengers in various necessary unload orders.

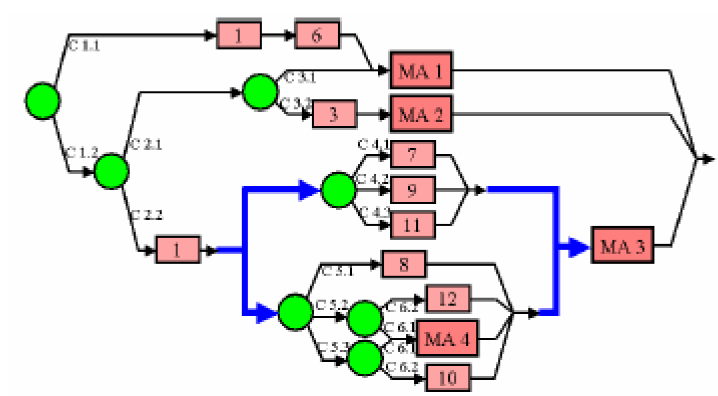


Figure 5: Contextual graph representing the official procedure for the "lack of train power" incident (Brézillon and Pomerol, 2001)

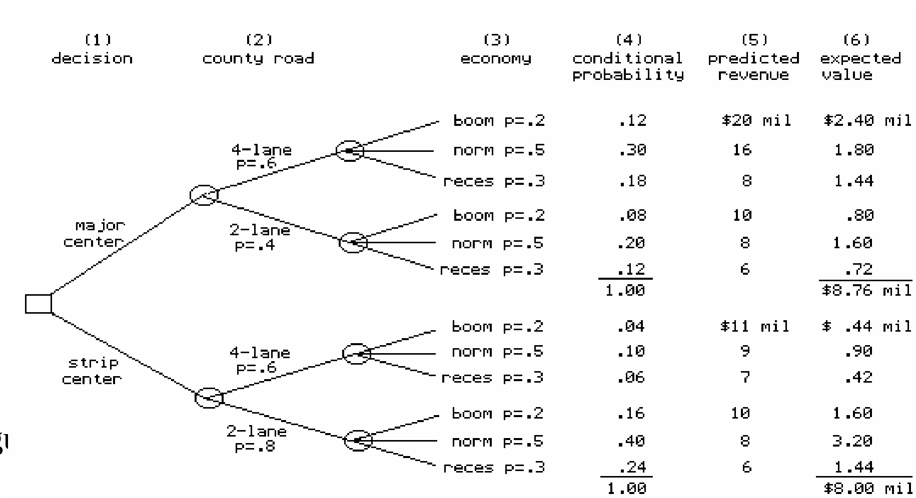
On the other hand, Figure 5 demonstrates the notion of *sub-graphs* (which are also CxGs), and that of *temporal branching* linking the sub-graphs and representing action sequences that can be done in different order.

In the end, the resulting application from the SART project is used as a decision-support system to provide assistance to the operators who are responsible to oversee the occurrences of incidents in the Paris Metro. The use of context in the application allows it to act as a catalyst in solving incidents at different levels. Essentially, the application itself is used by the organization that oversees the Paris Metro, and thanks to the SART project, the CxG formalism was also born.

3.2 The Human Behavior Representation (HBR) Challenge Problem

Problem Overview

The Defense Modeling and Simulation Office (DMSO) of the United States Department of Defense sponsored the HBR Challenge project. Essentially, the main focus is that of enhancing reuse and interoperability of human behavior and performance models.



Fig

Figure-7 demonstrates the need to obtain validated performance data and acquire knowledge through performance moderators, and the ability to represent such data/knowledge and model it using a chosen reasoning paradigm. Each DMSO contractor used their own modeling technique/reasoning scheme (e.g. neural networks, fuzzy logic, Bayesian Networks, value-driven decision tree, CxBR and Case-based Reasoning). UCF's chosen paradigm, is Context-based reasoning (Gonzalez, and Ahlers 1998.) The need to use visualization tools to have access to the various parts of the toolkit is quite obvious; after all, one needs to be aware of the occurrences at each given point in the process. As mentioned previously, DMSO decided to go only with the modeling techniques that apply to a certain type of combat domain, hence why the initial focus was on decision models (Figure 6.)

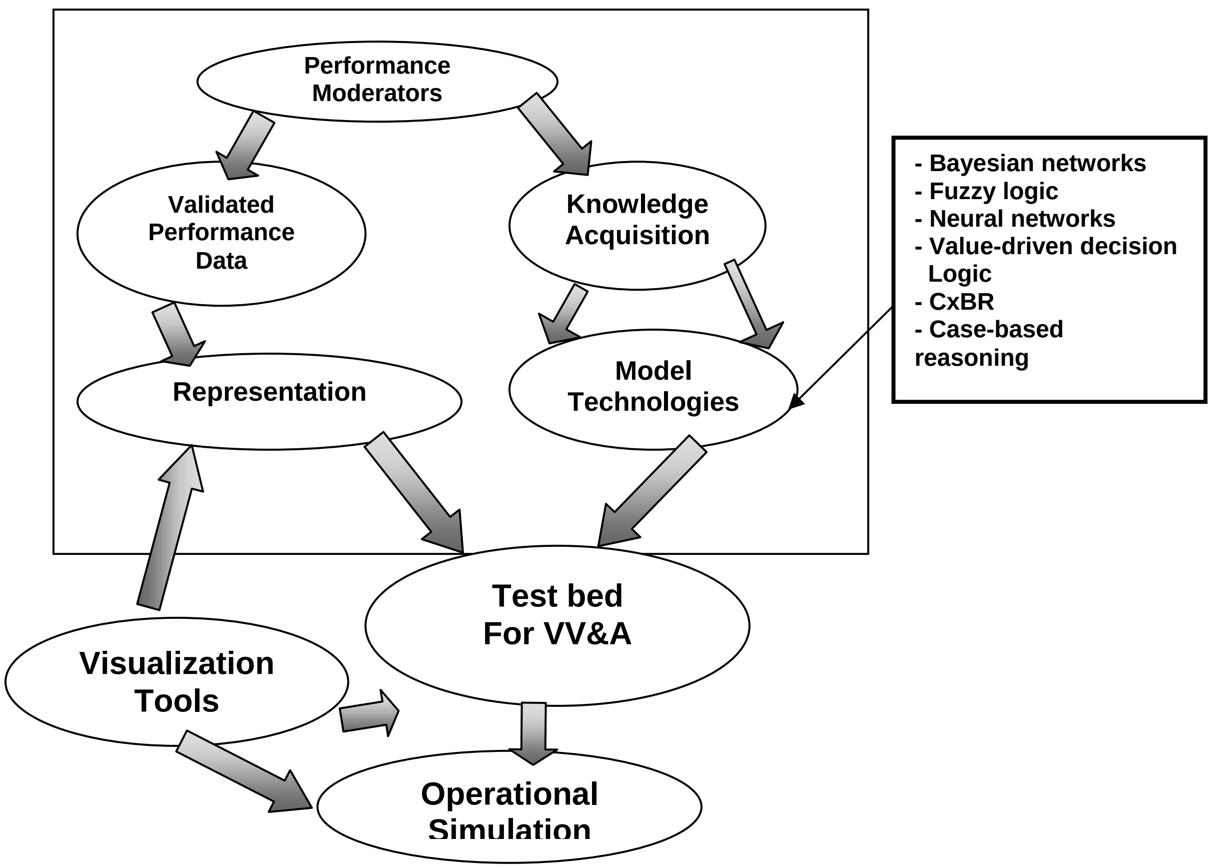


Figure 7: The HBR Toolkit (Spring Simulation Interoperability Workshop, 2001)

In order to understand the project, one has to be aware of some of its important underlying components, which are as follows:

- SPEEDES Simulation Engine/Acquarius Testbed
- VR Forces
- Integration Strategy - SPAWAR
- Project Timeline
- Air, Sea and Ground Vignette Scenarios, and
- MOE’s - Measures of Effectiveness (HBR Model Output Tactical Effectiveness)

3.2.1 Ground Vignette (Tanks) – Challenge Problem

The ground vignette scenario presents the following challenge:

1. Tank platoon (blue) vs. tank platoon (green)
2. Tank modeled as one decision entity vs. the entire crew
3. Blue tanks must exhibit doctrinally correct behaviors
4. Red tanks an scouts will use scripted behaviors
5. Explicit terrain modeling

3.2.2 Air Vignette (Aircraft) – Challenge Problem

The air vignette scenario presents the following challenge:

1. Fighter Combat Air Patrol, beyond visual usage, air-to-air combat.
2. Aircraft modeled as one decision entity (blue) vs. the entire crew (green)
3. Red includes strikers and escorts
4. Blue aircraft must exhibit doctrinally correct behaviors.

5. Red aircraft will used scripted behaviors.

3.2.3 Maritime (submarine) Vignette – Challenge Problem

The maritime vignette scenario presents the following challenge:

1. Submarine intelligence warning operation outside a port: Sub -- Picket Ships -- Carrier Battle Group
2. Red subs attack picket ships
3. Possibly have distracters such as civilian fishing boat

Some other possible scenarios are as follows:

- Stability operations
- Refugee Resettlement
- Disaster Relief
- Military Operations on Urban Terrain

The expectation is that the completed product may be used to rescue detained personnel from the Smart Sensor vignettes (based on a web application.)

3.2.4 Pending Implementation Issues Project Initiation

There were a number of issues that were pending prior to the contractors' attempts to work on this project and they are as follows:

- Finish developing vignettes
- Develop Measures of Effectiveness (MOEs)
- Design experiments

- Integrate decision models into test bed
- Run training vignettes
- Run “withheld” vignettes
- Evaluate results

3.2.5 Basic “TESTBED” Process

The Test bed process involves the establishment of overall goal (which is to evaluate the HBR systems), specific Measures of Performance (MOPs) and MOEs (Measures of Effectiveness), a test plan (Vignettes, **Decision Types** and MOPs), a test environment (SPAWAR test bed), an environment to establish performance, and data collection (SPEEDES writes to data files.) In addition, it involves ensuring that the data collection mechanism is functioning properly, testing of environmental factors, conducting of multiple runs for each factor (Monte Carlo runs), factor results development (spreadsheet import), factor results comparisons, and conduction any additional required tests.

3.2.8 Evaluation Criteria

The evaluation criteria consist of measuring/evaluating the following:

- Vignette Performance
- Execution time for *Decision Types*
- Extensibility
- Memory Requirements
- Usability Considerations (clarity/credibility of HBR system reasoning, HBR knowledge base maintenance)

- Validation and Verification

3.3 HBR Decision Process

The HBR decision resolution process involves how each vignette views other entities. Each vignette is empowered by some basic background information, namely, goal, current status, decision types, possible behaviors/actions, OOB, and non-HBR driven actions or events. The outcome of the vignette's view of the entities depends on the HBR decisions based on a Data Collection Plan that not only defines what data will be saved, but also when and why such data should be saved. Each vignette is executed multiple times and SPEEDES writes data to the MOE/MOP Results file appropriately.

The decision types play an important role in the decision making process because they indicate the different types of human decision-making functions that lead to particular behaviors. Accordingly, each vignette contains several decision types, established by the **DMSO Decision Taxonomy** (Appendix J – Tables C-1 to C-3), which is jointly developed by DMSO, Test Bed and HBR developers, and so are the MOPs/MOEs.

There are certain boundary conditions that constrain each HBR system and they are as follows:

- Engineering issues
- Time
- Process clarity and traceability
- Reasonable cumulative validity of results

- Documentation
- Utility.

As always the time factor is of major importance in a project of this caliber, it involves the time to (1) make a decision given an input, (2) modify for a new vignette, and (3) modify on variation in vignette. In addition, a utility is developed for each individual decision type, across all decision types and across most common decision types.

3.3.1 The HBR Sea Scenario as an example

As per Epsilon Systems Solutions, the following example is a valid description: Consider a situation where heightened tensions are experienced with the red country. Hostilities have not commenced but the blue country has sent a Battle Group to help stabilize the area. A high value asset (HVA) is about 200 Nautical Miles (NM) offshore, and AAW picket ships are approximately 80 - 100 Nautical Miles (NM) offshore. Red has claimed all waters within 100 NM to be territorial, and warned that any warships within that area would be considered hostile and face the potential of attack without notice. A blue sub is conducting Intel and Warning mission by the red port. Red sub(s) have been ordered to get underway. They may be conducting local ops, or have orders to harass or attack blue ship(s). As the red sub(s) prepare to get underway, the blue sub detects indications that units within the port will be getting underway soon.

CHAPTER FOUR: CURRENT APPLICATION ANALYSIS

This chapter explores the analysis illustrated in Table 2a. It provides a general overview of the types of metrics applied when contrasting the two paradigms through a comparison of their main parameters and that of their existing applications. Hence, the SART application, which is the domain engine for the CxGs-based project to help operators in the Paris Metro, is associated with the CxG paradigm. The HBR Challenge problem, on the other hand, is associated with the CxBR paradigm.

Table 1 Metric-based comparison table between CxBR and CxGs.

Similarity Metric	**CxBR**	**CxGs**
Context Representation	Discrete and Fixed, thus, models operate within one context in a given time interval	Acyclic, directed with serial/parallel organization due to the divergence of branches to the Contextual Nodes.
Contextual Transition	Occurs via a Context**-transition pairs**	Occurs in or out of the **Proceduralized Context**
Transition Rule	**Transition Sentinel Rules** are used to handle context transitions (movements)	Represents a transition in operator “practice” via a **Contextual Node**
Time Representation	The amount of **time** spent in a particular context	The amount of **time** spent in the **proceduralized context**
Knowledge Acquisition	Occurs via the CITKA automated knowledge acquisition tool through an interaction involving querying the human expert for an intended autonomous behavior for a particular situation/scenario	Occurs via ‘Practice’ Assimilation through a graphical interaction with a human operator who reports which actions sequence was taken to solve the problem to the system
Application Domain	Used more in an environment that is ‘Autonomous’ in nature such as military mission	Used more for ‘Decision Making’ in a decision-support setting. Environment is integrated at the contextual node level.

As indicated in Tables 2a, the comparison parameters used to compare the applications associated with these two paradigms are as follows:

- Domain
- Environment
- Engine
- Context Representation
- Context Base and Content
- Context Transition Conditions
- Meeting Context Transition Conditions
- Domain Knowledge Evolution
- Constraints
- Context Application Example

Hence, on one hand we have the CxBR paradigm whose framework is used as the engine to handle the HBR challenge problem. On the other hand, we have the CxBR paradigm whose framework (SART) is used to handle the incidents in the Paris Metro. The HBR Challenge domain uses the CxBR engine to collect intelligence information about surface and subsurface naval vessels passing through an inlet while also protecting a High Value Asset that is located out at sea. The Paris Metro domain uses the CxGs paradigm in order to manage/resolve incidents through operators working in the Paris Metro, using the SART system to allow them to decide and manage incidents (e.g. object on the track, lack of power supply) more accurately. As mentioned previously, the CxBR paradigm is typically tied to an autonomous environment whereas the CxG paradigm tends to be associated with an environment that is more decision-support in nature. However, the implication is not that the paradigms cannot be used in the other's natural environments. Indeed, this thesis is an evaluation of the possibility that the latter can be achieved successfully.

Beyond the environment parameter, Table 2a tackles the engine parameter. In the case of CxBR reasoning, the role of the engine (CxBR framework) is to exercise the knowledge represented in the context base. However, in the CxG scenario, the engine is the SART application framework, and its role is to exercise the knowledge that is present in the CxG. The engine is very important as it guides the reasoning process and also facilitates knowledge acquisition, which is what will be later referred to as knowledge refinement. The next comparison parameter is Context Representation. In the CxBR domain, it involves the development of a *Context Base* based of acquired knowledge regarding the autonomous environment to be incorporated within the CxBR Framework to reach an objective. In the CxG domain, it is strictly a representation of the development of 'External Knowledge' based on procedures from a decision-support oriented environment to support a task performing/decision process. Figure 2a illustrates the meaning of external knowledge and how part of it can be contextualized, hence the creation of contextual knowledge pieces. The next comparison parameter is the context base itself. In the CxBR reasoning scenario, it is the CxBR context base, and in that of CxG, it is simply the External Knowledge or the CxG. Another important parameter is that of context-base content. Accordingly, it is only fair that one's curiosity may be guided towards the types of content that each respective paradigm context-base has. Essentially, in the case of CxBR, there are a number of parameters that should be part of one's awareness and they include but are not limited to: Mission Objective, Major Context, Compatible Next-Major-Context and Sub-context. We have already established from the previous chapters that for the CxBR paradigm a context is an autonomous procedure and a Sub-context is a lower-level autonomous procedure. In the case of CxGs, the context base simply contains decision-support procedures, or knowledge-forming decision-support constraints. These contents facilitate the decision-making or task-performing process.

The next parameter involves the conditions that have to hold in order for transitions to occur. In the case of CxBR, Mission Contexts define the autonomous agent's knowledge of an autonomous situation, hence a form of Contextual Knowledge. A set of environmental conditions dictated by the autonomous agent's knowledge form the basis of transition rules for existing context transition pairs. *Sentinel rules* are placed to monitor such transition conditions appropriately. In the case of CxGs, the External Knowledge defines an operator's decision-support knowledge of a decision making or task performing process. The Conditions are based on *Task Focus* or *Event Triggering*, allowing a piece of contextual knowledge (contextualized procedure) to be applied to a particular task. The next parameter considers how the context-transition conditions are met, hence why it is called *meeting context transition conditions*. In the CxBR engine, when a context transition condition is met, the antecedent part of the transition rule allows the rule to fire. Accordingly, its consequent part is activated in order to process the transition from one context to the next. On the other hand, in CxGs, when a task focus demands the use of a particular contextual knowledge, the condition is met for the transition from contextual knowledge to proceduralized knowledge. The next comparison parameter is of utmost importance because it concerns the evolution of the domain knowledge itself. Hence, how does domain knowledge evolve in both paradigms? In the case of CxBR, knowledge evolves through eliciting a domain expert via the knowledge engineer. In the case of CxGs, it simply occurs through an operators' interaction with the system. The last but not least comparison parameter is the *constraints*. In the CxBR domain, they are as follows:

- All Major Contexts should be part of the Mission Context
- Transition Functions and action definitions can be shared among Contexts.
- Sub Contexts can be shared among major contexts.
- Deleting a Main Context eliminates all of its Sub-contexts
- Deleting a Context will eliminate its Action Definitions and Transition Criteria only if they are not used by any other context.

In the CxG domain, the constraints are as follows:

- People transform contextual knowledge into some functional knowledge or causal and consequential reasoning to anticipate result of their own actions
- A need for a consistent, explicative framework to anticipate the results of a decision or action. This consistency is obtained on reasoning about situational

causes and consequences.

- A conscious reasoning about causes and consequences.

In table 2a examples are included in an effort to further substantiate the current application comparison. In these examples, typical applications for CxBR and CxGs are analyzed respectively:

The provided example for CxBR involves defining what the *mission context* is, and in this case it is the collection of intelligence without detection and protect HVA from hostile contacts. It also defines the notion of a *major context*, which in this case is intelligence gathering about the autonomous environment. In addition, functions were defined including their recommended decision functionality (Taxonomy A - HBR decision taxonomy table.) In this example the following functions were used:

- *Function #1(Alert/Detect)*: Seek appropriate location. Determine safest position to avoid detection and accomplish mission context.

- *Function #2 (Identify)*: Define depth. Set depth.

- *Function #3 (Activate)*: Run quiet. Turn off engines.

- *Function #4 (Perceive)*: Monitor. Continually check all sensor equipment for contacts and contact information.

In addition the respective sub-contexts were as follows (Table 2a):

- *Sub-Contexts1 (Adapt)*: **avoid-floor -** If the sea floor gets to within a certain range or the sub approaches its maximum depth, steer the sub to a more shallow depth.
- *Sub-Contexts2 (Adapt)*: Steer the sub away from a target if a target approaches within a certain range.

On the other hand, the example for CxGs involves the following parameters (Table 2a):

- **Incident Resolution Procedure**: Resolving a Lack-of-Power issue for a Train in the Paris Metro.

- **Contextual Graph**: Lack of power incident resolution
- **Sub Graph**: **Train Aid -** Elementary or atomic task of helping a disabled train due to lack of power.
- *Train Aid's Sub procedure1*: **Damaged train emptying** Elementary or atomic task of emptying a disabled train due to lack of power.
- *Train Aid's Sub procedure2*: **Helping Train to Empty**. This is derived from the first sub procedure and adapted by the fact that an available train may run to the next station and evaluate its travelers in better conditions if the station is free.

Table 2a Cross Comparison Table between HBR Challenge (CxBR-based) and Paris-Metro (CxG-based)

	Project – Corresponding Modeling Technique	
Cross-Application Parameters	**HBR Challenge - CxBR**	**Paris Metro (SART) - CxGs**
Domain	Intelligence collection about surface and submarine naval vessels passing through an inlet while also protecting a High Value Asset located within hostile seas.	Incident management/solving by operators working in the Paris Metro, to allow them to decide and manage incidents (e.g. objects on the track, lack of power) more accurately.
Environment	Autonomy	Decision-support
Engine	CxBR Framework Exercises the knowledge represented in the 'Context Base'	SART Application Framework Exercises the knowledge represented in the CxG.
Context Representation	Development of a 'Context Base' based on acquired knowledge regarding the autonomous environment to be incorporated within the CxBR Framework to reach an objective.	Development of 'External Knowledge' based on procedures from decision-support driven environment to support a task performing/decision process

Context Base	Consists of a set of contexts organized hierarchically	Consists of actions and contextual nodes organized as a CxG.
Context Base Content	Mission Objective, Major Context, Compatible-Next-Major-Context, Compatible-Sub-Context.	Decision support procedures or constraints forming the knowledge necessary to make decisions or perform tasks – when a procedure is contextualized in a particular context, it becomes a 'Contextual Knowledge'
Context Transition Conditions	A set of environmental conditions dictated by the autonomous agent's knowledge forms the basis of transition rules for existing context transition pairs. 'Sentinel rules' monitor such transition conditions appropriately	External Knowledge defines human decision-support knowledge of a decision making or task performing process. The Conditions are based on Task Focus or Event Triggering, allowing a piece of contextual knowledge (contextualized procedure) to be applied to a particular task.
Meeting Context Transition Conditions	When a context transition condition is met, the transition rule fires. Accordingly, its transition from one context to the next is executed.	When a task focus demands the use of a particular contextual knowledge, the condition is met for the transition from contextual knowledge to proceduralized knowledge
Domain Knowledge Refinement	Occurs by eliciting knowledge from a domain Expert (a.k.a., Subject matter expert) via the knowledge engineer.	Occurs through an organization's Operators' interaction with the CxG system

Constraints	- All Major Contexts should be part of the Mission Context - Deleting a Main Context does not eliminate all of its Sub-contexts - Deleting a context will eliminate its Action Definitions and Transition Criteria only if they are not used by any other context.	- A need for a consistent, explicative framework to anticipate the **results** of a decision or action. This consistency is obtained by reasoning about situational **causes** and **consequences**. - A need to transform contextual knowledge into some functional knowledge or **causal** and **consequential** reasoning to anticipate result of their own **actions** - A necessity to have conscious reasoning about causes and consequences.

Application Example	**Mission Context**: Collect intelligence without detection and protect HVA from hostile contacts. **Major Context:** Intelligence gathering *Function #1(Alert/Detect*)*: Seek appropriate location. Determine safest position to avoid detection and accomplish mission. *Function #2(Identify)*: Define depth. Set depth. *Function #3(Activate*)*: Run quiet. Turn off engines. *Function #4(Perceive*)*: Monitor. Continually check all sensor equipment for contacts and contact information. *Sub-Contexts1(Adapt*)*: **avoid-floor,** *Sub-Contexts2(Adapt*)*: **avoid-other-vehicles** **avoid-floor:** If the sea floor gets to within a certain range or the sub approaches its maximum depth, steer the sub to a more shallow depth. **avoid-other-vessels:** Steer the sub away from a target if a target approaches within a certain range. * See HBR Challenge Tables.	**Incident Resolution Procedure**: Resolving a Lack-of-Power issue for a Train in the Paris Metro. **Contextual Graph****: Lack of power incident resolution Sub-graph link**: Train Aid** **Sub Graph***: **Train Aid** Elementary or atomic task of helping a disabled train due to lack of power. *Train Aid's Sub procedure1*: **Damaged train emptying** Elementary or atomic task of emptying a disabled train due to lack of power. *Train Aid's Sub procedure2*: **Helping Train to Empty**. This is derived from the first sub procedure and adapted by the fact that an available train may run to the next station and evaluate its travelers in better conditions if the station is free. * Elementary chunks of reasoning stored and reminded to the operators in case of an incident. Can be reused or adapted for other actions. ** Contextual information matters more than probabilities with decision making in a decision-support setting.

In the introductory chapter, several metrics were used (as provided in Table 1) to provide specific, yet in-depth comparisons among the relevant aspects of the CxBR and CxG formalisms. There is relevance in adding substance to this chapter by analyzing these metrics especially as a preparation for the next chapter, which furnishes us with a cross-application analysis for the two formalisms. This type of analysis is important since it attempts to explore the potential that the formalisms and their applications can be used interchangeably.

- The first metric provided in Table 1 is 'Context Representation'. In the CxBR paradigm, there is a set of discrete contexts occupying intervals of time in a fixed hierarchy of contexts, where a context is a module of knowledge applicable to a particular situation. On the other hand, in CxGs, there are two types of contexts. First, there is a static context, such as the context of A3, and thus as many discrete contexts as items. Second, there is a dynamic context, such as the context of a practice, which evolves along the practice use. For instance, in CxBR, if we have a context named 'intelGathering', it will contain all knowledge necessary for an agent to operate about marine and submarine vehicles passing through an inlet while also protecting a High Value Asset that is located out at sea, as provided earlier in the example of a sea vignette specification. Thus, the knowledge (e.g. location, depth, contacts, contact information, etc.) is exclusive to that context for the period of time that the naval vessel is being operated. In CxGs, one distinguishes the static context 'intelGathering', which is part of the context of operating a vehicle underwater, and the dynamic context of the command process executed by the submarine operator, which accounts for the specificity of the situation (e.g. avoid floor near at the location where the surrounding water is near the Red port.)

- The second metric is 'Application Domain: parallel between CxBR's autonomous knowledge and CxG's (Operator) Practices'. The application domain of CxBR is one that is autonomous in nature, thus the results are related to the environment (e.g. stimuli received by the intelligent agent, see figure 2) in nature and are beyond one's control. Whereas, in CxGs, the environment is integrated in the reasoning process at the level of instantiated contextual nodes along the path of a practice application (e.g. an operator may provide a sequence of actions on a corresponding path like the following: {A1, A2, A3, A5, and A9}. The latter on this distinct path can be intertwined with contextual nodes or node values C1.1, C2.1, C3.1 (emanating from contextual nodes C1, C2, and C3), and recombination nodes R3, R2, and R1 to be more clearly represented as follows: {A1, A2, C1.1, C2.1, C3.1, A3, R3, A5, R2, R1, A9})). Accordingly, in CxGs, the environment is considered at the same level as the actions. Thus, it can be argued that the contextual nodes relate to autonomous knowledge, in a manner that is similar to the link between sentinel rules and autonomous knowledge in CxBR. However, CxGs are not used in autonomous applications because the environment cannot dictate the contextual element(s) selected.
- The third metric is 'Context Movement and/or Contextual Change'. This metric introduces the notion of a 'unit of movement', referred here as 'context

movement' to serve as the basis for dealing with context dynamics in CxBR and CxGs. In CxGs, this type of movement can be characterized by the movement of a contextual element into or out of the proceduralized context, namely the execution of a practice. In CxBR, it is characterized by the transition between two contexts as indicated by the respective context-transition pair. In the two formalisms, 'context movement' occurs via an encounter with a transition rule. In CxGs, a transition rule would correspond to a contextual node (the first part of the contextual element). The actual transition occurs at the transition node. In the contextual node, a choice has to be made according to an external value. The contextual element is then instantiated, and thus enters the proceduralized context. Whereas, in CxBR, the transition rules called 'Transition Sentinel Rules' are used to handle the context movement for a given set of discrete contexts. These rules establish the condition under which a context transition is triggered. Figure 1 can thus be used to illustrate that if a mission provides a context-transition pair, say C1 to C3, C1 will contain a 'sentinel rule' to guarantee that once the transition criteria are encountered, the appropriate context transition will occur. In other words, once these conditions exist, the sentinel rule will fire and a context transition will be executed. The latter is similar to the instantiation of a contextual element in CxGs before it enters the proceduralized context. It is essentially the instantiation of a contextual element occurring only between a contextual node and the corresponding recombination node. Thus, when a piece of contextual knowledge becomes instantiated at a contextual node, it then enters the proceduralized context. Accordingly, using the above paragraph, we can deduce that a sentinel rule in CxBR is similar to a contextual node in CxGs.

- The fourth metric is *time representation*. In CxBR, time is represented indirectly through externally occurring events based on stimuli received by the intelligent agent from the environment; such stimuli may trigger change in contexts where each context occupies a set time interval. On the other hand, in CxG, time is represented indirectly via the introduction or removal of a contextual element into (or from) the proceduralized context, essentially a de-instantiation of that contextual element. It is good to remember that the construction of a proceduralized context involves the use of contextual knowledge from possibly various domains. However, when it comes to the representation of time, in CxBR it is based on the length of time that is spent into a particular context, while in CxG, the length of time spent into the proceduralized context.
- The fifth metric is *Knowledge Acquisition*. In the CxG formalism, a representation is capable of evolving through the accommodation and assimilation of practices via an operator reporting a problem solution to the system in terms of the sequence actions that was taken through his decision-making process. This type of acquisition corresponds to the addition of the minimum number of contextual elements (contextual node-recombination node pair and an action) and since the CxG's movement from external knowledge to contextual knowledge goes through its use in a proceduralized context, it is possible to face the problem of a context of infinite dimension.

In Context Dynamic and Contextual Graphs (Brézillon, 2003b), the development of a CxG-based system that exploits the CxGs formalism is mentioned. This system was implemented as a prototype using Java, where all of its data are stored in a database. An important part of this CxG-based system is the identification of a sequence of actions used by the operator for a problem solving session. This is accomplished via the interaction of human operator with the system through a *graphical presentation* of the current state of the CxG. Once the problem solution is reached, the human operator first reports the action sequence taken to solve the problem to the system, and then the entered sequence of actions. Then, the operator has to decide to report to the system, which one is the closest, and the entered sequence may or may not be known. If the system determines a discrepancy, it demands the reason for such a discrepancy from the operator, who then proceeds to provide the system with, the missing contextual element, its location (position of the contextual and recombination nodes), and its *known practice* and *entered practice* instantiations.

The system matches actions of the sequences in an ordered way. Hence, a discrepancy is detected when an action is different, new, or missing between the two sequences. This information is what allows the respective CxG to provide a uniform representation of actions series and contextual elements, which explain the reason of a choice of an action over another. Accordingly, the generation of explanations becomes quite seamless because the knowledge upon which the explanation relies is explicit in the explanation. It is at that point that the system asks the operator for the reason of the difference. The operator proceeds to provide the system with the missing contextual element (definition), its location (position of the contextual and recombination nodes on the path), and its instantiation for the known practice and the entered practice (Brézillon, 2003a).

Thus, such a structure allows the system to determine any discrepancy when an operator inputs a sequence of actions since it expects any corresponding contextual elements to be provided to determine discrepancies, or choice of an action over another. In CxBR, Gonzalez et al developed an advancement of the CxBR formalism in the form of an automated knowledge acquisition tool named CITKA. CITKA creates CxBR model specifications through querying an expert in the intended behavior. An important similarity can be established between CITKA and Brézillon's Java-based CxG tool in that they both query someone (e.g. expert, operator) in order to arrive to the expected result.

Although many of the projects that have utilized CxBR models in the past few years involved autonomous military simulations funded by organizations such as the US Army, and the Department of Defense to produce autonomous platforms (e.g. unmanned submarines, tank platoons, aircrafts and others). This is a capability that also requires that such systems be able to determine discrepancies. However, unlike the CxG application, these applications require that the respective knowledge of the intelligent autonomous agents is autonomous in nature so that they will be able to act not only intelligently, but also realistically in light of a trainee's action (Brézillon, 2003). Ferlund's work, which uses a genetic programming approach to produce and evolve entire CxBR models, is a great example of such enhancements. In short, such enhancements in the decision-making process of both formalisms will allow them to have more effective decision-making stage, thus allowing them to be more reliable in determining discrepancies in the process of acquiring knowledge.

CHAPTER FIVE: CROSS-APPLICATION COMPARISON ANALYSIS

Table 2b provides a general overview of the types of comparison parameters that are applied in an attempt to successfully cross-compare the two paradigms by interchanging their existing applications. Accordingly, the SART application, which is typically associated within the CxG paradigm, is now analyzed with the CBR paradigm. Likewise, the HBR challenge application, which was previously analyzed within the CxBR paradigm, is now analyzed within the CxG paradigm. Table 2b is very similar to table 2a, except the applications are crossed. The objective is to simplify this table and avoid unnecessary repetition by only specifying the parameters that have changed from their previous definitions in Table 2a.

Table 2b – Cross Comparison Table with modeling techniques switched around

	Project – Corresponding Modeling Technique	
Cross-Comparison Parameters	**Paris Metro (SART) - CxBR**	**HBR Challenge - CxGs**
Domain	CxBR intelligence collection enhanced by knowledge acquisition via operators in order to successfully manage/solve incidents (e.g. object on the track, lack of power supply) in the Paris Metro.	Using the CxG-based SART system to help operators decide on what type of intelligence to collect from submarine vehicles passing through an inlet while also protecting a HVA that is located about 200 nautical miles out of the sea.
...	...	...

Cross Application Example	**Mission Context**: Collect intelligence regarding resolving a lack of power issue for a train in the Paris Metro line… **Major Context:** Intelligence gathering about lack of power on a train line in the Paris Metro. *Function #1(Alert/Detect*)*: Seek appropriate location. Determine safest way to approach a recommended resolution for the issue in question. *Function #2(Identify*)*: Define Danger Level (0-low, 5- high) *Function #3(Identify*)*: Define Appropriate Warning Messages. *Function #4(Assess/Activate*)*: Monitor all sensor equipment for contacts and contact information. *Sub-Context1(Adapt*)*: **unload-train,** *Sub-Context2(Adapt)*: **help-train,** **unload-train:** If functions 1-4 have been activated, safely unload the train and ensure that the 'help-train' sub-context is or has been triggered. **help-train:** Alert mechanics of possible symptoms/diagnostics to prepare them to resolve the power malfunction, and then set Function 2 to zero (0) in order to clear the train in question and allow it to move by delaying other trains. * See HBR Challenge Tables	**Incident Resolution Procedure**: Resolving an issue with a detected hostile contact via the help of operators and also protecting HVAs from such contacts. **Contextual Graph****: Lack of detection of hostile contacts. Sub-graph link**: Detect Hostile Contacts…Train Aid** **Sub Graph***: **Detect Hostile Contacts or Avoid Detection** Elementary or atomic task of brute-forcing the SART system to detect a hostile contact. *DHC's Sub procedure1*: **Adding Contextual Knowledge** Elementary or atomic task of adding more contextual knowledge to catalyze detection *DHC's Sub procedure2*: **Warn about distance from floor and other vehicles**… This is derived from the first sub procedure and adapted by the fact that an available train may run to the next station and evaluate its travelers in better conditions if the station is free. * Elementary chunks of reasoning stored and reminded to the operators in case of an incident. Can be reused or adapted for other actions. ** Contextual information matters more than probabilities with decision making in an decision-support setting.

As indicated in Tables 2a, and 2b, the cross-comparison parameters are as follows:

- Domain
- Environment
- Engine
- Context Representation
- Context Base and Content
- Context Transition Conditions
- Meeting Context Transition Conditions
- Domain Knowledge Evolution
- Constraints
- Context Application Example

In the CxBR cross-application part, the domain is now the Paris Metro, and in the CxG part, it is now the HBR Challenge. Hence, the assumption in this instance is that the Paris Metro domain uses the CxBR paradigm successfully manage/solve incidents (e.g. object on the track, lack of power supply) in the Paris Metro. The HBR Challenge domain uses the CxG-based SART system to help operators decide on what type of intelligence to collect from enemy submarines. The decision to allow CxBR and CxG to remain tied to their typical application environment namely autonomous and decision-support , and simply switch their typical respective applications around for the cross-comparison is based on the fact that the respective modeling techniques are designed with such environments in mind. However, this is simply a convention adopted to simplify the analysis itself, since this thesis hypothesizes they could be used interchangeably with all the right considerations.

In Table 2b, a cross application table was included in an effort to further substantiate the provided cross-comparison. In these examples, a typical application domain for CxG was implemented in CxBR, while a typical application domain for CxBR was implemented in CxGs. The Example for CxBR involves defining what the major context is, and in this case it is the collection of intelligence regarding resolving a lack of power issue for a train in the Paris Metro line. It also defines the notion of a *main context*, which in this case is intelligence gathering about lack of power on a train line in the Paris Metro. In addition, functions were defined including their recommended decision functionality (Taxonomy A - HBR decision taxonomy table.) In this example the following model was used:

- *Function #1 (Alert/Detect*)*: Seek appropriate location. Determine safest way to approach a recommended resolution for the issue in question.
- *Function #2 (Identify*)*: Define Danger Level (0-low, 5- high)
- *Function #3 (Identify*)*: Define Appropriate Warning Messages.
- *Function #4 (Perceive/Activate*)*: Monitor all sensor equipment for contacts and contact information.

In addition, the respective sub-contexts were as follows:

- *Sub-Context1 (Adapt*)*: **unload-train -** If functions 1-4 have been activated, safely unload the train and ensure that the 'help-train' sub-context is or has been triggered.

- *Sub-Context2 (Adapt)*: **help-train -** Alert mechanics of possible symptoms/diagnostics to prepare them to resolve the power malfunction, and then set Function 2 to zero (0) in order to clear the train in question and allow it to move by delaying other trains.

On the other hand, the example for CxGs involves the following parameters.

- **Incident Resolution Procedure**: Resolving an issue with a detected hostile contact via the help of operators and also protecting HVAs from such contacts.
- **Contextual Graph**: Lack of detection of hostile contacts.
- Sub-graph link**: Detect Hostile Contacts**
- **Sub Graph**: **Detect Hostile Contacts or Avoid Detection -** elementary or atomic task of brute-forcing the SART system to detect a hostile contact.
- *DHC's Sub procedure1*: **Adding Contextual Knowledge -** elementary or atomic task of adding more contextual knowledge to catalyze detection
- *DHC's Sub procedure2*: **Warn about distance from floor and other vehicles**…

As mentioned previously, a sub-graph is an elementary chunk of reasoning stored and reminded to the operators in case of an incident. It can be reused or adapted for other actions.

5.1 Evaluation

These examples provide a general view of how the two formalisms can be used interchangeably. However, both formalisms offer their respective advantages and disadvantages. In the case of CxBR, DMSO noted that contextual information matters more than probabilities with decision making in a decision-support setting. On the hand, in CxGs, an activity as simple as seeking an appropriate location in order to move a submarine to it may be viewed as a very simple activity, yet it really presents various possibilities that can be viewed as sub-graphs. A possibility of a sub-graph could be to avoid the floor or other vehicles, or to avoid detection. Yet another possibility may be that of making noise to indicate presence. Nonetheless, there’s one thing that doesn’t change in any of those activities, essentially, the idea that, there’s a starting point (origin) and a potential endpoint (destination.) Accordingly, if the origin changes in one of the possibilities, it’ll also have to change in all the other possibilities. Hence, the origin and destination parameters can be viewed as an *activity*, which is a recurring structure observed in different CxGs, where each possibility can be viewed as a sub-graph and thus a CxG.

Table 3: Port Activity Reporting Criteria (DMSO, 2001)**)**

Activity #	**Activity**	**Probability of Sortie Within 2 Hours**
1	No ESM/COMMS above baseline; no reporting required	1%
2	Sporadic ESM/COMMS above baseline levels; reporting required	50%
3	Continuous ESM/COMMS above baseline levels; reporting required	95%

When a submarine’s sensor indicates to its driver that there’s traffic ahead, he’ll quickly think of the right ‘Action’ to take in that situation. The ‘Action’ is directly related to the ‘contextual knowledge’ regarding the current situation, and when applied to the current focus (the task at hand), it can be proceduralized to the next option. Essentially, this next action is taking an alternate route in order to bypass the traffic.

Table 4: Representative Asset Values (DMSO, 2001**)**

Asset	**Value**
HVA	10
Blue sub	10
Picket ship	4
Red sub kill under proper ROE	3
On time reporting	2

Accordingly, this procedure can be re-applied whenever this situation presents itself again. As mentioned earlier, when a procedure is applied in a particular context in CxG, it is called a *practice*, which itself is a sequence of Actions. It is very possible that one may discover a possibility that was not previously part of the existing current knowledge. However, once it is discovered, it becomes a contextual knowledge, which can be proceduralized when applied to a particular task. This transition occurs at the Contextual Node as demonstrated in the previous chapters.

In the instance of using the decision-support example into the CxBR framework, one can view the operator of the train as a pseudo entity of control. Thus, instead of having an autonomous agent, we have a human entity whose knowledge of the situation at hand is the Major Contexts. The mission requirements are stated in terms of environmental conditions that can be embedded into sentinel rules. These can be viewed as watchdogs that are monitoring the environmental conditions so that when satisfied, the rule fires, thereby warranting a certain action. The notion of a subject matter expert was not mentioned in that example. However it was implied since directional procedures have to come from an SME. Nonetheless, the pseudo controlling entity, in this case the driver will become an expert as he or she encounters more domain knowledge.

If a knowledge acquisition tool like CITKA is used, the KE will be able to simply focus on tweaking the system as opposed to adding the new Contexts and their Corresponding Context Transition pairs from scratch. The CxBR cross example also demonstrated that the controlling entity's behavior, in this case, in the submarine example presented in table 2b (as a snippet of all the other provided possibilities) the submarine driver would be controlled by the context that is active at the time. In addition, only a limited number of things can happen in any single context for a given time interval as the passenger goes through transitions (time intervals) to reach the ultimate goal of reaching the destination.

In the CxG cross example provided in table 2b, the goal is to resolve an issue with a detected hostile contact via the help of operators and also protecting HVAs from such contacts (as required by the HBR challenge project.) In such a case the operator still has to adapt a procedure for overcoming a decision making hurdle in the context in which the hurdle is presented. In the CxBR cross example, the goal is to utilize collected intelligence to define danger level or appropriate warning messages due to lack of power for a train line in the Paris Metro. These examples are provided with the hope of illustrating that it is possible to use the CxG paradigm with an application that was previously used with CxBR and vice versa.

The goal of this thesis is, essentially, that of arriving to a viable cross application comparison of the two formalisms. This was done using the cross-application comparison parameters provided in the analysis. Accordingly, it commenced with an investigation of CxBR and CxGs to provide a clear understanding to the reader as to their domains, implementation and environmental applications.

The information provided in those four previous chapters to arrive to the cross-comparison provided in Chapter 5 is based on every piece of datum that was compiled to arrive to a comprehension of what these formalisms really do individually in other to effectively compare them and ultimately cross-compare them. It started with an understanding that, as a human behavior representation paradigm, a CxBR uses context as the basis of the representation. It was important to realize that typically CxBR models are used to control autonomous agents in the performance of an autonomous mission. On the other hand, CxGs, represented as an acyclic directed graph with a unique input, a unique output, and a serial/parallel organization of nodes connected by oriented arcs, is typically used for decision-support processes while considering the environment in question.

This analysis provides us with some relevant information on how the formalisms work individually, allowing us to get insights and prepare information on how to effectively cross-compare them even prior to doing the cross-comparison itself. In the case of CxBR, the literature indicated the basics elements for implementation involves the creation of a CxBR model. In order to develop such a model, one must clearly identify the Mission Context, Subject Matter Expert (SME), the procedures required for controlling a simulated entity, the Context Transition Rules, the mission Objects and the specification and definition of their capabilities, and last (but not least) the identification, specification and definition of "Helping functions"; where examples of Helping Functions are that of finding the distance between two points"(Gonzalez, Gerber, Castro, 2002). Hence, the HBR challenge project was presented as an example of an application where CxBR models were used in an autonomous environment. In the case of CxGs, the SART system represents a CxG implementation as indicated by Brézillon and Pomerol (2002). These two domain applications provided a foundation to validate the use of the comparison and cross-comparison parameters used, namely, domain, environment, engine, context representation, context base and content, context transition conditions, meeting Context transition conditions, domain knowledge evolution, constraints, and context application examples. Table 5 presents a terminology-mapping matrix to further aid in cross-referencing key terms from both paradigms.

Table 5 – CxBR and CxGs Terminology Mapping Matrix

Mapping Metric	**CxBR**	**CxGs**	**Similarity Level**
Application Domain	Autonomous in nature	Decision Support in Nature	Different
Context Representation	Context Base (hierarchical contexts set)	External knowledge (actions and contextual nodes in a CxG)	Somewhat similar

Context Transition	Context-transition-pair	Contextual Knowledge - Proceduralized Context	Similar
Transition execution	Rule fires	Contextual Knowledge-becomes- Proceduralized knowledge	Very Similar
Transition Rule	Sentinel Rules	Contextual Node	Very Similar
Time Representation	Time-spent-in-context	Time-spent-in-proceduralized -context	Very similar
Engine	CxBR Framework	SART Application Framework	Somewhat similar
Sub-structure	Sub context	Sub graph	Same concept, different function

5.2 SART in CxBR

As previously discussed, SART is currently used with the CxGs modeling technique in order provide decision support to the Paris Metro operators. However, this section argues that it is also possible to use the SART project with the CxBR framework as a measure of minimizing decision support, thus optimizing autonomy, particularly in situations that are hazardous to human presence. For instance, if CxBR is used as the modeling technique in the Paris Metro scenario, the domain would become that of a CxBR-based intelligence collection enhanced by knowledge acquisition via operators in order to successfully manage/solve incidents (e.g. object on the track, lack of power supply). As demonstrated in Table 2b, aligning the CxBR framework with this project may minimize difficulties associated with speed and accuracy of decision by having the system take over routine decisions that don't require any human interactions; hence optimizing autonomy. Accordingly, on one hand, the Mission Context could be that of collecting intelligence regarding resolving a lack of power issue for a train in the Paris Metro. On the other hand, the Major Context could be that of gathering intelligence about that same issue. The system integrator or knowledge engineer could proceed to use functions such as:

- "Alert/Detect"- To seek appropriate location. Determine safest way to approach a recommended resolution for an issue in question.
- "Danger Identification" - Define Danger Level associated with executing a resolution (0- low, 5- high)
- "Warning Message Identification" - Define appropriate warning messages, particularly to alert operators when switching from an autonomous mode to decision support mode.
- "Sensor Equipment Monitoring" - To help system monitor sensors and provide alerts in the event of a hazardous situations.

The CxBR system's sub-contexts could also be coded with functions like:

- "Unload Train" - If functions 1-4 have been activated, safely unload the train and ensure that the ‘help-train’ sub-context is or has been triggered.
- "Help-train" - Alert mechanics of possible symptoms/diagnostics to prepare them to resolve the power malfunction, and then set Function 2 to zero (0) in order to clear the train in question and allow it to move by delaying other trains

The only surmountable hurdle would be to code a mode switching function that will allow the system to switch from a semi-autonomous mode to a fully decision support mode in situations that require total human control. In this case, the developer may simply integrate both systems so that the mode switching function would toggle from the autonomous mode to the decision-support mode and vice versa when necessary.

5.3 HBR Challenge in CxGs

The HBR Challenge project is a very good example of a project where an autonomy-driven paradigm like CxBR is well suited (particularly for situations that are hazardous to human presence.) However, there are situations that make it imperative to have human presence at various points in a mission. In such situations, CxGs can be used to provide decision support to the involved human operators. In the case of the HBR challenge, speed and accuracy are equally important whether the decision is being made autonomously by the system, or by a human being with the help of CxG-based decision support system.

The hurdle that would be hard to surmount is that of incorporating autonomy in CxGs for cases where it is simply illogical to have a human entity operate the system due to both speed and accuracy. My recommendation to overcome this hurdle is the use of CxBR as a back-end to the overall system so that it would be allowed to switch from an autonomous mode to a decision support mode and vice versa. This recommendation is based on the fact that it would be harder to create an autonomous CxG co-engine that would work hand-in-hand with the existing CxGs framework, which is systematically designed to work as a decision support system. Table 2b provided a potential scenario of an incident resolution procedure using HBR Challenge in CxGs where the goal is to resolve an issue with a detected hostile contact via the help of operators and also protecting HVAs from such contacts. In this instance, the CxG could represent the detection or lack of detection of hostile contacts. A potential sub-graph could be responsible to either detect hostile contacts or avoid detection. This is where it would be more feasible to use the same kind of mode switching function, but this time to switch from a decision support mode to an autonomous mode unless one would prefer to add functions to the CxGs framework that would allow it to brute force itself to detect hostile contacts with speed and accuracy. I feel that would be like re-inventing the wheel and could potentially be fruitless. Potential Sub-graph procedures are that of "adding contextual knowledge" which could be done in the decision support mode by operators after reviewing a recorded log of the occurrences. Another potential procedure is that of providing warning about distance from the ground or from other vehicles. This section argues that it is more practical to use a pre-existing autonomous engine like that of CxBR in CxGs than to attempt to incorporate such a feature in it as a new feature that would enable it to switch from a decision support mode to an autonomous one as required by a particular mission.

CHAPTER SIX: CONCLUSION, SUMMARY & FUTURE RESEARCH

The goal of this thesis was two-fold: Investigate both CxBR and CxGs as two leading context-based formalisms and discover the idiosyncrasies of their representation, implementation, and application via thorough literature reviews supporting their individual implementation in their typical domains. It also aimed to arrive to a viable cross-application comparison of the two formalisms via the use of the aforementioned comparison and/or cross-comparison parameters as indicated in chapters 4 and 5.

In section 1.3 the following question is posed: *Why should there be an effort to compare CxBR and CxGs*? Indeed, the answer that was provided then is even more valid after arriving at the cross-comparison itself as demonstrated in Chapter 5. Essentially, the goal is to provide an effective comparison of the two modelling formalisms. This provides an unfamiliar reader with an opportunity not only to discover the essence of two well-established context-based formalisms, but also understand their respective applications representing/modelling contexts. Although, each of the two formalisms has its own strengths based on its current area of application (as represented in Chapter 4), Chapter 5 presents a cross-comparison attempt that will allow future researcher to readily get exposed to different aspects of the two formalisms, particularly how their features can either be combined for synergistic use or appropriately exchanged for independent use. The literature review provided in Chapter 2 on both formalisms presents current research endeavours. However, as the years go by other researchers will embark on other comparable yet more advanced endeavours, and this thesis can serve as a way of quickly investigating how these two formalisms can help complement their research endeavours. The latter is facilitated by an exposure to this thesis because even a newly exposed researcher will be empowered with exact information on the use of both formalisms, and including an exploration of how two of their applications can be interchanged as provided in chapter 5's cross-application comparison.

The thesis presents chapter 1 as an introduction to the representations of both formalisms, while Chapter 2 presents a literature review that provides summaries of the CxBR and CxGs literature in an effort to provide more insights to the reader about both modeling formalisms. It not only effectively reviews their individual approach in addressing contexts and their representation through computational paradigms, but also presents new research endeavors (e.g. Computer Generated Forces (CGFs) and Genetic Programming in CxBR). In addition, chapter 3 discusses the reader with problem definitions in terms of current projects, namely the Paris Metro SART project and HBR Challenge Project. Subsequently, Chapter 4 solidifies chapter 3 by providing the reader with a current application analysis. This analysis is an exploration of the illustration provided in Table 2a in order to provide a general overview of the relevant types of comparison parameters. These parameters are applied in an attempt to successfully compare the two paradigms through a comparison of their main parameters and that of their current applications. Accordingly, the SART application, which is the framework for the CxGs-based project to help operators in the Paris Metro, is associated with the CxG paradigm, and the HBR challenge problem is associated with the CxBR paradigm appropriately. On the other hand, chapter 5, essentially, provides the desired cross-application analysis in an attempt to successfully cross-compare the two paradigms by interchanging their existing applications. Accordingly, the SART application, which is typically associated within the CxG paradigm, is now analyzed with the CBR paradigm. Likewise, the HBR challenge application, which was previously analyzed within the CxBR paradigm, is now analyzed within the CxG paradigm.

Chapter 5 also introduces a particular emphasis on the current application of the two formalisms as to arrive to conclusive insights to provide suggestions on possible synergistic or interchanged uses. Indeed, CxBR was conceived to provide reasoning in an autonomous environment (e.g. HBR Challenge project, automated pilots). On the other hand, the CxGs formalism was developed to work in systems that provide decision support (e.g. SART-Paris-Metro.) It is good to remember the importance of autonomous systems particularly when human presence is either not advisable because of hazards or simply too costly (e.g. space exploration). On the other hand, decision-support systems are important when human-decision making needs to be aided or supported to promote accuracy or consistency (e.g. medical expert systems). However, what about situations that require the use of both autonomous and decision support systems? For instance, even when complete use of an autonomous system is possible, it may be advantageous to have a subject matter expert (SME) who is present and ready to interfere in the event of an unusual occurrence in the course of a crucial non-hazardous mission. This SME can also be regarded as knowledge engineer because his or her job would be to consult or interact with a decision-support system in order to confirm each crucial decision point in the incident-solving scenario or add newly encountered issues into the system. The SME would not interfere with the operation/mission unless he/she is alerted by the autonomous system to the occurrence of an unusual event that may not be part of the existing context base. The applications that were analyzed in the previous chapters can be used to illustrate this conclusion. For instance, a train in the Paris Metro may be equipped with a CxBR-enabled autonomous system to control it without any human interaction. However, in the event of an unusual occurrence (e.g. presence of an unknown debris type on the track) where the system's context base is not equipped to handle it autonomously, a human SME who is present would be alerted. At this point, the system would switch from its autonomous mode to decision-support mode to accommodate the SME in assessing the situation, and take appropriate decisions after consulting with a CxGs-enabled decision support system. Conversely, a CxG-decision support system can be used at the beginning of a mission to help an SME direct the system through an area where autonomy is not advisable (e.g. engaging a submarine vessel or a space exploration system). Nevertheless, at some point in the mission, the SME could be alerted to switch the system to its autonomous mode so that speed and accuracy can be accomplished at the same time in a domain where the system is prepared to operate. The above examples can serve as insights to future researchers as they research ways to enable their systems with multiple techniques so that they will be both efficient and versatile.

APPENDIX A: ASSUMPTION FOR THE HBR ENGINES

- To provide necessary protection for the HVA, the area CINC will leave picket ships on station, even if provided information of a red sub(s) getting underway.
- If a red sub(s) fire, the blue sub will have knowledge of which ship was attacked.
- If any sub initiates an attack, all rules of engagement (ROE) change to weapons free.
- Red subs don't snorkel.
- Red subs have infinite battery life.
- The blue sub cannot report at speeds greater than 6.2 meters/second.

- **Red sub hostile actions**:
  1. The following actions by a red sub are considered to be hostile acts: weapons fire, closure to within 5 NM of a blue surface asset, two "hostile course changes," e.g. toward the blue sub.
  2. There have been no sorties from the port during the prior month, so any potential sortie, e.g. a sub moving out of port, would be considered "unusual" activity.
- Active sonar will not be played.
- A weapon will only kill the asset at which it is fired.
- Depth is not considered.

APPENIDIX B: HBR INITIALIZATION PARAMETERS

Initialization parameters shall [1F] be passed to the HBR federate via an initialization interaction.

- The total number of red subs in the port shall [2F] be an initialization parameter.
- The probability with which a red sub(s) may initiate an attack on a blue asset between 12 NM and 100 NM from the coast shall [3F] be an initialization parameter.
- The criteria for port activity and the probability for determining the probability of a sortie within 2 hours are [4F, 5F], the initialization parameters. The listed probabilities are representative, not fixed.
- The relative value of assets to the blue commander shall [6F, 7F, 8F, 9F, 10F] be initialization parameters.
- The number of torpedoes available to the blue sub shall [11F] be an initialization parameter.
- The maximum possible number of torpedoes available to the red sub shall [12F] be an initialization parameter. The red sub may carry fewer weapons than the maximum.
- The red exercise area in which red forces perform normal exercise training shall [13F] be an initialization parameter.
- The port monitoring area shall [14F] be an initialization parameter.
- The torpedo speed shall [15F] be an initialization parameter.

- Maximum blue sub speed shall [16F] be an initialization parameter.
- Blue torpedo probability of kill within .025 NM of the targeted asset shall [17F] be an initialization parameter.
- Blue torpedo range shall [18F] be an initialization parameter.
- Maximum red sub speed shall [19F] be an initialization parameter.
- Red torpedo probability of kill within .025 NM of the targeted asset shall [20F] be an initialization parameter.
- Red torpedo range shall [21F] be an initialization parameter.
- The port location shall [63F] be an initialization parameter defined by the center point of a circle.
- The coast location shall [64F] be an initialization parameter defined by six points listed in west to east order, indicating pairs of end points.

APPENDIX C: RED AND BLUE FORCES

- The probability of detection is tied to target speed, detecting platform speed, and range. The probability of detection shall [22] be.
- Establishment of contact shall [23] be based on one successful detection.
- Loss of contact shall [24] be based on a target not being detected for 4 successive 1-minute frames.
- Subs shall [25] maintain a minimum speed of .5 m/s.
- The maximum sub turn rate shall [26] be 2 degrees per second.
- Torpedoes' fire direction shall [27] be independent of submarine orientation.
- Torpedoes shall [28] be detectable with a probability of 100% within 10 NM.
- A sub shall [29] only have one torpedo in the water at a time.
- When stationed in an area, an entity shall [65] patrol the area along a path through the center of the area and parallel to the x-axis (an east-west direction) across the diameter of the defined area.
- The maximum number of torpedoes available to any sub shall [68] be 30.

APPENDIX-D: FEDERATION REQUIREMENTS

- Coordinates shall [30] be Cartesian.
- Velocities shall [31] be in meters/second.
- Simulation federates shall [32] be time regulating and time constrained with unit of time in decimal hours.
- The speed for stationing on port or in an area shall [66] be 5 knots.

APPENDIX E: HBR FEDERATE

- A report of detected activity shall [33] consist of “transmit” for 1 minute followed by “listen” for an additional 4 minutes, requiring traveling at reporting speed for a minimum of 5 minutes.
- The blue sub shall [34] report based on the priorities.
- The following commands shall [35F] be passed from the HBR federate:
    - ♦ Station on port (area) to monitor (patrol-along or patrol-between) [36F]
    - ♦ Transit from A to B (move-to) specifying speed and destination [37F, 38F, 39F]
    - ♦ Station on unit (follow-entity) specifying unit ID and true bearing [40F]
    - ♦ Station in area (patrol-along or patrol-between) specifying area [41F]
    - ♦ Trail contact (follow-entity) specifying unit ID [42F]
    - ♦ Disengage trailing on loss of contact specifying unit ID [43F]
    - ♦ Fire (set target command/set rules of engagement command) specifying unit ID [44F]
    - ♦ Report specifying including port activity, number of new contacts up to the maximum number of red subs, and red sub threatening blue assets [45F, 46F, 47F]
    - ♦ Cease reporting [48F] and Set speed [67F].

APPENDIX F: RED FORCES AND FRAMEWORK

- The red sub shall [49] depart the port in a detectable posture if the blue sub is within the port monitoring area.
- A red sub(s) shall [50] respond if attacked by the blue sub.
- A red sub(s) shall [51] fire on a blue asset within 12 NM of the coast.
- A red sub(s) shall [52] not initiate an attack on a blue asset outside of 100 NM from the coast.
- Red subs shall [53] react based on detection of the blue sub by the DF located at the port.

The following data shall be passed to the HBR federate:

- ♦ A sensor report per cycle with a list of red submarine contacts including unit ID, location, and velocity [54F, 55F, 56F]
- ♦ Location and velocity of blue assets [57F*, 58F*] as attribute updates
- ♦ Firing of weapon by red specifying target unit ID [59F]
- ♦ Arrival at “point B” of a transit specifying unit ID [60F]
- ♦ Detonation/expiration of weapons [61F*]

APPENDIX G: FEDERATION OBJECT MODEL (FOM)

The FOM is based on the VR Forces version 1.3 SOM and is included as a separate file in Object Model Development Tool format (.omd).

Initial Scenario Description

"Playbox" defined by following points:

(0,0)

(0, 444000)

(448000, 444000)

(448000, 0)

(Approximately 36N 1E, 40N 1E, 40N 6E, 36N 6E)

Coastline

For coastline purposes, the blue sub shall not go south of border defined by

West of port: (0, 91000), (180000, 91000)

Port area: (180000, 79500), (215000, 79500)

East of port: (215000, 100500), (448000, 100500)

Unit Information:

Blue HVA

VR Forces name: 1 BlueHVA, 1 Force

Starting position: (288125, 359465) (approximately 39-15N 4-15E)

Operating area: 46000 m (approximately 25NM) radius circle centered at starting position.

Blue AAW1

VR Forces name: 2 BlueAAW, 1 Force

Starting position: (224015, 247875) (approximately 38-15N 3-30E)

Operating area: 22000 m (approximately 12NM) radius circle centered at starting position.

Blue AAW2

VR Forces name: 3 BlueAAW, 1 Force

Starting position (355275, 249650) (approximately 38-15N 5E)

Operating area: 22000 m (approximately 12NM) radius circle centered at starting position.

Blue Sub

VR Forces name: 4 BlueSub, 1 Force

Starting position: (192130, 109085) (approximately 37-02N 3-08E)

"Monitor Port" area: 7400 m (approximately 4NM) radius circle centered at starting position.

Red Sub(s)

VR Forces name(s): 1 RedSub, 2 Force

2 RedSub, 2 Force

Starting position: (192175, 79500) (approximately 36-44N 3-08E)

<u>Red</u> port is considered to be at this location

Red Exercise area: 18500 m (approximately 10NM) radius circle centered at (158050, 118350) (approximately 37-05N 2-45E)

APPENDIX H: EXAMPLE OF AN ACTUAL VIGNETTE SPECIFICATION, A SEA VIGNETTE SPECIFICATION

Mission Context: Intel and Protect HVA

Description: Collect intelligence about marine and submarine vehicles passing through an inlet while also protecting a High Value Asset that is located about 200NM out at sea.

Weather: Underwater

Lighting: Underwater

Location: Surrounding water near Red port

Objective: Collect intelligence without detection and protect HVA from hostile contacts

1. **Intel Gathering** Main Context

Function #1: Seek appropriate location. Determine safest position to avoid detection and accomplish mission context.

Function #2: Define depth. Set depth.

Function #3: Run quiet. Turn off engines.

Function #4: Monitor. Continually check all sensor equipment for contacts and contact information.

Sub-Contexts: **avoid-floor, avoid-other-vehicles**

2. **Transit to Intel Position** Main Context

Function #1: Sprint and drift. After seeking the appropriate location, move sub to that location.

Sub-Contexts: **avoid-floor, avoid-other-vehicles**

3. **Protect HVA and hold position** Main Context

Function #1: Warn. Make noise to indicate presence. Fire warning shot, if necessary.

Function #2: Attack.

Sub-Contexts: **avoid-floor, avoid-other-vehicles**

4. **Track** Main Context

Function #1: Get into position. Move sub into safe position to follow contact.

Function #2: Monitor. Follow contact at safe distance and bearing.

Sub-Contexts: **avoid-floor, avoid-other-vehicles, follow-contact**

5. **Attack** Main Context

Function #1: Get into position. Move sub into safe position to follow contact.

Function #2: Fire. Fire torpedo at contact.

Function #3: Take evasive maneuver. Change heading away from contact.

Sub-Contexts: **avoid-floor, avoid-other-vehicles, follow-contact, get-in-firing-position, fire-torpedo, evade-contact**

6. **Evade Attack** Main Context

Function #1: Take evasive maneuver. Change heading away from contact.

Sub-Contexts: **avoid-floor, avoid-other-vehicles, evade-contact**

7. **Surface (if damaged)** Main Context

Function #1: Blow ballasts.

Sub-Contexts: **blow-ballasts**

8. **Communicate** Main Context

APPENDIX I: SUB-CONTEXTS

1. **avoid-floor:** If the sea floor gets to within a certain range or the sub approaches its maximum depth, steer the sub to a more shallow depth.
2. **avoid-other-vehicles:** Steer the sub away from a target if a target approaches within a certain range.
3. **follow-contact:** Steer the sub towards the rear of a contact and fall in line behind that contact while maintaining a velocity equal to that of the contact.
4. **evade-contact:** Steer the sub away from the contact to avoid detection, collision, or attack.
5. **get-in-firing-position:** Get within certain range of target contact
6. **fire-torpedo:** If the contact is within range, target the contact and fire a torpedo
7. **blow-ballasts**: Blow water from ballasts into the surrounding seawater. Monitor the depth.

APPENDIX J: HBR CHALLENGE TABLES

Table C-1 DMSO Decision Taxonomy A: Sensation

Verbs	References	Simple Definitions
Acquire	10	To gain by one's own efforts, to obtain
Alert	11	To warn to be ready or watchful
Detect	2, 5, 11, 12, 14, 16	To discover something hidden, to notice, to observe
Discriminate	2, 5, 14, 15, 17	To distinguish between things
Extract	2	To deduce or derive, to take out from
Filter	2, 11, 21	To strain out unwanted data and so forth
Identify	2, 5, 11, 12, 13	To fix a person or thing as the one described
Inspect	2, 5, 16	To look at carefully
Localize	2, 5, 14, 16	To trace to a particular place, discover the position of
Monitor	2, 10,14	To watch, check, regulate performance
Recognize	2, 14	To identify as known before
Orient	14	To adjust to a particular situation
Perceive	14	To become aware of via senses, grasp mentally
Queue	11, 12	To form up in a line
Read	5, 11, 14, 16	To get meaning by interpreting characters
Receive	2, 11,14	To take or get freely given information
Search	2, 11, 12, 14, 16	To examine carefully for a thing concealed, survey

Table C-2 Taxon B: Mediation

Verbs	References	Simple Definitions
Adapt	11, 12	To adjust to new circumstances by changing
Adjudicate	3, 11	To hear, understand, form opinions, and decide
Aggregate	2	To gather into a mass, to total
Analyze	5, 11,14	To examine parts carefully to determine the nature of
Appraise	11	To estimate the quality or quantity of
Assess	14	To set an estimated value on, to set amount of
Categorize	2, 5,11, 12, 14, 16	To classify, to place in a class or category
Code	5, 11, 12 ,16	To put into a system or set of symbols
Compare	2, 5, 11, 14	To examine for similarities or differences
Compensate	14	To make up for, counterbalance
Compile	11	To gather together in order from various sources
Compute	2, 5, 11, 12, 14, 16	To determine an amount by reckoning, calculate
Copy	11	To imitate, to make just like another
Correct	14	To make right, to find errors, to make conform
Count	11, 12	To name the numbers in regular order, to add up

Decide	2, 4, 5, 11, 12, 13, 14, 15	To arrive at a judgment of one over another, select
Derive	14	To deduce or infer, to trace from or to a source
Determine	14	To set limits to, define, to establish exact nature of
Diagnose	1	To decide the nature of a problem by analysis
Edit	11, 12	To prepare for exhibition by arranging, revising, and so forth
Estimate	2, 5, 14	To determine generally, a judgment of probable cost
Extrapolate	2	To estimate an unknown on the basis of known facts
Evaluate	10	To find the value or amount of, appraise
Induce	14	To draw a conclusion from particular facts, persuade
Innovate	11	To introduce new methods or devices
Interpret	2, 3, 11, 12, 13, 14	To explain or translate
Interpolate	2, 5, 14, 16	To insert between or among others
Investigate	14	To search into or inquire
Itemize	2, 5, 14, 16	To specify the separate articles of
Know	11, 15	To have understanding of or skill in, as ideas
Match	14	To make equal or similar to, to fit things together
Name	2	To designate with the label for which it is known
Organize	10, 11, 14	To provide with an order or structure, systematize
Plan	2, 5, 8, 10, 11, 12	To create or have in mind a scheme for doing
Prioritize	2	To put in order on the basis of time or importance
Process	11, 16	To subject to a series of developmental changes
Purge	11, 12	To cleanse of impurities, to empty of that not needed
Quantify	2	To specify by numerical amount
Remember	2, 11	To bring back to mind with effort, recall
Revise	14	To read over to correct, to change or amend
Solve	13, 16	To find an answer to a problem, explain
Tabulate	2, 5, 14, 16	To put facts into columns or tables
Test	2, 11, 12, 14	To subject to events that verify the qualities of a thing
Think	11	To use the mind to reflect, reason, and conceive
Translate	2, 5, 14, 16	To put into another language, to change medium

Table C-3 Taxon C: Reaction

Verbs	References	Simple Definitions
Activate	5, 11, 14, 16	To cause motion or change
Align	5, 14, 16	To bring into proper coordination, into a straight line
Assemble	14	To fit or put together the parts of
Attack	14	To use force against in order to harm
Carry	11	To take from one place to another, to support
Climb	14	To go up by using the feet and often the hands
Close	5, 11, 14, 16	To block, bring together or finish
Combat	14	To fight or struggle against
Complete	14	To finish, to make whole or perfect
Connect	5, 11, 14, 16	To join or link
Deposit	14	To set down or leave lying
Destroy	14	To tear down, demolish, ruin, kill
Display	11	To unfold to the eye or mind, disclose, reveal
Do	11	To perform, to finish, to deal with as required
Drive	11	To control the movement or direct the course of
Eliminate	14	To get rid of, remove, leave out of consideration
Extend	14	To make longer, prolong, to stretch
Feed	11	To provide Something necessary for operation
Fill	14	To put To put as much as possible in, supply requirements
Glean	14	To collect gradually
Insert	11	To fit into something else
Lift	11	To bring up to higher position
Load	14	To put something to be carried into or upon a carrier
Loosen	14	To make or become unbound, unconfined
Maintain	10	To keep in continuance or in a state, as of repair
Manipulate	11	To operate with the hands, especially with skill
Move	5, 11, 16	To change the place or position of
Open	11	To make or become available for use
Operate	2, 11, 12	To put or keep in action

APPENDIX K: IRB APPROVAL LETTER

UNIVERSITY OF CENTRAL FLORIDA
THESIS APPROVAL

The members of the Committee approve the thesis entitled "A Comparative Analysis Between Context-based Reasoning (CxBR) and Contextual Graphs (CxGs)" of Peterson M. Lorins, defended May 23rd 2005.

Dr. Avelino Gonzalez, Chair

Ronald F. DeMara
Committee Member

Fernando Gonzalez
Committee Member

It is recommended that this thesis be used in partial fulfillment of the requirements for the degree of Masters of Science from the Department of Computer Engineering in the College of Engineering and Computer Science.

Issa Batarseh, Department Head

Jamal Nayfeh, Associate Dean, Academic Affairs and Graduate Studies

Neil Gallagher, Interim Dean College

Patricia J. Bishop
Vice Provost and Dean of Graduate Studies

LIST OF REFERENCES

Adam, F., (2002) how to use case studies for teaching and research in decision making and decision support systems, Proceedings of DSIAge2002, (pp.176-188) Cork: Oak Tree Press.

Barrett, G.B. and Gonzalez, A. J. (2002). Modelling Collaborative Behaviours in Context Based Reasoning, Proceedings of the Swedish-American Workshop on Modeling and Simulation (SAWMAS).

Barry, P (2001), Human Behavioral Representation Challenge Problem Kick off meeting, Defense Modeling and simulation Office (DMSO), U.S. government

Blemberg, P, and Bjorkman, A.E, (2001) Review of the Defense Modeling and Simulation Office Human Behavior Program, *Spring Simulation Interoperability Workshop*

Brézillon, P. (2003) Context dynamic and explanation in contextual graphs. In: Modeling and Using Context (CONTEXT-03), P. Blackburn, C. Ghidini, R.M. Turner and F. Giunchiglia (Eds.). LNAI 2680, Springer Verlag (http://link.springer.de/link/service/series/0558/tocs/t2680.htm). pp. 94-106.

Brézillon, P. (2003a). Context-based Modelling of Operators' Practices by Contextual Graphs, 14th Mini Euro Conference, Luxembourg.

Brézillon, P. (2003b) Representation of procedures and practices in contextual graphs. The Knowledge Engineering Review, 18(2): 147-174.

Brézillon, P., Pomerol, J., and Pasquier, L. (2002). Learning and explanation in a context-based representation: Application to incident solving on subway lines. LIP6, Case 169, University Paris, Paris, France.

Brézillon P., Pasquier L. and Pomerol J.-Ch. (2002) Reasoning with contextual graphs. European Journal of Operational Research, 136(2): 290-298.

Brézillon, P., Pomerol, J., and Pasquier, L. (2001). Operational knowledge representation for practical decision-making. LIP6, Case 169, University Paris, Paris, France, Proceedings of the 34th Hawaii International Conference on System Sciences.

Brézillon P. and Pomerol J.-Ch. (2001) Modeling and using context for system development : Lessons from experiences. Journal of Decision Systems. P. Humphreys and P. Brézillon (Eds.) : " Decision Systems in Actions ", 10(2): 265-288.

Brézillon, P. (1999). Context in Artificial Intelligence, Computer and Artificial Intelligence, Vol. 18, Number 4, pp. 321-340.

Brézillon, P. (1998). Modelling and Using Context in Applications, International Journal on Human-Computer Studies, vol.48 (3)

Brézillon, P., Pomerol, J., (1998). Using contextual information in decision-making. Conxtext Sensitive Decision Support Systems, Berkeley, D., Widmeyer G., Brézillon P. and Rajkovic V. (Eds.), Chapman et al., London., 1998, 158-173

Brézillon, P., Pomerol, J.-Ch. (1997) User acceptance of interactive systems: Lessons fromKnowledge-Based and Decision Support Systems. International Journal on Failures & Lessons Learned in Information Technology Management 1(1) 67–75.

Brown J.S., and Duguid P., (1991) Organizational learning and communities of practice: towards a unified view of working, learning and organization, Organizatoin Science 2 (1), 40-57

Fernlund, H.K. and Gonzalez, A.J. (2002) An Approach towards building Human Behavior Models Automatically by Observation, Intelligent Systems Laboratory, Orlando, FL.

Ganet, L., Brézillon, P. and Tijus, Ch. (2003) Explanations as contextual categorization. In: Modeling and Using Context (CONTEXT-03), P. Blackburn, C. Ghidini, R.M. Turner and F. Giunchiglia (Eds.). LNAI 2680, Springer Verlag (http://link.springer.de/link/service/series/0558/tocs/t2680.htm). pp. 142-153.

Gerber, W.J. (2001) Real-time synchronization of behavioral models with human performance in a simulation, Dissertation, ECE Dept. University of Central Florida

Giunchiglia, F. (1993) Contextual reasoning: Proceedings of the IJCAI-93 Workshop on Using Knowledge in its Context, Research report 93/13, LAFORIA 1993, pp. 39-48.

Gonzalez, A.J., Gerber, W.J., & Castro, J. (2002) Automated acquisition of autonomous knowledge through contextualization, BRIMS 2002.

Gonzalez, A.J., and Dankel, D.D. (1993). The Engineering of Knowledge-Based Systems: Theory and Practice, Englewood Cliffs, N.J.: Prentice Hall.

Gonzalez, A.J., Saeki (2000) Soft-Coding the Transitions Between Contexts in CGF's:

The Competing Context Concept, Mitsubushi Research Institute and University of Central Florida (UCF)

Gonzalez, A.J., Stensrud, B.S., Barrett, G.B., and Lisetti, C.L. (2002) Modelling Effect in Context Based Reasoning, Proceedings of the Swedish-American Workshop on Modeling and Simulation (SAWMAS).

Gonzalez, A.J. and Ahlers, R.H. (1998) Context Based Representation of Intelligent Behavior in Training Simulations, Naval Air Warfare Center Training Systems Division Conference.

Humphreys, P. and Brézillon, P. (2002) Combining rich and restricted languages in multimedia: enrichment of context for innovative decisions. Decision Making and Decision Support in the Internet Age. In: F. Adam, P. Brezillon, P. Humphreys and J.-Ch. Pomerol (Eds.). Oak Tree Press, pp. 695-708.

Kouadri Mostéfaoui, G. and Brézillon, P. (2003) A generic framework for context-based distributed authorizations. In: Modeling and Using Context (CONTEXT-03), P. Blackburn, C. Ghidini, R.M. Turner and F. Giunchiglia (Eds.). LNAI 2680, Springer Verlag (http://link.springer.de/link/service/series/0558/tocs/t2680.htm). pp. 204-217.

Leake, D. B.(1992) Evaluating Explanations. Lawrence Erlbaum Associates Inc.

Mackie, J.(1965) Causes and Conditions. American Philosophical Quaterly. 2 245–264

Nonaka I., (1994). A dynamic theory of organizational knowledge creation, Organization Science 5, 14-37

Norlander, Lars. (1999). A Framework for Efficient Implementation of Context-Based Reasoning in Intelligent Simulation, Master's Thesis, ECE Dept. University of Central Florida.

Pasquier L., Brézillon, P., and Pomerol J., (1999.), Context and decision graphs in incident management on a subway line. Modeling and Using Context (CONTEXT-99). In: Lecture Notes in Artificial Intelligence, Number 1688, Springer Verlal, 499-502

Pomerol J., and Brézillon, P. (1999)., Dynamics between contextual knowledge and proceduralized context. Modeling and Using Context (CONTEXT-99). In: Lecture Notes in Artificial Intelligence, Number 1688, Springer Verlag, 284-295

Pomerol J.-Ch. and Brézillon P. (2001) About some relationships between knowledge and context. Modeling and Using Context (CONTEXT-01). Lecture Notes in Computer Science, Springer Verlag, pp. 461-464.

Pomerol J.-Ch., Brézillon P.(1999) Dynamics between contextual knowledge and proceduralized context, in Modeling and Using Context (CONTEXT-99), Lecture Note in Artificial Intelligence n° 1688, Springer Verlag,284–295

Saeki, S., and Gonzalez, A.J. (2000) Soft-Coding the Transitions between Contexts in CxGs: The Competing Context Concept, Proceedings of Computer Generated Forces (CGF's) and Behavior Representation Conference.

Stensrud B. S, and Gonzalez, A.J. (2003). Context-Based Reasoning: A Formal Definition, Master's Thesis, ECE Dept. University of Central Florida. Forthcoming.

Turner, R.M. (1998) Context-Mediated Behavior for Intelligent Agents, International Journal of Human-Computer Studies: Using Context in Applications, Volume 48, Number 3, March, pp. 307-330.

Turner, R.M. (1999) A model of explicit context representation and use for intelligent agents, Proceedings of the Second International and Interdisciplinary conference on Artificial Intelligence (CONTEXT'99), Trento, Italy, September 9-11.