STELLINGEN

behorende bij het proefschrift

An intelligent gaming environment

Dirk Coppieters

12 november 1990
I

Voor het valideren van beslissingsondersteunende systemen is de vertrouwdeheid met de gebruikersinterface belangrijker dan de gebruiksvriendelijkheid.

II

Complexe organisaties zullen in toenemende mate gebruik maken van gaming voor het trainen en selecteren van hun managers en voor het ondersteunen van strategische beslissingen.

III

Het gebruik van een computer-"muis" is voor een beginnende gebruiker moeilijker dan de bediening van een toetsenbord.

IV

Verspreide systeem-architecturen zijn goedkoper in de aankoop en in het gebruik dan geconcentreerde. Verspreide systeem-architecturen zijn echter moeilijker te ontwerpen, te implementeren en te beheren dan geconcentreerde.

V

Het ontwerpen van kennis-verwerkende systemen zal pas door experts kunnen worden uitgevoerd, wanneer een omgeving wordt geboden die een grafisch ondersteund formalisme gebruikt. Verder is een geautomatiseerd, inductief mechanisme nodig dat voorbeelden kan analyseren en vertalen in het eerder genoemde formele kader. Tenslotte moet een opgebouwde beschrijving automatisch vertaald kunnen worden in een executeerbaar prototype.

VI

Budgetten voor de ontwikkeling en aankoop van Informatie Systemen (IS) die uitgaan van een jaarlijks goedkeuringsproces en die gedetailleerde specificaties van hard- en software vragen werken vertragend op het tot stand komen van IS. Aankoop-procedures die de concurrentie maximaal de kans laten en waarbij extra controles nodig zijn door het overschrijden van bepaalde kosten-drempels leiden
tot vertragingen in de ontwikkeling van IS en verhogen de kans op een kwalitatief minder goede oplossing. Een betere afstemming tussen budgettering, aankoop en ontwikkeling kan bereikt worden door vroegtijdig een globale kostenraming te maken van het uiteindelijk te realiseren IS en daarvan een bepaald percentage (15 - 20) aan ontwikkeling toe te wijzen ongeacht de daarvoor noodzakelijke middelen en het moment van uitgave.

VII

Het in meer detail beschrijven van componenten van een model leidt niet altijd tot een verbeterde correspondentie met de realiteit. Het tegendeel kan zelfs optreden.

VIII

Het gebruik van een simulatie-omgeving om de kwaliteit van Informatie Systemen te meten is slechts nuttig wanneer de informatie stromen en de bijbehorende infrastructuur van het Reële Systeem in het simulatie model beschreven zijn.
AN INTELLIGENT GAMING ENVIRONMENT
AN INTELLIGENT GAMING ENVIRONMENT

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van Rector Magnificus,
Prof. Drs. P.A. Schenck,
in het openbaar te verdedigen
ten overstaan van een commissie
door het College van Dekanen daartoe aangewezen,
op maandag 12 november 1990 te 14.00 uur

door

DIRK COPPIETERS

geboren te Dendermonde, België

Delft University Press
Dit proefschrift is goedgekeurd door de promotor
Prof. Dr. H.G. Sol
en door de leden van de promotiecommissie
Prof. Dr. H. Koppelaar
Prof. Dr. Ir. M. Looijen
Prof. Dr. Ir. C.A.Th. Takkenberg
Prof. Ir. D.H. Wolbers
Ir. J.J. Meinardi
To all the players on my team
Contents

PREFACE

Chapter 1: Warfare simulation models
1.1 Introduction 1
1.2 Taxonomy of war gaming models 7
1.3 Important characteristics of war gaming models 11
1.4 Conclusion 12

Chapter 2: Modelling and decision making
2.1 Problems, models and simulation 15
2.2 Bounded and unbounded rationality 19
2.3 War gaming evaluation factors 20
2.4 Current war gaming problems 22
2.5 Modelling based on the paradigm of bounded rationality 27
2.6 Artificial intelligence 31
2.7 Problem definition and research questions 35
2.8 Theory 37

Chapter 3: Case description
3.1 A problem situation 43
3.2 Current situation 43
3.3 Main war gaming goals and strategies 49
3.4 War gaming development strategy 54
3.5 Application development planning 58

Chapter 4: Developing decision objects
4.1 Modelling and structuring of decision objects 64
4.2 The brigade decision object 65
4.3 A logistical decision automaton 68
4.4 An intelligence decision automaton 75
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5 A staff decision automaton</td>
<td>77</td>
</tr>
<tr>
<td>4.6 A tactical decision automaton</td>
<td>82</td>
</tr>
<tr>
<td>4.7 A communication decision automaton</td>
<td>86</td>
</tr>
<tr>
<td>4.8 An engineering decision automaton</td>
<td>88</td>
</tr>
<tr>
<td>4.9 A fire support decision automaton</td>
<td>90</td>
</tr>
<tr>
<td>4.10 The internal communication of the brigade decision object</td>
<td>90</td>
</tr>
<tr>
<td>4.11 A mission planning decision automaton</td>
<td>92</td>
</tr>
<tr>
<td>4.12 Development approach and knowledge acquisition</td>
<td>97</td>
</tr>
<tr>
<td>4.13 Implementation</td>
<td>100</td>
</tr>
</tbody>
</table>

**Chapter 5: The dynamics of the IALTA gaming environment**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Modelling and structuring</td>
<td>113</td>
</tr>
<tr>
<td>5.2 Phasing and management</td>
<td>128</td>
</tr>
<tr>
<td>5.3 Implementation</td>
<td>130</td>
</tr>
</tbody>
</table>

**Chapter 6: IALTA decision support objects**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 A game designer support environment</td>
<td>133</td>
</tr>
<tr>
<td>6.2 A game controller support environment</td>
<td>141</td>
</tr>
<tr>
<td>6.3 A player support environment</td>
<td>141</td>
</tr>
<tr>
<td>6.4 Implementation</td>
<td>144</td>
</tr>
</tbody>
</table>

**Chapter 7: Experiments**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Experimental design</td>
<td>147</td>
</tr>
<tr>
<td>7.2 The ILTA game</td>
<td>148</td>
</tr>
<tr>
<td>7.3 The IATA game</td>
<td>156</td>
</tr>
</tbody>
</table>

**Chapter 8: Epilogue**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>References</td>
<td>171</td>
</tr>
<tr>
<td>List of abbreviations</td>
<td>177</td>
</tr>
<tr>
<td>Samenvatting</td>
<td>181</td>
</tr>
</tbody>
</table>
PREFACE

Trying to understand the world we live in is something each of us does every day. The larger the part of reality that we must understand and manage, the more difficult it is for us to build our model of that reality and the more important it is that we are capable of making sound decisions. This dissertation presents a modelling approach that supports the development of a gaming environment that has a good correspondence with that part of reality which we are trying to understand. We discuss the necessary theoretical framework and illustrate the feasibility of our approach through a case study. Hence, we show that an object-oriented simulation model can be built that contains automata which can replace human players. This provides the required flexibility and cost-effectiveness to support decision makers in their process of learning to understand reality.

The reality of writing a dissertation is also complex and many people play a part in it. I would like to take this opportunity to thank all the people who have contributed to the successful completion of this work. First and foremost I wish to express my gratitude to Henk Sol for his patience, advice, criticism and support in the course of writing this thesis and developing the case model.

In this context, I would like to thank all the officers and staff of the Royal Belgian Defense College for placing their trust in me and providing me with the necessary support and expertise. More particularly, I would like to thank to Kol v/h Vlw SBH Roger Van Herck, Col d’Avi BEM Bernard Dubois, Col Avi BEM Jacques Verhaeren, Kol SBH Vermeerenbergen and LtKol SBH Daniel Dewever. I would also like to express my appreciation for my colleague Capt Jean-Paul Ganseman who protected me from the trappings of government administration and ensured that a pleasant and well furnished working environment was always available. I also want to thank all the conscripts who provided the necessary manpower and intelligence to complete the IALTA system and without whose cooperation this thesis could not have been validated.

For all the personal support, I am especially grateful to my family and friends who displayed the necessary humour to overcome my doubts and provided enough distraction to replenish my energy reserves.
Chapter 1: Warfare simulation models

1.1 Introduction

War gaming originated in Prussia in the late 18th century where it was used to train Army officers. Quality was already one of the main issues as illustrated in 1874 by Von Trotha when he states: "die Anleitung zum Gebrauch des Kriegsspiel-Apparates ... ist das kräftige Selbstdurchführen selbständig entworfener Dispositionen. Und das kann nur zu Geltung kommen, wenn die Uebungen auf neutralem Boden ausgeführt werden, d.h.so, dass nach Hauptgrundzügen verfahren wird, die sowol fur die Partelen als fur den Leitenden gleichmässig Geltung haben."

Growing emphasis has since then been given to gaming in both the military and civilian domain because high level decision making is increasingly crucial in a highly competitive environment. Again the military domain illustrates this clearly. Indeed, as Bartlett [1989] states: "Simulation undoubtedly will be a major component of future training strategies, perhaps even becoming the centerpiece of Army training. This crescendo of interest stems from many factors. Modern weapon systems are expensive training aids. A succession of zero growth budgets probably will lead to reductions in operating tempo and consequently, to unacceptable degradations in readiness unless alternatives to field training strategies can be found. Environmental awareness is another factor, especially in Europe, where the United States has substantial forces. Noise, manoeuvre damage and accidents have aroused ordinary citizens. Their ire, combined with shrinking training space, beg for alternative solutions. Glasnost, the Intermediate Nuclear Forces (INF) agreement, Soviet unilateral force reductions, Conventional Force Europe (CFE) talks and a general sense that 'peace is breaking out all over' are having an impact. The public perceives less need to endure the expense and inconvenience of military training." Furthermore Bartlett illustrates clearly the need for the training of decision making when citing the results of the US Army in opening battles: "In the first 10 battles during nine wars, the Army has suffered five defeats, four costly victories and only one clear victory. Peacetime training was the culprit." And he concludes: "In the future, high level commanders and their staffs must be among the trainees, and large-scale manoeuvres cannot be the classroom."

An intelligent gaming environment
The analogy with the market place can be drawn when one considers the growing globalisation and speed of trade. Many examples can be found in the mergers in Europe between large companies in all economic sectors in their preparation for the unified European market of post 1992.

The field of war gaming is particularly interesting to conduct research because it combines the modelling of a complex part of reality, namely the battlefield with a well documented body of knowledge i.e. military decision making. Furthermore, as discussed previously, military war gaming has always been a precursor in the gaming domain. The field of war gaming is currently faced with the need to devise novel approaches since existing models have evolved to such an extent that they have become difficult to manage in terms of consistency and cost. This is compounded by the rapid development of weapon systems and changing strategies and tactics. Therefore there is growing concern about the correspondence current models have with reality.

In order to further illustrate the complexity of the war gaming domain, let us consider the command echelons which are nationally relevant. Firstly an Army Corps comprises between 70000 and 100000 men and women, some 6000 vehicles and a wide variety of weapon systems. Secondly an Allied Tactical Air Force manages some 50 airbases, 1500 airplanes which represents approximately 10000 men and women. Both these command echelons are further structured into some 3 to 10 subordinate echelons that correspond to units of different size and military capability.

It appears that managing such a wide variety of units is mainly an information handling process since higher level command echelons do not execute any actions that have an immediate and direct effect on the battlefield. Rather they process information about the situation to generate orders for their subordinates.

The information paradigm defined by Brussaard and Tas [1980] is often used in the information systems discipline to differentiate between those aspects of reality that deal with the execution of actions and those aspects that manage and control these actions. This paradigm states: "Every group of interdependent dynamic phenomena can be described as a real system (RS) and an information system with the information system (IS) controlling the real system.". This paradigm encourages us to consider every activity in reality as the combination of a planning and control activity with an execution activity.

An intelligent gaming environment
We apply this paradigm to the war gaming arena and identify three nested levels of RS-IS combinations, see figure 1.1

![Diagram showing three nested IS-RS combinations](image)

*Figure 1.1: The three nested IS-RS combinations*

An intelligent gaming environment
The highest level is the game design level. We look at the war gaming environment as a tool to achieve certain objectives and we must therefore define how we intend to use this environment. So, the RS consists of the war gaming environment and the participants in the gaming experiment. The IS on the other hand is composed of the game control staff, the procedures that govern the operation of the gaming environment and the tools that are available to tailor the gaming environment and set up experiments.

The following level is the gaming environment level. Here we consider the activity of actually performing gaming experiments. The RS consists of reality as represented in the simulation model and the IS is formed by the player staffs, the gaming procedures and the tools that support the communication between staffs and the decision making of players.

The third level is the simulation model level. We focus on the activities of objects represented in the simulation model and on their interactions. The RS in this case is the battlefield and the IS is constituted by the command echelons, the interaction procedures and the communication and decision support tools represented in the model.

Figure 1.2 provides a more detailed description of the previous figure. It shows the general structure of a war game and highlights the relations between the three levels of IS.

This description also allows us to identify the main constituents of a war gaming environment:

a. the player staff: this is a group of decision makers who perform the planning and the conduct of the battle at one or more command echelons;

b. the support staff: this is mostly a group of people who represent the echelons below the lowest one being played by the player staff. Their task is twofold:

- translate the decisions made by the player staff into actions that can be understood by the model of military reality;

- translate the information about the state of reality into information that can be used by the player staffs;

An intelligent gaming environment
Fig. 1.2: Overall war gaming structure

An intelligent gaming environment
c. the game control staff: this group of people is responsible for:

- verifying that the game evolves according to its objectives;

- providing the player staff with orders from the higher echelons and other organisations that, in reality, interact with the player staffs;

d. the military IS which is constituted by units. They are the active component that can achieve the goals of the player staff according to their orders. There are several types of unit that can execute actions in the model of military reality. Mostly a unit is defined as consisting of a number of people, vehicles of different types and several types of weapon systems. It should be noted that units communicate and coordinate actions and as such form the Command and Control Information System (CCIS).

e. a model of military reality (MMR): this is a representation of that part of reality that is relevant to the player staff and the game they are performing. The representation of the military IS is also part of the MMR since the resources it uses, e.g. sensors and command posts, are also modelled. Aspects of military reality which are particularly important are:

- resource: this corresponds to weapon systems, support equipment and personnel. As such, it is closely linked to the unit because as is described above a unit uses these resources to execute its orders. Therefore, its relevant characteristics are described in the model e.g. the characteristics of a vehicle or of a weapon system;

- environment: this represents time, terrain and weather and is intended to show the dynamic and spatial aspects of military reality;

f. an evaluation model: it is responsible for the evaluation of interactions between objects of the model of reality either interactively by hand or automatically. It decides about the outcome of the interactions between the simulation objects and may thus redefine their state.
1.2 Taxonomy of war gaming models

After the introduction of the historical background of gaming and its growing importance in both the military and civilian field, we have illustrated the general structure of a war game. In order to understand this field more clearly, we describe the applications of war games and propose a categorisation for them.

Many taxonomies have been suggested to categorise war gaming models, e.g. Ackoff [1962], Low [1977], Battilega and Grange [1978], Brewer and Shubik [1979], Taylor [1980] and Hoeber [1981]. As suggested in Military Modeling [1984] these taxonomies can be summarised as follows:

1. Models characterised by the supporting resources used, e.g. computer simulation or a field training exercise.

2. Models characterised by their purpose, e.g. training or plan analysis.

3. Models characterised by their level of representation, e.g. micro-level or single unit engagement, meso-level or multiple unit engagement, macro-level or all units in a specific region of the world.

4. Special or general purpose models, e.g. models built to illustrate a specific problem in time or models that are maintained and are used to support various objectives.

Each of these taxonomies is applicable to conduct research about war gaming models. It can also be remarked that these taxonomies can actually be used in a nested manner. Within the categorisation of models by their level of representation, we could categorise by purpose or by supporting resources. Therefore, we see that these taxonomies can be integrated. In the next paragraphs, we have chosen to categorise warfare simulation models by purpose, namely training, plan analysis, weapon effectiveness, weapon mix and force mix testing. We believe this taxonomy is interesting because it closely mirrors the composition of the community of war gaming users. Let us then discuss each of these types of gaming model separately.
1.2.1 Models for training

Models for training can be used for many purposes which centre around two main domains:

1. the training of procedures, i.e. the military method of working;

2. the training of decision making, i.e. the military method of thinking and of making decisions.

The objective of models in the first category, is to provide officers with a realistic environment to train military procedures at each command echelon. Military procedures entail the writing of orders, reports and messages and the presentation of these orders. Each command echelon uses some specific procedures which must be rehearsed. Models that support this kind of training must therefore provide an evolving situation to trigger the use of the military procedures by the player staffs. However it is not required that the models provide accurate results because the emphasis is on the efficiency (how the work is done) rather than on the effectiveness of the staff work.

War games intended to train the decision making process must address the fact that military decision making involves the planning and conducting of the battle. This type of training using computer aided simulation is most suited for the higher command echelons because no other type of exercise can realistically provide a player staff with the necessary freedom to make whatever decision they deem necessary. Indeed, field training exercises are constrained because of peace time conditions which limits the freedom of decision making. Furthermore, the high level decisions are often scripted before the exercise takes place because the dynamic evolution of the situation cannot be allowed to progress in an uncontrolled manner due to peace time constraints. War games that support the training of decision making emphasize the effectiveness of the staff work which entails the making of the necessary decisions at the various command echelons.

The significant difference in application between the training of procedures and the training of decision making lies in the user interface and the correspondence with reality which the model must provide. Models that support the training of procedures must provide a user interface which emulates as closely as possible the current format of orders, reports and messages. In the case of supporting the training of

An intelligent gaming environment
decision making, the format of the information is less important. The flexibility with which information can be accessed, presented and updated is essential. However the trainees should not be presented with information processing capabilities that are too distant from the support that is currently available.

As stated above, the correspondence with reality is paramount when training decision making. This is due to the fact that the decision makers are trained at a high level and develop certain expectations towards the course of events in reality based on the evolution of the war games. Furthermore, these decision makers already possess a large experience with reality and must recognize this correspondence in the model if they are to use the war game effectively.

Therefore a multi-purpose training model should combine several user interfaces with a high degree of correspondence with reality.

1.2.2 Models for plan analysis

Models for plan analysis can be used to evaluate plans at the tactical, logistical and political level. The tactical level addresses national Army and Navy and conventional Air force missions. The logistical level spans the entire spectrum of logistical support. Finally, the political level considers multi-national missions.

These models are typically closed models, which are defined in the following paragraphs, or models with a low level of resolution, i.e. only a few (2-4) command echelons are represented in the model. Therefore these models rely heavily on highly aggregated payoff or cost functions to determine the outcome of interactions between high level components. These functions are determined using the outcomes of war games or closed simulations that simulate the lower command echelons. To this they add the high level tactical concepts of a supposed adversary.

Models for tactical plan analysis represent the highest national command echelons and consider only limited use of nuclear weapons in a restricted region of the world. They are intended to determine the outcome of a plan carried out with one or more of these high level units.

Models for logistical plan analysis concentrate on the support of the fighting units with supplies, repair capability and medical support. The aim here, is to evaluate

An intelligent gaming environment
the capability to provide sufficient support to the troops in the field and how this can be achieved.

Models for political plan analysis concentrate on the balance of forces between the two main military alliances, NATO and the Warsaw Pact. The world wide balance of military resources is also studied. These models represent the highest national commands and the command echelons of the existing alliances. These models can be used to evaluate the deployment of forces or the means to coordinate these forces.

For these three types of model, the emphasis is on the evaluation of the outcome of courses of action when own forces are opposed to an expected opponent.

1.2.3 Models for weapon effectiveness testing

Models for the testing of weapon effectiveness compare different types of weapon within a given situation. They describe in great detail the weapon performance and application process and do not require any command echelons to be represented within the model. This means that the effect of a given weapon is evaluated against a given target. Such an analysis can be done within the context of an existing weapons mix or purely on the basis of weapon-target combinations. Models for the testing of weapon effectiveness rely heavily on the laws of physics and ballistics since a supposedly purely mechanistic process is being evaluated.

These models are used in the testing of weapon systems and the evaluation of new technologies.

1.2.4 Models for weapon mix testing

Models for weapon mix testing are comparable to the previous ones, but they put the emphasis on evaluating the combination of weapon systems against a given target mix. These models are used for long term planning and purchasing of weapons systems.
1.2.5 Models for force mix testing

Models for force mix testing concentrate on the allocation of resources to units and the organisation of these basic units into larger units. The composition and resource allocation of units starts at the lowest levels with the allocating of basic weapon systems. The higher unit levels are organised using these building blocks. The ensuing organisation is then tested using the defined tactical and logistical concepts against a given opposing force.

These models are used for the long term planning of weapon systems purchasing and for personnel planning.

1.3 Important characteristics of war gaming models

1.3.1 Level of abstraction

Each of the types of model described earlier, can be represented at three levels of abstraction i.e. micro-, meso- and macro-level.

Models at micro-level attempt to evaluate the effectiveness of a given basic unit, as described above, against another. These models describe the conflict between single small sized units.

Models at meso-level represent multiple unit engagement. These models describe the conflict between several medium-sized units and are used to evaluate the effectiveness of unit combinations or the organisation of units versus a potential opponent.

Models at macro-level, i.e. theatre level e.g. Western Europe, represent the highest levels of command and possibly political decision making in order to evaluate high level decision problems in global matters.

1.3.2 Type of interaction

War gaming models can further be characterised by the command echelons where human decision making is required to drive the simulation:

An intelligent gaming environment
- closed simulation: all decisions and evaluation functions are represented in a simulation model;

- open simulation: all decisions are taken by human players and the evaluation functions are carried out by a simulation model and by human game controllers;

- semi-closed simulation: like closed simulation but some command echelons are enacted by human players and some evaluation functions may be carried out by human game controllers.

1.4 Conclusion

As discussed in this chapter, a taxonomy based on the objective of war games allows us to define five types of war gaming models. Based on the descriptions given in paragraph 1.2, we may identify the main experimental variables of each of these types of model.

1. Models for training must enable the user to apply an existing strategy using a given set of resources within a well defined command structure. It is therefore acceptable that certain detailed rules of strategy, resource descriptions and command structures are implicit in the model.

2. Models for plan analysis put the emphasis on the variation of the applied strategy. They must therefore be void of any implicit rules of strategy.

3. Models for weapon mix testing focus on the variation of the resource combinations to implement a given strategy. These models should therefore refrain from having any implicit resource representations. The second main variable is hereby identified: weapon mix.

4. Models for force structure testing emphasize the variation of the level of decision making within a given strategy and set of resources. This type of model cannot contain implicit representations of organisational structures or distribution of resources to command echelons. Force structure is the third major variable.

5. Finally, models for weapon effectiveness testing focus on the behaviour of a well defined type of equipment within a very limited number of well specified situa-
tions. These models do not take into account the organisation or strategy they might be applied for.

Our analysis results in three main variables with which we wish to experiment in war gaming models:

1. strategy: doctrine or tactic e.g. rules of war;
2. weapon mix: combination of weapon systems;
3. force structure: distribution of decisions and resources to echelons.

We contend that war gaming models, except for weapon effectiveness testing models, must address these three main variables to a certain extent and with a given measure of flexibility. However, due to current modelling techniques and development approaches, we have noticed that models specialise in one main variable. Indeed, the other main variables that are not required to vary have been reduced to a static description which corresponds to their current state. Such an approach leads to a real proliferation of models on the basis that each new problem requires a new model.

We may therefore build a taxonomy based on two categories within which models can be characterised as follows:

1. Weapon effectiveness testing model: models are categorised by type of weapon system.
2. General purpose model: models are characterised by the level of flexibility of strategy, of force structure and of weapon mix that they provide.

As shown in figure 1.1 and 1.2, warfare simulation models must provide sufficient means of interaction to player staffs, require decision making support for design and execution of war games and must represent the dynamics of military reality.

The objective of our research is to bring together novel concepts from the domains of highly interactive systems, decision making support systems and simulation. We intend to define a method of thinking, modelling, working and managing to develop intelligent gaming models. These models are general purpose models which repre-

An intelligent gaming environment
sent complex sub-systems of reality and provide an improved degree of correspondence with reality compared to existing models. A case study in the field of war gaming models is used to show how this objective can be achieved.

In order to define the framework and provide an illustration of how to build such a general purpose warfare simulation model Chapter 2 sets the methodological context of this dissertation by defining the concepts and processes associated with modelling and decision making. Furthermore it describes problems that current war gaming models face and scientific knowledge domains that can contribute to their solution. We introduce the concepts we suggest to solve current war gaming problems. The proposed framework should also enable us to achieve an improved correspondence with reality.

Chapter 3 describes how this framework will be tested. It illustrates war gaming problems at the Belgian Royal Defense College and shows that this example is generally applicable to the field of war gaming by defining the overall methodological and cost-effectiveness problems faced by current models.

Chapters 4 through 6 describe the building of the IALTA, Integrated Air Land Training Application, model. Chapter 4 illustrates the development of decision objects which should be able to replace player staffs when this is required. Chapter 5 describes the development of the model of military reality. Chapter 6 discusses the development of the player staff environment.

Chapter 7 illustrates the typical applications of the IALTA as it is used in the context of the Defense College. The organisation of these games is described as well as the experimental results each of these games provided to evaluate our approach.

Finally chapter 8 draws conclusions about the conceptual framework that was tested and attempts to show how these conclusions applying to the war gaming field can be generalised to define a methodology for the broader spectrum of gaming models that represent complex sub-systems of reality.

An intelligent gaming environment
Chapter 2: Modelling and decision making

2.1 Problems, models and simulation

Participating in a war game as a player or as a unit in the simulation model supposes the ability to perceive and solve problem situations. In order to understand and model the process of problem recognition and solving, we use the definition of a problem and a problem solving process as formulated by Sol [1982 p.2,3]:

*a problem exists when a problem owner is in doubt as to which choice is best to remove his dissatisfaction with his present state, where he can identify one or more outcomes that he desires, two or more unequally efficient or effective courses of action, and an environment containing factors that affect the outcomes.*

The problem-solving process is defined as a four stage process:

*a problem as perceived in reality, a conceptual model, an empirical model and a solution. The stages are connected by the activities of conceptualization, problem specification, solution finding, implementation, consistency check and correspondence check.*

The relations between each of the stages are defined as follows:

*Conceptualization comprises firstly the choice of a vehicle for communication i.e. a language which can be based on words, diagrams and possibly empirical models derived from conceptual models for similar problems [D.C.] and a metatheory. The metatheory comprises a Weltanschauung and a construct paradigm i.e. a way of observing and structuring that part of reality one is trying to model [D.C.]. Thereafter, a conceptual model of the problem situation is formulated with the help of these concepts. When the conceptual model of the problem situation has been arrived at, an empirical model has to be specified. The analysis of the empirical model of the existing situation must clearly define the problem specification. The connection between the empirical model and the problem as perceived in reality reflects the correspondence between these two.*

An intelligent gaming environment
This definition is useful in our discussion of modelling because it emphasizes the need for a reduction of the problem domain and shows the interaction between the state of the decision maker and his environment. Furthermore, this definition implies an explicit conceptualisation of the problem domain and its environment.

It also introduces the concept of correspondence. A good measure of correspondence is essential for models and thus for war gaming models. Achieving a good measure of correspondence in a model forms the basis for the definition of concepts to enhance current approaches for the development of warfare simulation models.
In this context we adapt the definition of a model as given by Elzas [1986 Ch. I.1]:

*an empirical model is a compact representation of a problem perceived in reality which can generate a behaviour comparable with some behaviour of interest in the perception of reality and moreover, a vehicle to make more evident key characteristics of an object under study.*

This definition shows the need for a reduction of the problem domain based on the goals of the study which can be the solving or understanding of a problem. So the link between problem and empirical model becomes apparent namely that the empirical model is a reduction in the course of the conceptualisation and solving of the problem. This is particularly important because, as we will show in the next paragraph, a great number of problems in current warfare simulation stem from a reduction that is too great and that overlooks certain key aspects of the real world in which military units evolve.

It is also important to note that both previous definitions support the use of the Singerian problem-solving process. This is a synthetic, interdisciplinary, process which attempts to integrate scientific, ethical and esthetic modes of thought [Sol 1982 p. 4].

In contrast with the Singerian problem-solving process, the Leibnitzian problem solving process is a purely formal and deductive process of problem-solving. We will show in the next paragraphs that the latter forms the current basis for gaming models which is the cause of many problems of correspondence with reality. The next example illustrates the difference between both problem-solving processes.

We place a military decision maker in the problem situation described below and apply both problem-solving processes to reach a solution.

**Problem situation statement:** a military unit has received the order to attack and is faced with the problem whether or not to define an intermediary objective. Indeed defining an intermediary objective would allow the unit to use its forces in a more effective way. However the unit must reach its objective within a very limited time frame and defining an intermediary objective will mean that the unit has to stop for at least an hour to restructure its forces.

**Question:** Should the unit use an intermediary objective?
The Leibnitzian-type solution to this problem would be to calculate the time gained by using the intermediary objective and compare this with the one hour time delay. If there is sufficient time an intermediary objective would be defined.

The Singerian-type solution would apply the heuristic that when time is short, it is better not to define an intermediary objective, the decision maker will not use one but will reinforce his leading unit(s).

We consider the second solution to be preferable because it takes into account the intrinsic uncertainty of reality, the so-called "fog of war". Indeed, if reality were deterministic then the first solution would be better because it would be correct. There would be no variation during the real execution of the battle from the mathematical model of reality expressed by the battle duration equations. However as many scientific and philosophic theories have shown reality is not deterministic. A first example is quantum mechanics which defines a probability area around an atom's nucleus where one can expect to find the atom's electrons. However, it is not possible to determine their exact location at any given moment in time. Also, Plato's image of the cavern wall, that shows the shadows of the objects and people around the fire in the middle of the cavern, illustrates that reality is not what we perceive it to be and therefore doesn't look deterministic. And finally, Kant's combination of aesthetic and rational thought shows that knowledge is the result of a subject's experiments. Knowledge does not spring from the object itself. Therefore, absolute knowledge cannot be attained and we cannot expect to build a deterministic model of reality.

Chapter 1 showed that a war gaming environment comprises a simulation model. In order to define concepts for the development of such a model, we use the following definition of simulation based on Shannon's [1975, p.2]:

_simulation is the process of designing a model of a concrete system and conducting experiments with this model in order to understand the behaviour of a concrete system and/or to evaluate various strategies for the operation of the system._

Defining simulation this way is particularly useful because it combines the use of simulation and thus warfare simulation for both training e.g. understanding and analysis e.g. evaluation of various strategies. Furthermore the distinction between designing a model and experimenting with a model is interesting. Indeed, the war
gaming community often identifies a model by the experiments that are conducted with it. Chapter 1 showed a taxonomy that is based on the objective of a model and concluded that in fact only 2 types of model are required. So, war gaming models for training, plan analysis, weapon mix analysis and force structure analysis can be replaced by a multi-purpose model which allows these four types of experiment to be conducted.

2.2 Bounded and unbounded rationality

Building a simulation model of a part of reality requires a vehicle to observe and describe reality, a Weltanschauung and a construct paradigm. Simon [1976, p.65-86] introduced the paradigm of substantive rational behaviour to characterise the basis for the modelling of decision making processes.

We adopt the terminology defined by Sol [1982, p.7-8] to formulate this paradigm which he referred to as paradigm of bounded rationality. As such, unbounded rationality corresponds to the following rewritten definition based on Simon’s definition, adapted to our problem domain:

a. Unbounded rationality models prescribe how a decision should be made in artificial circumstances, where e.g. perfect knowledge and no computational restrictions exist.... The artificiality refers to the low degree of correspondence between descriptions and problems as perceived in reality. An example of this is the high degree of uncertainty of identification in air combat.

b. Unbounded rationality assumes a ‘military’ environment, a ‘military’ decision-maker and ‘military’ variables.

c. Unbounded rational behaviour is the model of the homo militaris. It assumes at the same time, a homo militaris who knows everything and for whom all data is accessible.

In contrast with this, bounded rationality:

a. is primarily concerned with the contents of procedures describing decision processes in ‘normal’ circumstances. Determinants of normal circumstances might be delays in the processing of data, imperfect foresight, inaccurate data, restricted capabilities to generate alternative solutions and to evaluate these.
b. emphasizes a good correspondence of the problem specifications and does not try to reduce the problem specification on beforehand by the use of a 'military' system of concepts.

c. can use a satisficing procedure to find a solution or solutions in a finite solution space, which might be expanded if an impetus originates in the satisficing procedure. In general this results in the best solution attainable, without any guarantee that this solution is an optimal one.

d. specifies decision-makers in their own environment with own definitions and limited scopes of accessible data.

2.3 War gaming evaluation factors

Having introduced the Weltanschauung that can be used to describe and build a war gaming environment, let us investigate the reasons for the definition of a new modelling approach towards the building of war gaming environments. In order to clarify the current modelling problems in the field of war gaming, we define an evaluation framework for existing war gaming environments.

First of all, we categorise problem situations as suggested by Sol [1982]:

a. we define a problem as being well-structured if the following conditions are met:

1. The set of alternative courses of action or solutions is finite and limited.

2. The solutions are consistently derived from an empirical model that shows good correspondence.

3. The effectiveness or the efficiency of the courses of action can be numerically evaluated.

b. problems that do not fulfil these requirements are defined as being ill-structured.

In the military field most command decisions at every level of the military hierarchy are ill-structured problems. Indeed, due to the fact that information is always uncertain and an infinite number of courses of action are available to both sides, it is impossible to evaluate quantitatively the result of the decision making process.
Therefore it is obvious that the quality with which ill-structured problems are modelled in a war gaming environment is an important factor in the correspondence of a model.

Correspondence is defined in Military Modeling [1984] as follows:

*the correspondence of a model is judged by how well a model achieves its purpose.*

In order to judge the correspondence of a model, we propose to use the following factors as adapted from Brewer and Shubik [1979, p. 329]:

- flexibility: this is defined as the applicability of a model for several types of war games; it measures the number of war gaming applications for which a model is suited;

- experimental validity: this is defined as the measure with which the model can be used for a given type of war game and the measure of reproducibility of a given experiment; it expresses how well a model is suited for a certain war gaming application;

- military realism: this is defined as the measure with which
  a. each object describes the characteristics and behaviour of its real counterpart,
  b. events are generated, processed and evaluated as is the case in reality,
  c. the measure in which ill-structured problems or decisions are represented;

- responsiveness: the ability for the user to interact with the model at given times and events compared to the interaction possibilities in reality;

- sensitivity of the model: the measure in which the model’s results vary according to the variation in the initial situation;

Furthermore we would like to highlight the practical applicability of a model by analysing:

An intelligent gaming environment
resources required: this is defined as the set of:

a. the required hardware and software configuration,

b. the required personnel to maintain the model,

c. the required personnel in a support staff role,

d. the required personnel in a player staff,

e. the required personnel in a game control staff;

- the level to which the models are open as related to the openness required by the objective of the simulation.

The correspondence and practical applicability factors cover the quality and cost-effectiveness aspect of a war gaming environment. As such they provide us a framework with which to measure the cost-quality relation.

2.4 Current war gaming problems

A number of problems concerning current war gaming models have been reported in recent literature.

A study by Candan, Dewald and Speight [1987] has shown that current war gaming models represent Command, Control, Communication and Intelligence (C^3I) implicitly. It means that C^3I is considered to be perfect. Orders are understood and carried out by subordinate command echelons without any mistakes. This is clearly in contradiction with reality as described in Sovereign and Coppieters [1989]. We consider the explicit modelling of the C^3I structure and processes to be paramount in the choice of a modelling technique to build war gaming models that provide a good correspondence with reality. Therefore the approach to develop war games must provide the necessary concepts for the modelling and implementation of decision making at the subordinate command echelons.

It seems that current models can only represent the C^3I process and more specifically Command and Control (C^2) in a realistic manner if these functions are carried
out by people. The U.S. Department of Defense [1988] has assessed C³I in military games and simulations and confirms this impression when the authors state:

"C³I is generally recognized as one of the most difficult areas to portray.... some of the burden is removed from the model designer when the decision is made to allow the player to perform the command and control functions directly.... Simulations surveyed for this project have followed this trend by placing the responsibility for command and control of all gamed units on the players."

The trend referred to in this study leads to a situation which is very well characterised by Hogan and Mlght [1989]:

"Prior to the introduction of sophisticated bookkeeping and adjudication models, most war games did not have the capability to show differences among the various research issues (e.g. design of a Command and Control system or training objectives (D.C.)). Currently available models are so detailed that they require a host of support personnel and approximately five days to play 10 days of combat. This is a classic "Catch 22" situation."

Also shown here is the problem of the lengthy preparation and execution times required by current war games.

Preparation is lengthy because of the complexity of the models and the large amount of data that is required to define the setting of an experiment. Furthermore all participants must be instructed in the use of the model and the playing of the game because each model implements its own working procedures that do not mirror or correspond to procedures generally applied by the military.

Execution times, the actual time it takes to play a game, are lengthy because the subordinate decisions that are not the goal of the game must be taken by human decision-makers. This generally implies that the minimal time resolution is real-time and often longer. This is caused by the complexity of the model and all the aspects that must be considered to simulate reality correctly.
These authors introduce the current trend toward an ever increasing level of detail. Wood [1987] proposes the following explanation for this phenomenon:

"there is always a tendency for them [the models] to become more complex as the years pass, in attempts to achieve greater realism, incorporate more aspects of the battle, and as a result of the games being used in a series of studies on different systems.... they (these tendencies) lead to a continuous growth in complexity, because, once having been added, a factor is hardly ever removed. This represents an abdication of responsibility by the analyst."

So, adding more detail suggests that a model is more realistic but in fact hides the inability to introduce really important aspects such as C^3. Furthermore it tends to make analysis of the gaming results a lot more difficult because of its complexity. Wood logically arrives at a similar conclusion when he states:

"It is therefore a mistake to think that we can, by going into more and more detail and representing more and more factors, construct a model that we can regard and use as a test-bed for World War III.... The fidelity of the real world, desire for which is the force that drives some analysts towards increasing detail, is a mirage."

We may therefore say that it is less important to model every possible detail of the battle than to discern the important aspects and model these in a satisfactory manner. Reduction is the key factor in the modelling effort and as the above references have shown C^3 should be included in the model and described realistically. However as Wood correctly remarks, this is a very difficult task:

"If the conclusions of a study are to have any claim to general validity it is essential that the study take into account of a range of assumptions concerning the major factors determining the outcome of the model. It is also necessary in the case of models incorporating Monte Carlo processes to run them a sufficient number of times to bring out variations in output that result from these processes. These two factors make it highly desirable to have faster-running models to extend what has been learned from games.

This has led to the use of computer simulations and of analytic models such as Lanchester's equations. Such models have become established as the workhorses of OA (Operational Analysis) and their predictions of absolute or relative
battlefield performances form the basis of perhaps the majority of studies. Two points on the relationship between these models and games are important. First, although they are in general faster than equivalent games, even the fastest is unable to explore the full or even an adequate range of uncertainties in scenarios and other assumptions. Second, whereas it may be reasonable to claim that we can write acceptable decision rules to cover short periods of low-level combat, it is extremely difficult to believe that we can do the same for the much more complex situation represented by high-level operations covering hours or days."

It would therefore seem that current practice in military gaming and modelling is forced to incorporate human decision makers. Farrell & al. [1987] describe in a concise and rather complete manner the problems related to human participation:

"(1) the variability introduced by the gamers (including learning effects, gamer biases, random variation, etc.), which is typically very difficult to separate from the systematic differences in results which are the intended subject of analysis; some experimentation has shown this variability to be greater than that introduced by random processes in the physical events of combat.

(2) the expense of gaming, which tends to reduce the number of replicates which can be conducted, exacerbating the problems of analysing the results described above; ...

(4) the difficulty in ex-post analysis of cause and effect when there is little or no record of why gamers made decisions or took actions, just as a record of what actions or decisions were taken."

The apparent paradox of having to integrate people in the gaming process to achieve relevant results which leads to the inability to do just that because of the problems cited above, leads to a problem situation which Wood [1987] characterises as:

"All this seems very obvious, so obvious that it is surprising that one frequently comes across claims that a game (or other model) has demonstrated that with a given weapon mix and concept of operations, a given NATO corps can defeat a given threat in so many days (or is defeated by it, as the case may be) without any hint of the level of confidence that can be attached to that prediction."

An intelligent gaming environment
It appears that the problems reported above are linked to the fact that many war
games apply an optimising technique to determine which course of action a
simulated unit should take. As shown by Melvin Dresher [1961] the MINIMAX
theorem can be applied to war games with both a finite and infinite number of
courses of action. It is then shown that an optimal solution exists for all games having
a finite number of courses of action and that those having an infinite number of
courses of action can have an optimal solution. However the solution to these
problems is based on the assumption that a payoff function exists through which
the payoff for a given strategy is uniquely computed. This contrasts with the
paradigm of bounded rationality which questions such a payoff function in absolute
terms. Furthermore, the definition of structured problems indicates that the payoff
function only exists for this class of problems in absolute terms. For ill-structured
problems the set of alternatives is not finite. As indicated in paragraph 2.2, most
military problems are ill-structured due to the inherent incompleteness and uncer-
tainty of the information that is used to solve them and because of the dynamics of
the military environment. Therefore the corresponding payoff functions may or may
not exist.

Summarising, it may be concluded that current closed or semi-closed models have
an unacceptable degree of correspondence with reality because they simulate
decision making at the lower echelons in the manner described above or do not
simulate it at all. In the latter case lower echelon units are considered to perform
their decision-making tasks perfectly.

It might be argued that open simulation models, e.g. models that incorporate human
decision makers at every command echelon, alleviate the problem of modelling
ill-structured decision-making. However, as discussed extensively above e.g. Farrell
[1987], human participation introduces as many problems as it solves. Furthermore
it transgresses the practical applicability factor defined in our evaluation framework
in paragraph 2.3. In fact, all command echelons have to be played by human
operators even when this is not the objective of the experiment.

We call this the realistic Command and Control (C\textsuperscript{2}) dilemma. Namely, current war
gaming models lack correspondence with reality because they do not model the
C\textsuperscript{2} process of subordinate echelons. Therefore they require human players to
participate in the war games. However, players are inherently variable in their
decisions and expensive to use.
We may consequently say that models, based on purely deductive reasoning e.g. based on the hypothesis that a payoff function exists for every problem situation that is solved by command echelons, do not provide us with a sufficient measure of correspondence to base military gaming on. Furthermore it has become evident that models which attempt to circumvent the correspondence problem are not cost-effective.

Farrell & al. [1987] have shown that the realistic C² dilemma can be overcome in a limited problem environment by using models that simulate the decision makers using rule based techniques.

2.5 Modelling based on the paradigm of bounded rationality

The paradigm of bounded rationality can be expressed in a declarative modelling approach which considers a number of characteristic problem situations. A basic concept for the description of problem situations is the notion of objects. Objects have attributes that describe their state and actions or methods that portray their behaviour. These situations include the environment of the decision maker and the objects that act upon the decisions. No attempt is made to model an optimising decision making process but rather a process that tends to satisfy given objectives. Therefore, applying this approach to the building of an empirical model of the problem situation perceived in reality e.g. the war gaming problem, should allow us to describe a model that provides a good degree of correspondence with reality.

It has been shown by Faught, Klahr and Martins [1980], Klahr, Faught and Martins [1980] and McArthur and Sowizral [1981] that object-oriented models facilitates the organisation and expression of expert knowledge. Furthermore an object-oriented approach provides the tool for the development of highly flexible and credible simulations. We associate the ability to organise and express expert knowledge with the modelling of ill-structured problems. This ability is further described in Klahr, McArthur, Narain and Best [1982]. They advocate the following modelling principles:

*The appropriate decomposition principle:* Select a level of decomposition into objects that is 'natural' and at a level of detail commensurate with the goals and purposes of the model.
The appropriate knowledge principle: Try to embed in your objects only legitimate knowledge, i.e. knowledge that can be directly accessed by the real-world objects that are being modelled.

The proposed principles seem to be relevant and effective because they guide a reduction of reality which enhances the correspondence between the real system under study and the model thereof.

However as stated by the afore mentioned authors [1982], the set of object types, they defined, provides insufficient basis for the modelling of non-intentional events e.g. an airplane being unwantedly monitored by a radar. Therefore the above mentioned authors have introduced the following objects:

1. The Scheduler: may be interpreted as an omni-scient, god-like being which given current information, anticipates non-intentional events in the future and informs the appropriate objects as to their occurrence.

2. The Physicist: accounts for the effects of physical phenomena such as bomb explosions and ecm (electronic counter measures).

3. The Mathematician: executes all complex mathematical computations. With its help, it is possible to remove mathematical details from the behaviours of objects, producing particularly readable code, e.g.,

   (ask mathematician determine time and position of interception of the fighter with the penetrator)

This shows that the fighter and penetrator which are both aircraft do not have to know e.g. must not incorporate in their behavioural description, how the interception time and location are computed. It is hidden from these objects.

However, the object types that have been introduced are too imprecise to allow a transparent modelling of the environment surrounding the part of reality being modelled. This is due to the fact that the objects that are responsible for non-intentional events and also for the actual evaluation of these events are environmental objects. Terrain or time, for example, or its influence, cannot satisfactorily be represented using the object types described above. The reduction of reality is
insufficient to show in a clear and credible manner the effect of the environment on the actions of objects.

Furthermore the evaluation of actions is still fully modelled using a purely formal framework. Using the scheduler, physicist and mathematician objects implies that the interaction between objects is described globally and cannot be accepted for the same reasons as described in the previous paragraph.

Finally, no models that address the complete battlefield have so far been developed using an object-oriented approach. So, the applicability of this approach to relevant multi-purpose large scale war gaming environments remains to be proven.

In order to provide a good level of correspondence when modelling complex sub-systems of reality e.g. military reality, we define the following concepts, based on Sol [1982]:

**object**: this is the basic construct of a model, it is described by:

- state: this represents the value of all the characteristics of an object at a given moment in time;

- action: an action has two components:

  -- event part: this can be a message containing specific information or a significant change of state;

  -- consequent part: passes a message to another object or initiates an execution based on the analysis of the event part;

- evaluation action: this type of action evaluates the result of the actions of objects based on four aspects. The first aspect is the state of the objects that are concerned by the action. Secondly, situation rules are applied. These are rules that have an influence on the action based on the state of the environment objects involved. Thirdly, execution rules are used to determine the result of an action. Fourthly, duration rules determine the duration of an action.

An intelligent gaming environment
An evaluation action consists of:

- an event: action of one or more execution objects;
- a consequent: judgement or calculation of the result of the event and change in object's state.

So, similarly to reality an object in our model is in a certain state. It can perform a certain number of actions and it is subject to outside influences.

**complex system**: open system in which ill-structured problems occur;

**simulator object**: object that manages all the objects in the model;

**decision object**: object that manages resources and uses them to try and achieve certain objectives; we distinguish between:

- human-driven decision object: human participants in the game;
- computer-driven decision object also called decision automaton;

**support object**: object that supports decision objects in each step of the decision making process that is performed by military decision makers:

- information gathering: the process of assembling the necessary data relevant to the solving of the problem;
- situation assessment: the process of data manipulation which produces information about a limited number of aspects of that part of reality that is of interest;
- option generation: the process of determining several courses of action that fulfil the objectives of the decision object, based on the situation assessment;
- decision making: the process of evaluation and choice between the generated options;

**control object**: object that monitors resources and environment to determine events defined by the decision objects and report these to decision objects;

An intelligent gaming environment
**execution object**: object that has a direct impact on the state of the environment or objects and can report the result of its actions to decision objects;

**environment object**: object that can have an influence on all other objects and upon which other objects can exert no or very few actions which require a considerable amount of scarce resources.

We expect that these concepts allow us to describe the entire C2 process and the battle execution. Decision support and control objects can be used to model the C2 and intelligence process. The environment and execution objects provide the vehicle to model the battle execution process.

### 2.6 Artificial intelligence

Paragraph 2.4 stressed that one of the main problems in current war gaming is the modelling of military decision making processes.

We argued that this problem is caused by the attempt to model these processes in a purely formal deductive manner and that on the contrary the modelling should be based on bounded rationality. This hypothesis leads us to search for other methods than purely mathematical ones to represent decision making. Modelling decision making relates directly to the representation of the knowledge one has and the way this knowledge is applied to decision making and problem solving. The field of artificial intelligence which is currently displaying a growing ability to represent knowledge and solve problems has contributed to the definition of concepts to solve this problem as illustrated by Farrell [1987]:

"The simulated commanders (expert systems) which were developed were quite complex and realistic, dealing with both the usual combat decisions implicit in any campaign simulation and such problems as: (1) the selection of transportation modes for a demand ..., (2) route selection, (3) the generation of transport demand for the movement of forces."

Since the field and scope of artificial intelligence is rather vague and ill-defined, it is deemed necessary to specify it more closely by considering a number of general definitions, providing our own synthesis of the field and succinctly describing the concepts commonly used to implement intelligent behaviour into a computer-executable model.

An intelligent gaming environment
Many definitions of artificial intelligence have been given e.g. Minsky [1977]:

Artificial intelligence is the science of making machines do things that would require intelligence if done by man.

Buchanan and Feigenbaum [1982]:

Artificial intelligence research is that part of computer science that investigates symbolic, non-algorithmic reasoning processes and the representation of symbolic knowledge for use in machine intelligence.

We consider both previous definitions to be unsatisfactory because they do not define the concept of "machine intelligence" and because intelligence is defined in such a broad manner that too many science applications qualify as being intelligent. We would therefore like to integrate these definitions and specify artificial intelligence as being:

that part of computer science that combines symbolic nonalgorithmic reasoning processes, symbolic knowledge representation and algorithmic reasoning processes to make machines do things that would require intelligence if performed by man.

Indeed, it seems that algorithmic reasoning processes can be combined with newly developed nonalgorithmic reasoning processes in order to build models that have a satisfactory degree of correspondence with real life decision making processes.

Current simulation models built using the paradigm of bounded rationality have applied such tools as Simula. Sol [1982] showed that the definition of the simulation language Simula as introduced by Dahl et al. [1970] supports the application of the paradigm of bounded rationality through its class concept.

Other languages such as Smalltalk, ROSS e.a. have been defined using similar concepts. These languages are currently referred to as object-oriented languages. An object in these languages consists of an attribute and an action part which is similar to the entity concept [Sol 1982, p.26-27].

As stated in paragraph 2.3, Faught, Klahr and Martins [1980], Klahr, Faught and Martins [1980] and McArthur and Sowizral [1981] showed that expert knowledge

An intelligent gaming environment
can be organised and expressed in a satisfactory manner using object-oriented languages. We therefore hypothesize that object oriented languages provide the appropriate tool to build war gaming models based on the paradigm of bounded rationality because they are capable of joining the modelling of decision making with dynamic execution. However the correspondence of the war gaming models built using object oriented languages depends on the framework of concepts used to model reality.

Furthermore, it should be made clear that a war game is an information system and should therefore be developed as such. A similar point is also argued by Elzas [1986, p.21] when he writes: "On knowledge acquisition methodology the modelling and simulation community could profit handsomely from cooperation, not only with the Artificial Intelligence field, but also with the much larger group of persons engaged in General Information Systems Research." Although Elzas advocates cooperation, he seems to introduce a distinction between modelling and simulation and the development of Information Systems. However we believe that Sol [1982] showed clearly that this distinction should not be made. The same applies to the artificial intelligence techniques like knowledge acquisition and representation which can be applied to the modelling and simulation field.

Generally we want to stress that the war gaming modelling and simulation community should not segregate itself from the Information Systems community but should acknowledge the developments in the field of informations systems methodologies and declarative problem solving as applied in artificial intelligence based information systems.

Furthermore as described in Chapter 1 war gaming models should contain the ability to simulate decision-making, Command, Control, Communication and Intelligence (C³I) between decision making groups i.e. staffs. It is our contention that this is only possible if these staffs are simulated as independent objects with specific behaviour patterns. Let us therefore look at the representation concepts commonly used in Artificial Intelligence that allow us to achieve exactly that.

As discussed by Elzas about the applicability of AI techniques to knowledge representation in modelling and simulation, the following techniques are useful in the case of war gaming. We will discuss them briefly and point out the ones that seem most appropriate for the modelling of military complex decision making.

An intelligent gaming environment
a. **classes**: a class is a set of attributes and has a set of related rules and demons that related to it. Together they form a logical whole since the attributes describe the class and the rules and demons its possible operations e.g. its behaviour. Concepts may be structured in specialisation trees where each level down adds specific attributes and/or rules or demons and may inherit those from higher level classes.

b. **rules**: production rules are derived from propositional logic and have a general "If ... Then ... " structure. Combined with a limited class definition i.e. classes only have attributes, powerful rule based systems have been defined. Furthermore inheritance is maintained so that rules applying to higher level classes also apply to descendants of that class and specific rules only apply to those classes. As such specialisation of knowledge can be represented. However production rules must be triggered by some goal seeking process which is not always a natural way of describing a reasoning process. This conclusion driven reasoning process is known as backward chaining.

c. **demons**: this construct has a "when ... then ..." structure. Its fundamental difference with rules is that it is triggered as soon as its condition part is fulfilled i.e. it monitors the status of the classes for which it is defined and executes as soon as any instance(s) of a class meet the conditions. This condition- or data-driven reasoning process is known as forward chaining.

Based on the previous descriptions, we put forward that applying a combination of these techniques will allow us to explicitly model the C3I structure and decision making processes.
2.7 Problem definition and research questions

As shown in this chapter, models based on a purely formal deductive problem-solving process often start from unbounded rationality and therefore do not provide a satisfactory measure of correspondence for the simulation of complex sub-systems of reality. The main problem lies in the reduction of reality and the decision making process that is assumed by these models.

Current applications of frameworks for developing war games display the potential to solve these problems but have not yet been applied to large complex battle simulations.

The previous discussion can be summarised by stating the following two main problems war gaming models currently face:

a. the lack of experimental validity;

b. the high percentage (40-60%) of participants that are required to take the subordinate level complex decisions in a high level war gaming exercise that is aimed at the rest of the participants namely the players, the lengthy preparation and execution times that accompany this.

Problem a stresses the methodological aspect of the war gaming effectiveness. On one hand, open models do not allow an experiment to be reproduced exactly because they involve human decision makers. These decision makers change their decision making process which is based on their real life experience in order to improve the results they can obtain by exploiting the features of the models. Therefore this human learning process impedes the use of open war games to produce statistically valid results. On the other hand, closed models are often criticised for their lack of military realism because decision-making at complex subordinate command echelons is not modelled adequately as the C^{3}I structure and processes are considered perfect as was shown earlier. Furthermore, the surrounding real world is simplistically modelled which leads to an insufficient correspondence with reality.

Problem b addresses the organisational problem of having to commit a large group of people (most war games include 200 participants or more). Half of these people are required to perform the simulation because they must make the necessary
subordinate command echelon decisions that are not made by the players of the game. Indeed the simulation model requires decision making at command echelons that are not the objective of the game. This poses the problem of cost-effectiveness and flexibility because this amount of highly qualified decision makers cannot be made available regularly to go through different scenario’s. This is further compounded by the often lengthy preparation and execution times that are necessary to define a war game's initial situation. Typically a war game that simulates 4 days of war will take several weeks of preparation and 8 days to play.

These problems are largely due to the inability of current war gaming development approaches to provide a framework that achieves the correspondence with reality that is required in a multi-purpose war gaming model.

Summarising, we have defined the boundaries of the problem domain in paragraph 2.5. We identified a paradigm which we use as a basis for building empirical models that provide a good degree of correspondence with reality i.e. the paradigm of bounded rationality using object oriented modelling.

It has also become apparent that the field of war gaming combines aspects from simulation, artificial intelligence and highly interactive systems since war games are semi-open simulations. Each of these fields have been researched and present a number of methodologies, development environments and software tools. However it is not clear which methodologies, development environments and tools are appropriate to support the integration of these three distinct fields in order to build war gaming models that present a high degree of correspondence with reality. This leads to the formulation of three main research issues:

1. Define a methodology for the development of war gaming models that present a high degree of correspondence with reality.

2. Define a development environment that supports this methodology.

3. Design the tools such an environment should offer.
The methodological aspect of the research concentrates on the following issues:

1. a way of thinking: which concepts should be used;

2. a way of modelling: how are the concepts translated into the constructs that are supported by the tools which are being used;

3. a way of working: how should the development of a war gaming model be planned;

4. a way of managing: how should the development be managed.

2.8 Theory

In order to solve the problems described previously and provide an answer to our research questions, we present a theoretical framework that consists of four main points:

i. The research will investigate whether the concepts described in 2.5, namely simulator object, decision object, support object, control object, execution object, environment object, are sufficient to model complex sub-systems of reality for war gaming purposes which provide a good measure of correspondence with reality.

ii. In order to develop and build decision objects the research will investigate whether the model of the decision-making process shown in figure 2.2 can be used. We examine whether it is a sufficient vehicle to model the process of solving ill-structured problems.

This model consists of two major components:

- the MMR (Model of Military Reality), previously described in chapter 1, is the view of the world that is accessible to decision makers;

- the model of the decision maker's or group's decision-making process which is derived from the problem solving process introduced in 2.1. It is represented by the four steps, drawn as boxes, that can be supported by support objects.
The model shows that a distinction should be made between two flows:

- the problem definition flow: this flow has its source in a decision making problem that requires a number of options to choose from; in order to provide these options the current state of that part of reality which is of interest must be assessed; the need for a situation assessment is translated into a need for data from the MMR;

- the problem solving flow: this flow shows that crude MMR data is used to assess the current situation; this assessment allows a number of options to be generated and these options are used to make a decision.
Bots [1989] introduces the concept of a task corresponding to an activity in a problem-solving process. He states that a task can be divided into sub-tasks. Within the overall task, a precedence between sub-tasks can be defined and the results of sub-tasks must be coordinated. Similarly, we would like to model a problem-solving process by distinguishing its different steps. However instead of focusing on the activities themselves, we rather look at the knowledge that is used to perform them. So, we suggest that decision objects i.e. objects that make decisions about the use of resources, can be structured by applying the following modelling principles:

a. **The knowledge structuring principle**: conceptually, a problem domain should be structured by defining sub-problem domains through the process of dividing the knowledge that is applied to solve the problem into independent knowledge fields.

b. **The communication principle**: support objects suggest a solution to problems in a sequence defined by their precedence in the decision-making process. We can define a decision precedence which illustrates the decision making order. However, decision-making is an iterative process which means that in certain cases part of the decision making sequence must be repeated due to unsolvable problems in the lower branches of the decision precedence tree. Therefore the communication between decision and support object must be bi-directional. We suggest that a decision threshold mechanism be used which forces preceding decision or decision support objects to adapt their conclusions to the constraints of following support objects.

Complex decision-making in the military realm entails the specification and choice between a number of options that use heterogeneous complementary resources to solve the same problem. In other words, decision making is complex when the resources that can be applied to solve the problem have different characteristics and can replace each other for certain parts of the problem. So, several resource combinations may solve the problem and present specific disadvantages and advantages. However, there is no clear framework to evaluate each combination and thus aid in the choice process. Therefore we suggest a modelling approach which concentrates on two specific aspects of the decision-making process.

The first aspect is resource specific. Each resource has specific characteristics and application possibilities and particular knowledge is applied to determine its use within the scope of a problem. So, this resource specific knowledge is independent.

An intelligent gaming environment
from knowledge that applies to other domains and should be modelled separately in order to structure the decision-making process in a manageable manner. This aspect corresponds to the knowledge structuring principle.

The second aspect concentrates on the coordination of resource specific decision-making processes. Indeed, to define resource combinations, it becomes clear that the results of applying independent knowledge to a problem must be coordinated to determine a coherent and consistent overall solution. The communication principle presents the mechanism we suggest to achieve this.

These principles are directly linked to the following definitions we use to categorise decisions:

a. **the command decision**: this type of decision specifies what an object’s mission is and how it will be achieved using operational resources e.g. resources that have a direct effect on objects that compete for the fulfilling of the mission.

b. **the support decision**: this type of decision specifies how support resources are applied to enable the implementation of the command decision to which the support decision is linked.

These two types of decisions define the relations that can exist between decisions namely:

a. **equivalence relation**: the fact that a pair of command decisions influence the overall decision in the same manner defines an equivalence relation between these decisions.

b. **support relation**: a support decision is linked to a command decision and the support decision can only influence the command decision if certain thresholds are reached; when this occurs, the command decision must take this into account and must consider this as a supplementary constraint in its decision making process.

iii. The research investigates whether the relations between objects defined in 2.5 and illustrated in the object-relationship diagram in figure 2.3 and their specialisations are sufficient to build war gaming models that offer a good degree of correspondence with reality.
Figure 2.3: The conceptual object-relationship model

An intelligent gaming environment
iv. The research attempts to show that war gaming simulations based on the principles defined in the hypotheses 1 to 3 have the capability to train people to handle that part of reality that has been modelled with a good measure of correspondence if:

- players are decision objects;

- player decisions are enhanced by support objects for every step of the decision making process.

In the next chapter, we introduce the case that is used to test this theory. The way of modelling, working and managing will be illustrated in chapters 4 through 6.
Chapter 3: Case description

3.1 A problem situation

The Royal Belgian Defense College (RBDC) teaches three courses to Belgian and foreign officers:

- Staff Techniques course: 5 week course at battalion echelon, called the first cycle;

- Higher Officer Training course: 7 month course at battalion and brigade echelon, known as the second cycle;

- Higher Staff Training course: 10 month course at division, corps and NATO command echelon, known as the third cycle.

These courses are aimed at all Belgian officers but focus in particular on the air and land forces.

3.2 Current situation

The situation referred to as current in the rest of this chapter, is the situation that was used until 1989 and which is described in Coppieters [1987]. The current war gaming support environment consists of three main applications:

- the TELEBATTLE simulation model which runs on a DEC 20-40 and is written in FORTRAN 4 and 77, Pascal and PCL, a machine dependent command language;

- a decision support application for Army players which runs on Apple Macintosh and IIC machines;

- an airforce information dissemination system which runs on Personal Computers (PC) and is written in DBase III Plus.

Each of these application is further analysed in the following paragraphs.
Application and system software

The decision support application for Army players consists of Basic programs and spreadsheets developed using the AppleWorks integrated software package. The maintenance of this application is very labor-intensive because of the limited readability of the code and poor documentation.

The structure of the TELEBATTLE model is shown in figure 3.1. It consists of 4 main modules. The first, called WARGM1, constitutes the user interface which analyses the syntactic validity of the orders that are provided by the users. Orders are stored in a user-specific file and are processed every game cycle by the second module. WARGM2 consists of 7 sub-modules which each address a specific combat aspect. Units are described in each functional module and evaluated accordingly. The next step preparation module indicates that the TELEBATTLE model is a time-stepped simulation i.e. within a given period or time step, all interactions are supposed to be concurrent. The third module, WARGM3, prepares the hardcopy output for players and the relevant data to execute the logistical simulation. This data is processed by a link module which makes it suitable for the fourth module, the logistical simulation. The logistical situation is evaluated in a time-stepped manner in six hour cycles and provides hardcopy output about units to players.

The software structure is unclear due to an incomplete documentation of the logical structure of the program. The interactions within the model have therefore become untraceable. Numerous changes and extensions have only lead to an even worse situation. Indeed, unpredictable events take place e.g. supply trucks seem to lose part of their shipment en route without any discernible reason. Furthermore, the software is machine dependent since it relies partly on the DEC 20 command language. Another important aspect is the use of data files which are managed manually. This leads to a high frequency of mistakes and lengthy correction times.

Based on our experiences acquired during the war game carried out in February 1987 called TELEBATTLE'87, we make the following remarks.

Firstly, we noticed an important translation problem from orders to a machine executable form and it appeared that users are mostly inexperienced in the use of the computer equipment. This translation work is carried out by a group of officers who support the execution of the war game. They are referred to as the lower control staff.

An intelligent gaming environment
Figure 3.1: The TELEBATTLE model

An intelligent gaming environment
A second important remark is the time pressure that is put on the lower control whereas the Corps and Division staffs have too much spare time. This is in contradiction with the objective of the game. Indeed, the player staffs, e.g. corps, division and brigade should be put under stress. However at the moment, the lower control staff must translate orders from brigade staffs into data that is processable by the model and vice versa within 30 minutes. This 30 minute time frame is the simulation step which is divided into six 5 five minute time steps i.e. every 30 minutes the past 30 minutes are simulated in slices of 5 minutes. Within a single time step, 5 minutes, all events are considered concurrent.

More generally, the tactical aspect of the exercise suffered from:

- problems with the definition of the terrain especially on the borders between two terrain squares;

- an unrealistic modelling of obstacles. Obstacles cause a time delay and possibly some losses to a unit. However as soon as the delay has expired, the unit can overcome the obstacle without having to take any specific measures;

- the availability of perfect information to both sides about the other’s situation because the game map is accessible to the entire lower control staff i.e. a lower control staff exists for each higher echelon unit on both sides and reports about the situation based on the situation perceived from the game map. Since the game displays the real situation both sides receive perfect information;

- Insufficient personnel in the opposing force staff which makes for an unrealistic situation and many interventions from the game control staff to correct the situation;

- a lot of user mistakes due to insufficient training;

- the fact that electronic warfare and combat helicopters are not modelled;

- an information flood because the model produces too much data which is of little or no use to players.

The logistical function which simulates the supply, material and medical support activities, is not performed because it is incorrectly integrated with the tactical

An intelligent gaming environment
function. The logistical and tactical functions are not integrated in the same model but are actually two separate simulation models that pass information to each other. No dynamic interaction takes place e.g. logistical actions are not influenced by the battle itself, but only by its results. So, a supply convoy may drive through a battle area without ever suffering any damages. Furthermore, the initial logistical situation is always optimal which causes this function to work perfectly because of the limited duration of the exercise. The didactic objective of illustrating the importance of the logistical support can therefore not be achieved.

Finally, the air-land integration cannot be used effectively because of the manual interface between the land battle simulation model and the air battle simulation model. This situation leads to a very limited use of Close Air Support and Battlefield Air Interdiction.

The airforce simulation is carried out manually. A data base is used which allows players to introduce missions based on the accessible information about assets and targets. Missions are evaluated by a team of teachers who update the situation and provide reports about the mission's results. The information flow between staffs is well supported but the simulation of the evolution of the situation is insufficiently supported. This is due to the manual simulation of missions. Therefore the current airforce simulation application places a heavy burden upon the game control staff which also plays all red operations.

**Hardware capacity and operations**

Hardware capacity and availability is a major problem for the RBDC because it does not own the means to execute the TELEBATTLE model. The RBDC is permitted to use all the computer equipment of the Royal Military Academy (RMA) for 1 week per year. This restricts the availability of the equipment and thus of the war gaming model to 1 week a year. It also causes lengthy preparation times because all the terminals must be moved from the RMA to the RBDC. There, they are installed in a classroom that is not suited for this purpose which causes numerous defects. Furthermore during the execution of the game, only one hard copy facility is available at the RMA which sometimes leads to important delays. Also, an insufficient number of Apple computers is available for the transmission of information between staffs and for the execution of necessary tasks e.g. word processing. Furthermore the equipment is obsolete and impedes the development of effective decision support systems.
Model development and maintenance

As can be deduced from the previous paragraphs, the maintenance of the model is difficult and labor-intensive. Furthermore, we can question the integrity of the TELEBATTLE system because it has often been adapted by different people and not been documented.

Development is therefore restricted to the air force model which will serve as a war gaming support tool until the new war gaming environment is available.

Conclusions and recommendations made to the RBDC

The conclusions that can be drawn from the previous description of the RBDC’s war gaming environment and the recommendations made at the start of the project, i.e. march 1987, are described below.

Generally we may conclude that the land game insufficiently models the necessary battle factors and does not satisfactorily support the air-land integration. Furthermore, the decision pressure is put at too low a level, namely, the lower control. The game is not aimed at these people. We may therefore conclude that the goal of the game is only partially reached. We suggest that all manual operations at the lower control be suppressed. This is not possible with the current model.

The air force game provides good support for the information flow but lacks support for the simulation in time of the interaction between both opposing sides.

Therefore we recommended that the land game be completely renewed and that the air force game be used as a basis for further developments. The coordination and integration between both models should be completely described and implemented.

An intelligent gaming environment
3.3 Main war gaming goals and strategies

Based on these didactic goals, the RBDC has defined the war gaming goals per cycle.

The first cycle does not require any war gaming support because of its limited duration and its focus on staff techniques e.g. military symbology.

The second cycle, on the other hand, requires war gaming support to provide training in the use of procedures at the brigade, division, ALO (Air Liaison Officer), ASOC (Air Support Operations Centre), ATOC (Air Tactical Operations Centre) and SOC (Sector Operations Centre) echelons. Devising orders at each of these command echelons involves the use of a method of thinking, e.g. identify the factors that are important in the military decision-making process, and working, e.g. the structure and sequence of the decision-making process. The aim of the war gaming exercises is to support the learning of the military method of thinking and working.

The third cycle focuses even more on the actual effectiveness of the military reasoning process. Therefore it requires war gaming support for the training of the appreciation and decision making process at corps, division, ALO, ATOC, SOC, ASOC and ATAF (Allied Tactical Air Force) echelons.

It is required that the resulting warfare simulation environment can be enhanced to support:

- the correction of manoeuvre plans submitted by the students;
- the process of defining and choosing operational and logistical concepts and the evaluation of new weapon systems;
- the training of the brigade, division and corps staffs of the 1(BE) Corps, the first Belgian Corps.

In order to realize these didactic goals the war gaming environment should fulfil the requirements described below.

In contrast with the current gaming environment TELEBATTLE, the new environment should allow a more frequent use of the war gaming facility. It is aimed at being
a tool to support courses in a repeated manner e.g. support each exercise that is part of a course and illustrate the different military problems e.g. planning and battle management.

In order to provide this repeated support, a major objective is to achieve the integration of the air and land battle which is essential in providing a realistic view of the battlefield. Furthermore, it is also required that the number of participants in a war game be reduced to the actual player staffs. This implies that support staffs are replaced by part of the gaming model. Therefore the gaming model will have to incorporate components that realistically replace the support staffs e.g. these components should replace the support staff in a manner that is transparent to the player staffs. An other important aspect in providing this continuous support is the rapid preparation of games which should allow the teachers to quickly set up a war game that illustrates their courses.

Finally, the gaming model should allow the possibility for post-analysis to illustrate the impact of certain decisions in time and show the sensitivity of military reality to battle management decisions.

Having stated the war gaming requirements, we clarify the military reality that is relevant for the courses of the RBDC. This sub-system of reality, described in figure 3.2, must be modelled in the simulation model. Figure 3.2 illustrates the command echelons that are within the scope of the RBDC’s courses and the information flows between the different command echelons involved. The diagrams have been designed in English. However the common Belgian abbreviations have been used to denote the echelons.

The Belgian army consists of 1 corps, the 1(BE) Corps, which can be placed under a NATO command in case of conflict namely, NORTHAG i.e. Northern Army Group. Within NORTHAG, the 1(BE) Corps must cooperate with other corps and is dependent for its logistical support on the Belgian government. A corps consists of a number of divisions and has specific assets which are called corps troops e.g. artillery battalion (Ale), Anti-Tank battalion (ATk), engineering battalion (Gn), helicopter battalion (Lt Avn), transmission battalion (TTr), light armoured battalion (Recce), hospitals (Med) and logistical installations (Log). A division (Div) comprises a number of brigades (Bde) and may have specific troops also. A brigade consists of a number of battalions (Bn) and companies (Cie).
Figure 3.2: Military Reality

An intelligent gaming environment
The Belgian airforce is completely allocated to NATO. Allied Air Force Command Europe (AAFCE) directs airforce operations and consists of a number of Air Tactical Air Forces (ATAF) which are responsible for a certain area. An ATAF commands a number of Air Tactical Operations Centres (ATOC) and Sector Operations Centres (SOC) which plan offensive and defensive operations. Missions are carried out by the WING's.

The army and airforce coordinate at corps, ATAF echelon trough the Air Support Operations Centre (ASOC).

The RBDC [1987] has defined the following two main types of war game which it would like to perform and which serve as the basis for the research project:

1. Air War Game:

   The war game represents an air battle between a blue and a red airforce.

   - the third and second cycle Air Force students only play the blue team;

   - the red team is played by teachers with increased automated support to perform the subordinate echelon's decision making;

   - more than one blue team should be able to play simultaneously against the red team;

   - air-land integration is provided through a team of Army teachers and ground operations information from previous land war games or from a simultaneous land war game.

2. Army Corps War Game:

   The Army Corps war game can be played in two different ways:

   1. Corps and Division game aimed at the Army third cycle students;

   2. Division and Brigade game aimed at the Army second cycle students.

An intelligent gaming environment
Integration is provided through the ASOC and ALO echelons played by the Air Force second and third cycle students and the air operations information from previous air war games or from a simultaneous air war game.

The design and implementation of a new war gaming facility for the Defense College will provide a tool to make more efficient use of the available time than is now possible because the number of participants in a game will be reduced to the students.

This will allow for a more effective training of students by providing them with a simulation of all realistic aspects of battle. Instead of only making plans and discussing each plan's flaws and merits, a dynamic evaluation of the plans can be performed. Furthermore a new aspect of the officer's function can be trained namely conducting the battle.

In a second stage, it is conceivable that the gaming environment could provide training for the actual Corps, Division and Brigade staffs.

It could also provide a basis for the further development of a game to evaluate weapon mixes, to carry out operational plan analysis and to develop and test advanced decision support systems.

This last remark supports the conclusion of chapter 1 which advocates the definition of a multi-purpose war gaming environment. Let us then consider how to achieve this goal.
3.4 War gaming development strategy

Development summary

Based on the model of reality that must be simulated and that is illustrated in figure 3.2 and the war gaming requirements expressed earlier, the global model of the war game we intend to develop is defined and is shown in figure 3.3. Double-lined objects may interact with all other objects.

It should be noted that the brigade echelon can be found within the player team and in the simulation model. This constitutes the novelty of our approach. The corresponding automated decision object can play at the brigade echelon. Thus eliminating the lower control function that is required in the TELEBATTLE system.

Similarly, either players or the corresponding automated decision object can play at the ATOC, SOC and WING echelons. It is our objective for the red air force team to reduce this to one person with all subordinate echelons being performed by automated decision objects. This enables us to use the model to define games at different echelons as required by the different cycles' didactic goals.

The model called Intelligent Air Land Training Application (IALTA) represents an open simulation of an air-land battle. Interaction with the simulation model will be possible at two levels:

- player level: the officers in training have only a limited and well defined interaction with the simulation model;

- control level: the teachers directing the game will have freedom of interaction with all simulation objects in order to fulfill the training goals.

The students will form the staffs at Corps, division and brigade echelons for the Army and ATAF, ATOC and SOC for the Air Force.

So, the IALTA simulation the environment allows us to play the following games:

- Air war game with the relevant land battle influences;

An intelligent gaming environment
Figure 3.3: The IALTA Model

An intelligent gaming environment
- Corps land war game: this game is played by the third cycle, the brigade is enacted by the simulation model;

- Staff procedure land war game: this game is played by the second cycle, the brigade staff is played manually.

In order to provide this flexibility, the simulation environment will contain automated decision objects at the brigade, corps and division troops and wing echelons. The future system has therefore been named the Intelligent Air Land Training Application (IALTA).

The IALTA system is composed of two sub-systems namely: the Intelligent Land Training Application (ILTA) and the Intelligent Air Training Application (IATA). This structure was adopted because the simulation of air and land combat is fundamentally different in its environmental and resource descriptions.

The ILTA system is a two team warfare simulation environment in which several blue teams can simultaneously play against one red team.

The IATA system is a blue team air warfare simulation system in which the red team is played by the Air Force teachers who provide high level directives to decision objects which simulate the red team. The resolution of both sides, e.g. the level of execution, will however be the same.

We will discuss these systems separately in the following paragraphs.

**The Intelligent Land Training Application**

Since people interact with the simulation model, the ILTA is an open simulation model in which players act at the Corps, Division and Brigade echelons for both sides. The simulation model will address the following four main military functions:

1. Tactics which entails the planning, execution and the evaluation of the consequences of orders.

2. Logistics which focuses on the setup of the logistic chain, the supply, material and medical support.

An intelligent gaming environment
3. Coordination is represented as the information flow between command echelons of one force and between the Army and the Air Force.

4. Planning is also addressed, it involves the preparation of future manoeuvres.

The environmental conditions model the cycle of night/day, the weather and the terrain for which there are different representations according to the echelon i.e. the level of detail is adapted to the command echelon. Certain relevant aspect systems will also be addressed namely Command, Control, Communication and Information (C³I) systems and the effects of Electronic Warfare (EW). Furthermore, the gaming environment will incorporate the influence of psychological aspects such as morale and stress on the operation of units and the decision-making process of decision objects. The resolution of the simulation, e.g. the lowest simulated echelons, is at battalion and company echelon.

Time contraction, the real time:simulation time ratio, is variable based on the cycle that is being trained and ranges between 1:1 and 1:8. The air-land integration is achieved by the representation of the ASOC and ALO echelons and by the common situation information that is provided by the simulation.

The Intelligent Air Training Application

The IATA system is an open simulation model in which the players act at the ATAF, ATOC and SOC echelons of the blue team. The red team is played by teachers at the same echelons but with increased automated support e.g. decision objects will be used to carry out the decision-making at ATOC, SOC and WING echelons. The simulation model supports the functions as described in the NATO directives. It represents the locations of airbases, the effects of weather and night/day, the influence of the land battle and the C³I system. The resolution of the IATA is mission level e.g. individual aircraft are not identified, they are considered as a global resource. Time contraction is variable between 1:1 and 1:8.

Air-land integration is provided here through the ASOC function and the available land battle information.

An intelligent gaming environment
3.4 Application development planning

Phasing goals and strategies

The goal of the proposed phasing is to structure the development in such a way that its feasibility is ensured and that the necessary resources for its implementation can be specified in a timely manner.

In order to achieve this, the following three phases are defined:

   Phase A: Decision automata development
   Phase B: Simulation model development
   Phase C: Player support development

We choose to start the design with the definition of the automated decision objects based upon the fact that these objects trigger the dynamic interaction between the other simulation objects. Defining these decision objects entails modelling the decision making process and the objects that play a part in the reasoning process. Therefore the modelling of these objects implies an early definition of the scope and resolution of the simulation i.e. the reduction of reality that is required. The modelling of the decision objects shows the level of detail required by the decision making process and the relevance of introducing details or reducing reality. This applies to objects that are directly controlled by the decision object and to objects that create the environment for the execution of orders generated by the decision object. Furthermore the modelling of the decision objects may lead to a better understanding of the problem area through the close interaction between design team and experts.

Expert participation is further exploited during the second main phase that entails the definition of the other objects except the decision support objects. The modelling of these objects implies another reduction of reality through the process of eliminating irrelevant objects. The relevance of objects is assessed in relation to the goal of the model. It may be clear that the knowledge acquired during the development of the decision objects is of great use for the development of the simulation model. The necessary execution, control and environment objects have already

An intelligent gaming environment
been identified before starting phase B. Therefore phase B need only concentrate on the dynamic aspect of the previously defined objects.

Finally, phase C will develop the decision support objects which will be used by the players and controllers of the war games. It may be expected that decision objects provide a good basis for the development of decision support objects. Furthermore, this sequence is adopted because the player support environment focuses primarily on the ease of use of the simulation model. Phase B defines the interfaces between decision objects and all other objects. This last phase merely uses these interfaces and enhances them with highly interactive tools that support players.

Implementation approach

An implementation plan has been defined per phase.

Phase A applies an iterative approach that comprises the following four steps.

Phase A.1 entails the description of the military decision making process at Brigade, ATOC, SOC and wing echelons e.g. those echelons that must represented by decision automata.

Phase A.2 determines the description of the information requirements per decision making process.

Phase A.3 identifies the decision rules that are applicable for each decision making process.

Phase A.4 entails the development of the decision automata.

Phase B follows an iterative approach which consists of the following steps.

We start, step B.1, by describing the objects involved in the simulation separately. Then in, step B.2, we determine their interactions. Based on the requirements of the gaming environment, in step B.3 a reduction of the object's description and interactions is conducted. A prototype, step B.4, is developed which has a suitable graphical interface to support the verification and validation.
The iterations should lead us to an acceptable system specification which will allow us to develop a production system with its graphical interface. In order to ensure the acceptance of the IALTA simulation model, we will perform a number of extensive tests.

Phase C applies an iterative approach which centres around a rapid identification of user needs followed by the development of a functional prototype. As soon as the prototype presents the required behaviour, a study will be made of the required technical enhancements and a production version of the player support environment will be developed.

![Phase Diagram]

Each square represents two months

*Figure 3.4: Phases in time*

An intelligent gaming environment
The execution of the phases is shown in figure 3.4.

In each of the following chapters, this phased approach is refined. A description of the actual course of action is also given. The combination of plan and reality should allow us to draw conclusions about the aspects that are important in each phase of the development of the war gaming environment.
Chapter 4: Developing decision objects

The previous chapter described the case. The first step in the development of the IALTA system is the development of decision objects that must be capable of replacing players at the Brigade, Wing, ATOC and SOC echelons. This chapter describes how the model of the decision-making process introduced in chapter 2 is used to design and build decision objects. More particularly, we focus on the action part of the decision objects. We describe the decision making process as a set of event-consequent combinations where the event is the occurrence of a problem and the consequent the activation of a mechanism that solves this problem.

A Brigade is responsible for a certain area of the terrain and uses a number of battalions and companies to achieve the objectives which are given by either the commanding division or corps.

A Wing is responsible for the routing and choice of armament to accomplish airforce missions given by the ATOC and SOC.

The ATOC and SOC echelons allocate their resources for respectively offensive and defensive operations that are derived from the directives given by the ATAF.

The selection of the command echelons mentioned above is based on the didactic objectives of the Defense College and the complexity of the associated decision making process. Indeed, the decision-making at these echelons is representative for the solution of ill-structured military problems because it focuses on the use of scarce complementary heterogeneous resources. Military units can employ different types of resources in different combinations to achieve the same objective. However, choosing one combination implies that certain combinations are no longer available to fulfil other missions. Thus, the military decision-making problem entails the actual understanding of the different resources that are available to solve a problem, the ability to define resource combinations that provide a valid solution and the choice of a combination that provides sufficient flexibility to meet the inherent uncertainty of any military operation.

We refer to figure 3.3 and the accompanying description for a general overview of the IALTA system and the role that decision objects play in it.
We start by giving a more detailed description of our modelling and structuring approach as related to decision objects. Subsequently, we discuss the structure of each of the decision objects mentioned above. In each description, we illustrate the use of the model of the decision-making process.

4.1 Modelling and structuring of decision objects

The knowledge structuring principle that we discussed in chapter 2, advocates the division of a problem domain into independent sub-domains. Therefore, we may distinguish between two types of decision object:

- complex decision object: this type of object calls upon a number of other decision objects to solve a problem;

- simple decision object: this type of object solves a specific problem and does not call upon any other decision object to define part of the problem.

Simple decision objects are defined by applying the knowledge structuring principle in an iterative manner. Complex decision objects employ and coordinate the results of decision objects to reach a solution to a problem.

We adopt the set of concepts described in chapter 2 to model the knowledge that is elicited from experts.

Each resource type that is used in the decision making process can be described as a class. Furthermore, a decision is also modelled as a class since it is a logical set of attributes. Special types of resources or decisions can be modelled by creating sub-classes. Attributes, rules and demons are added which are relevant for this class. In this way, we define a class hierarchy that is relevant to model the decision making actions of a decision object.

The reasoning sequence of the decision making process is described in the procedural body of a decision object. Rules and demons are triggered in a stepwise manner by defining the class and its attributes that must be determined. The actual inferencing mechanism, i.e. how a value for a class attribute is obtained, is not specified yet. Communication with other decision objects and other types of objects i.e. execution objects is handled using communication messages. 

An intelligent gaming environment
4.2 The brigade decision object

The goal of the brigade decision object is to realistically emulate the decisions of a brigade staff. A staff is composed of a commander, chief of staff, a personnel officer, an intelligence officer, a tactics officer, a logistics officer, a fire support officer, an engineering officer and a communications officer. When conducting a manoeuvre, the staff work together to define a manoeuvre plan which achieves the objectives given by the commanding division. This plan always comprises the following six aspects for every officer’s specific field:

- **attitude**: the type of mission being carried out e.g. offensive, defensive, delaying action;
- **objective**: the location where the mission should be completed;
- **disposition**: place of the units commanded by the brigade and their composition;
- **axis**: part of the terrain where the main effort lies;
- **rhythm**: the way the mission should evolve in time and in space e.g. an attack consisting of two phases with a secondary objective in location B;
- **mile stones**: a set of times upon which certain parts of the mission should be accomplished.

These six aspects are used to define the operations order that is derived from the manoeuvre plan and which is then circulated to the lower echelons and to the commanding echelon.

In the brigade model, we simplify reality by not considering the personnel problem. This is mainly a peace time problem and the brigade model focuses on combat decisions. Therefore the brigade as referred to from now on will be in a combat situation. The brigade model emulates the staff’s decision making in the following fields of expertise:

- **staff**: defining how a brigade moves to its initial operational location, the delay that is required to formulate a plan and the earliest time to start the execution of the brigade’s hour called H-Hour;
- **Intelligence**: the gathering of information about the enemy to determine his possible courses of action and the most likely one;

- **tactics**: the manner in which the available combat resources can be applied to achieve the brigade's mission;

- **logistics**: this comprises, the supply of all brigade units, the maintenance of all brigade vehicles and the medical support to all brigade personnel;

- **fire support**: this entails the use of artillery units, light aviation and close air support;

- **engineering**: attempts to apply resources to overcome or set up of obstacles;

- **communications**: ensure that the brigade headquarters are at all times in communication with their commanding and subordinate units.

The set of decision objects that correspond to each of these knowledge domains constitutes the model of the brigade. It clearly divides the brigade decision making into a number of specific knowledge fields as identified by using the knowledge structuring principle. This means that each function requires knowledge that is independent from any other function. Communication between functions is defined by the precedence of functions and the definition of thresholds. When a certain threshold is reached, the reasoning must recede in its reasoning sequence and attempt to reduce the mission it has given to the function which reported a threshold problem.

Following this discussion, the decision making action of the brigade decision object has been structured as shown in figure 4.1.

This diagram illustrates the precedence in the decision making process of a brigade staff. The different boxes within the decision making action of the brigade decision object, also called brigade decision automaton, represent the decision making actions of decision objects that correspond to the before mentioned brigade functions. The double lined boxes represent the other decision objects of the ILTA simulation model.

The staff decision object defines a brigade’s initial capability to carry out its operation, prepares it for the execution of its mission and defines the amount of time

An intelligent gaming environment
that is necessary to define the brigade's plan. As such the staff decision-making process is a crucial step in the entire brigade's decision making process because it defines the time that is available for the entire problem-solving process of the
brigade and of its subordinate units. It specifies the manner in which a brigade reaches the location to start its manoeuvre i.e. the route and timing of the movement. Also determined are the forces that participate in the movement.

The intelligence decision object addresses the information gathering and analysis problems. It is carried out continuously regardless of the brigade’s mission and sets priorities in order to support specific missions.

The tactical decision making process defines the plan to achieve the brigade’s operational mission.

The other decision making processes solve the necessary tactical support problems. They are geared towards the implementation of the tactical plan and merely fill in the details of their corresponding field of expertise. However each support automaton may restrict the implementation of the tactical plan. Indeed, support which is vital for the execution of the tactical mission e.g. the availability of mobile bridges, may be limited and sometimes insufficient. Encountering such an unsolvable support problem forces the entire brigade’s reasoning cycle to iterate i.e. the tactical automaton repeats its decision making process.

Clearly the brigade decision object is driven by the staff, intelligence and tactical decision objects. Using the taxonomy of decisions introduced in chapter 2, the staff, intelligence and tactical decisions are described as command decisions. All other decisions are support decisions.

The following paragraphs discuss each decision object separately.

4.3 The logistical decision automaton

The logistical problem consists of three independent problem areas and corresponding knowledge fields that can be derived from RBDC [1989], [1987], [1983], namely:

- the supply problem which investigates how to provide adequate supplies in food, fuel and ammunition to the brigade’s units;

- the maintenance problem which looks at the evacuation and repair of damaged wheeled or tracked vehicles;
Figure 4.2: The logistical decision object

An intelligent gaming environment
- the medical problem which addresses the evacuation of- and medical assistance to wounded personnel.

Figure 4.2 shows the structure of the logistical decision object. Applying the knowledge structuring principle, a separate decision object has been defined for each independent problem area.

Data flow diagrams are used to illustrate the reasoning process of the logistical decision maker. In this type of diagram data is represented by a square and reasoning steps are represented by rounded rectangles. Each data item has a unique identifier and a set of attributes. Each reasoning step is uniquely identified and is described using rules which fall into four categories namely information gathering, situation assessment, option generation and decision making rules. These categories are derived from the model of the decision making process as introduced in chapter 2.

4.3.1 A supply decision automaton

As described in RBDC [1982] on the role of the supply unit within a brigade, the supply problem consists of the definition of supply points and transport circuits to support the brigade’s consumption of food, fuel and ammunition. A single resource is used to solve this problem namely trucks. Therefore, all supply types compete for the allocation of trucks. The reasoning process is illustrated in figure 4.3.

In order to plan the supply transport requirement, a set of rules is applied that provide an estimate of the brigade’s consumption based on its composition and its mission e.g. a brigade that attacks has a fuel consumption of 1.5 times the standard fuel consumption unit per kilometre that must be covered. This set of rules constitutes the information gathering rules for the supply decision object. The rules that relate the estimated consumption to the number of required trucks, are the situation analysis rules e.g. a truck can carry 1500 rations, so a brigade of 6000 persons requires 4 trucks to transport its daily rations. These rules are based on the logistical assessment framework that is taught at the RBDC and which is described in RBDC [1982], [1983].

Based on the comparison between required and available transport capacity, a set of rules are triggered that suggest a number of solutions until a satisfactory distribution of resources is obtained or all rules are exhausted e.g. standard rations

An intelligent gaming environment
Figure 4.3: The brigade supply decision making process

An intelligent gaming environment
can be replaced by combat rations which reduces the number of required trucks for food by 25% because a truck can carry 25% more combat rations than normal rations.

Clearly the supply problem cannot be divided into any independent sub-problems. Therefore the supply decision automaton constitutes a basic, undividable component of the brigade decision automaton.

4.3.2 A maintenance decision automaton

The maintenance problem, as described in RBDC [1982] on the maintenance unit within a brigade, focuses on the choice of a location for repair installations and the evacuation of damaged vehicles to and from these installations. The maintenance decision making process is shown in figure 4.4.

In order to plan the maintenance transport requirement, a set of rules are applied that provide an estimate of the brigade's damages based on its composition and its mission e.g. a brigade that attacks a well prepared enemy may expect to have 5% slightly damaged tracked vehicles. This set of rules constitutes the information gathering rules for the maintenance decision object. The rules that relate the estimated damages to the number of required tank transporters or tow trucks are part of the situation analysis rules e.g. a tank transporter can carry 2 tracked vehicles and moves at a speed of 20 km/hour, in attack the mean distance between the front and the repair installation is 25 km, so it takes 2.5 hours to transport 2 tracked vehicles; a tank transporter works for 12 hours, so the number of required tank transporters is ((estimated damages/2)*2.5)/12. These rules are completed by a set of rules which estimate the repair capability of the brigade installations e.g. it takes 6 hours to repair a slightly damaged tracked vehicle. The level of repair capability defines the number of vehicles that must be transported from the brigade's repair installation to the higher echelon's. So, the repair capacity defines the evacuation requirement from the brigade to the higher echelon and the evacuation capacity defines the place of the repair installation.

Based on the comparison between required and available transport capacity, a set of rules are triggered that suggest a number of solutions until a satisfactory location for the repair installation is found and sufficient vehicles can be evacuated or until all rules are exhausted e.g. moving the mean distance from 25 km to 20 km could increase transport capacity by 20%. Clearly the close inter-dependence between

An intelligent gaming environment
installation location and evacuation distance and capacity, prevents us from dissociating these problems.

Figure 4.4: The maintenance decision making process

An intelligent gaming environment
However, the evacuation and the repair problem could be subdivided into two separate fields, namely the tracked vehicles problem and the wheeled vehicles problem. Indeed, tracked and wheeled vehicles do not require the same type of evacuation resource but they are repaired at the same installation. We have not structured the decision object accordingly because repair capacity can be transferred from tracked to wheeled vehicles when it is available. So, they may share resources. Therefore these problems must be solved simultaneously and the maintenance automaton is a non-divisible decision object.

4.3.3 A medical support decision automaton

![Diagram](image)

*Figure 4.5: The medical decision making process*

An intelligent gaming environment
The medical support decision automaton handles only a single knowledge domain and constitutes a basic building block of the brigade automaton as discussed in RBDC [1982]. We do not describe it any further because of its analogy with the previous problem. The decision making process is shown in figure 4.5.

4.4 An intelligence decision automaton

The Intelligence problem addresses two main problem areas as described in RBDC [1987], [1985] on the intelligence work.

Firstly, the information gathering problem which is centred around the question of identifying suitable information sources and subsequently acquiring them.

Secondly, the information analysis problem which focuses on relations that can be inferred from freshly gathered information and already available information in order to determine the opponent's opportunities and his most probable course of action.

The Intelligence decision making process is described using the entity relationship diagramming technique as shown in figure 4.6. Entities represent data items. Relationship symbols depict steps in the decision making process. Indications about the type of relation, e.g. 1:m, are not used.

Clearly information gathering precedes information analysis. Furthermore the resources used to solve both problem areas are different. Therefore the intelligence decision object has been sub-divided into two decision automata:

- the intelligence gathering decision automaton;

- the intelligence analysis decision automaton.

Figure 4.6 illustrates the information and reasoning steps that are used in the intelligence decision making process.
Figure 4.6: The intelligence decision making process

An intelligent gaming environment
The intelligence gathering reasoning process describes:

- how to identify the information that must be gathered based on the mission of the brigade and the expected attitude of the opponent e.g. if one is attacking against a delaying enemy important information is the location of enemy prepared defensive positions;

- how to define an information gathering plan which identifies the resources that should be used, where they should be used and when e.g. an air reconnaissance over the brigade’s sector two hours before the start of the attack can provide important information about the location of enemy units.

The intelligence analysis decision making process entails the following decision steps:

- identify the prospective opponent and verify the expected attitude based on freshly gathered and past information e.g. if 5 tanks are spotted where an enemy unit used to be, it can be supposed that the unit has not moved;

- determine what the opponent’s possible courses of action are;

- determine the opponent’s most probable course of action.

These last two steps apply the tactical decision making process that is described later.

4.5 A staff decision automaton

The staff problem is constituted by two separate problems as described in RBDC [1988], [1987], [1986], [1985], [1982] on the staff function within a brigade.

The first problem is referred to as the initial place taking or staff movement problem and focuses on the way units are able to reach given positions with certain movement restrictions imposed by the commanding echelon. Restrictions range from limited choice of route, or movement time to the entire definition of the staff movement plan.
The second problem addresses the problem of command time availability. Indeed, the brigade must take into account its own command delay and that of all its subordinate units. This delay may prove to be the constraining factor in the definition of the earliest start time of the brigade’s manoeuvre e.g. the time the brigade will be ready to actually start its mission.

![Diagram](image)

Figure 4.7: The staff decision making process

An intelligent gaming environment
These problems focus on separate areas but both influence the definition of the brigade's mission start time. Solving both problems can be a concurrent process. The most constraining result of both decision-making processes will determine the brigade mission start time. Therefore the decision object is divided into two sub-objects whose results are coordinated by the staff object. Figure 4.7 illustrates the staff decision-making process.

As an example, we illustrate the classes and rules that are used to determine the rhythm of the movement and the location where a temporary dispersion zone i.e. a place to stay some time, could be envisaged. The format of the example describes the classes and their attributes that are used, the rules and the demons that are applied and finally the procedural body that describes the sequence of the reasoning process. The abbreviations are explained at their first occurrence.

Classes:

Order:

Mov restriction : Restriction type. (User defined)

Time:

Day : int. (Integer)

Night : int.

Mov plan:

Rythm : Mov type. (User defined)

Temp DispZ : str. (String)

Mov time : Int.

Location:

Name : str.

Loc type : Terrain type.

An intelligent gaming environment
Mov duration to start : int.
Mov duration to end : Int.
Candidate : truth. (Boolean)

Rules:

Rule1 :

o: Order, t: Time, m: Mov plan
if o.Mov restriction = Mov by night
and m.Mov time  t.Night
and m.Mov time  2 * t.Night
then m.Rythm = Mov in two nights.
endif.

Rule2 :

m: Mov plan,
l: Location
t: Time
if m.Rythm = Mov in two nights
and l.Loc type = wood
or l.Loc type = village
and l.Mov duration to start  t.Night
and l.Mov duration to end  t.Night
then l.Candidate = true.
endif.

An intelligent gaming environment
Demons:

Demon1:

l: Location, m: Mov plan, l1: Location
when l.Candidate
and l.Mov duration to end l1.Mov duration to end
and l1.Name = m.Temp DispZ
then m.Temp DispZ = l.Name.
endwhen.

Actions:

read order.

create member m1 in class Mov plan.

obtain m1.Rythm.

if m1.Rythm Mov in one
then obtain m1.Temp DispZ.

These rules determine whether a temporary dispersion zone is required and which locations would be suitable. Whenever a location is suitable, demon1 is triggered which determines the location that is furthest from the end location. Hence, the second night is used at its fullest in order to delay any detection. The complete description of the decision automaton is given in RBDC [1990].
4.6 A tactical decision automaton

The tactical decision making process combines information provided by the staff and intelligence decision objects with a terrain and resource analysis to define a tactical plan that achieves the objective(s) set by the commanding division.

The tactical terrain analysis determines the possibilities and constraints that follow from the terrain that must be used to achieve the objective that has been given by the commanding division. The important aspects that must be inferred are the portions of terrain that allow mobility of units and those that hinder it considerably. Therefore the result of this automaton's reasoning process will be a number of unit accesses and key terrain portions that can be found on these accesses. Accesses that appear on the flank of a brigade's sector i.e. that part of the terrain that has been allocated to that brigade, are also considered. Furthermore, the main lines that specify a terrain's counter-mobility possibilities are identified.

The information gathering and situation analysis steps of the tactical decision making process are carried out by the staff, intelligence and terrain analysis decision automata. Indeed, they provide three of the four major situation descriptors namely the mission, the environment and the opponent.

However the situation assessment process is completed by combining these separate information sources with the tactical resources that the brigade possesses to carry out its mission.

The process of generating manoeuvre options is structured according to specific mission-oriented rules e.g. attack, static defense, delaying action. For example, specific knowledge which relates to the importance of terrain characteristics and the use of available units and weapon systems, is applied to define an attack. Furthermore enemy possibilities must be considered, so the possible defensive enemy manoeuvres must be designed. Therefore both main knowledge domains of defense and attack are used concurrently. This decision making process is illustrated in figure 4.8.

The following rules illustrate the above mentioned combination and how it is used to choose the preferred manoeuvre. The previous option generation process generates five possible offensive manoeuvres.
Figure 4.8: The tactical decision making process

An intelligent gaming environment
Class:

Manoeuvre:

Main axis : str.
Speed : str.
Nr enemy positions : int.
Best : truth.
Axis:
Name : str.
Access list : list of str.
Access:
Name : str.
Point list : list of point.

Point:

quadrant : str.
xcoord : int.
ycoord : int.

Defensive position:

Name : str.
Point list : list of point.

Rules:

Rule1:

m: Manoeuvre, a: Axis, c: Access, d: Defensive position

An intelligent gaming environment
if m.Main axis = a.Name
and c.Name in a.Access list
and c.Point list * d.Point list 0
then m.Nr enemy positions = m.Nr enemy positions + 1.
endif.

Demons :

Demon1 :

m1: Manoeuvre, m2: Manoeuvre
when m1.Best = true
and m2.Nr enemy positions = m1.Nr enemy positions
then m2.Best = true.
m1.Best = false.
endwhen.

Demon2 :

m1: Manoeuvre, m2: Manoeuvre
when m1.Best = true
and m2.Nr enemy positions = m1.Nr enemy positions
and m1.Speed = m2.Speed
then m2.Best = true.
m1.Best = false.
endwhen.

Actions :

An intelligent gaming environment
forall m:Manoeuvre do

obtain m.Nr enemy positions

endforall.

At the end of the "forall" loop the manoeuvre that has the main axis which crosses the least number (Nr) of enemy defensive positions and which is the fastest, is marked as being the best. In this manner, the decision automaton can make an evaluation and decide upon its course of action. RBDC [1990] describes the complete decision automaton.

4.7 A communication decision automaton

The communication decision automaton ensures that the brigade headquarter stays in contact with the higher level echelons and its subordinate units. It combines information about the communication network configuration for the brigade and higher echelons which is provided by the Corps, the brigade's own tactical plan and the terrain to decide:

- which network gateway will be used and for how long e.g. the closest gateway that is open during the command post's installation;

- the successive locations of the brigade gateway e.g. the gateway moves every eight hours;

- when to move to the following location and how.

This reasoning process is shown in figure 4.9 and is based on RBDC [1988] on a brigade's communication with higher echelons.

We have chosen not to specify the brigade's internal communication network because brigades have a redundant well-defined communication structure which is sufficient to provide the necessary communications. Furthermore, the status of the electronic warfare environment attribute is sufficient to evaluate communication delays in the simulation model e.g. in an area with heavy jamming of certain frequencies, radio communication may be severely impeded which implies longer delays for reports to reach their destination.
This automaton can inform the overall brigade automaton that communication with higher echelons may not be possible for a certain time due to the tactical automaton’s imposed rhythm of movement. If this duration exceeds a certain limit, the tactical manoeuvre’s rhythm will have to be adapted.

Since this problem solving process focuses on a single type of resource, the automaton is not sub-divided.

![Diagram of communication decision making model]

*Figure 4.9: The communication decision making model*

An intelligent gaming environment
4.8 An engineering decision automaton

The brigade’s engineering decision automaton plans the distribution of engineering resources to combat units in order to ensure mobility or counter-mobility support. These two problems can be sub-divided into a terrain analysis portion which identifies the terrain features that are important during the course of the manoeuvre and an engineering work definition part which allocates resources to units which may encounter or defend these terrain portions. This analysis is based on RBDC [1987], [1986] about engineering in the brigade. The automaton has been structured into:

- the engineering terrain analysis automaton which identifies critical engineering terrain sections e.g. a narrow open section of the terrain can be used to lay a mine field

and

- the engineering work definition automaton which analyses the amount of personnel and material required per critical terrain section, sets priorities among these sections based on the tactical manoeuvre plan and allocates personnel and resources to the most critical sections e.g. the most complete line of difficult terrain across a brigade sector is allocated engineering platoons to prepare defensive positions.

The engineering decision making process is illustrated in figure 4.10.

It might seem possible that the second decision automaton can further be divided by considering the rules that apply for each specific resource. However, resources are complementary and cannot be handled independently. Therefore the strong relation between resource applicability rules indicates that division is not suitable.

This automaton may have an important influence on the brigade decision making process if it decides that terrain sections critical to the success of the manoeuvre cannot be covered with the existing resources e.g. a river cannot be crossed within the required period. It may request extra support from the commanding echelon or force the brigade automaton to reconsider the tactical plan e.g. a main thrust axis might be chosen differently because of lack of engineering support.

An intelligent gaming environment
Figure 4.10: The engineering decision making process

An intelligent gaming environment
4.9 A fire support decision automaton

Fire support focuses on the use of three different types of resource namely artillery, combat helicopters and offensive air support. These resources have been named in a sequence of increasing scarceness. Therefore they are applied differently. Complementarity is less evident. So, the decision object comprises three sub-objects.

The artillery decision making process is illustrated in figure 4.11 based on RBDC [1988]. It shows that the tactical plan, TacAle, determines the targets e.g. for an attack, artillery support must be given at the start. Based on the nature and width of a target e.g. open or wooded terrain an estimate is made of the required artillery battalion fires. Comparing this requirement with the available fires and taking into account the priority of targets e.g. the start of an attack has first priority, firing credits are allocated to targets. If a movement of the brigade occurs, the artillery battalions must also move to provide continuous support. So, a phased movement strategy is adopted e.g. in the case of 2 artillery battalions, the first moves while the other stays in place and vice versa.

The combat helicopter decision automaton is not developed because this resource is currently unavailable and rules of application are not yet clear.

A brigade may request offensive air support when the before mentioned resources are depleted. So, the decision automaton is currently not implemented.

4.10 The internal communication of the brigade decision object

The previous paragraphs have probably made clear that the brigade decision making process is not straight-forward and linear. Therefore, an overall brigade automaton has been created which coordinates all sub-automata, executes them in a given sequence and analyses their results. This means that the reasoning path may be adapted based on an automaton's results. A management by exception strategy is adopted to decide whether the decision making path continues as planned or must return to a previous step. In the latter case, the decision making must restart with extra constraints. By way of example, let us suppose that the communication automaton concludes that the 4 hour non-communication between brigade and division threshold is reached. The brigade decision making process
Figure 4.11: The artillery decision making process

An intelligent gaming environment
must step back and define a new tactical plan which has a rhythm that constrains the brigade to advance at a slower pace. The change of rhythm ensures that the 4 hour non-communication threshold is not reached. In order to control this looping process and ensure that the reasoning process is convergent, re-planning is registered for each automaton. The number of re-planning loops is limited because every return to a previously executed automaton indicates that the brigade is insufficiently capable of carrying out its mission. Reaching the loop-threshold triggers the brigade to inform its commanding echelon that it cannot fulfill its mission. Using the decision making rules, the brigade decision object can provide a sufficient explanation to its commanding unit. So, the necessary dialogue between decision objects is supported and can be presented in a manner that is understandable by human decision objects.

4.11 The mission planning decision automaton

The air mission planning decision making process centres around the need for and the allocation of resources to a certain number of mission types, namely:

- Offensive Counter Air;
- Defensive Counter Air;
- Suppression of Enemy Air Defense;
- Close Air Support;
- Battlefield Air Interdiction;
- Reconnaissance.

Each mission type corresponds to a specific problem domain. However, different mission types may compete for the same resources.

Applying the knowledge structuring principle to the air mission planning problem, the structure of the automaton as shown in figure 4.12, has been defined. It is based on the organisation of the airforce planning functions at ATAF, ATOC, SOC and Wing echelons as described in RBDC [1988] on airforce mission planning.

An intelligent gaming environment
Figure 4.12 illustrates that the ATAF interacts with a DOO (Daily Operations Order) automaton which gathers the necessary information to execute offensive and

![Diagram](image-url)

*Figure 4.12: The mission planning decision making process*

An intelligent gaming environment
defensive planning. So, the first action of the mission planning automaton is to call the DOO automaton. The defensive planning automaton is always called to make an assessment of the situation. Offensive planning is only called when assets are allocated to offensive operations and target priorities are defined in the DOO.

Within each component the knowledge structuring principle has been applied recursively as will be described in the following paragraphs.

4.11.1 The defensive mission planning automaton

Four distinct problem and thus decision making domains can be identified when analysing the defensive mission planning problem namely:

- the situation assessment problem: this concerns the analysis of the opponent’s actions upon our air defense resources and the intentions these may imply e.g. if a large percentage of radars are destroyed in a certain area, this may indicate that the opponent wishes purposefully to reduce our detection capability in order to attack us more easily;

- the detection resources problem: this involves the decision whether and if so where to enhance the detection capability by the use of the NAEW system e.g. a detection gap can be filled by the NAEW;

- the CAP problem: this concerns the use of Combat Air Patrols to protect resources e.g. the NAEW system or defend certain key areas of airspace;

- the squadron allocation problem: this involves the use of distributing the available air defense resources to CAP’s that have been defined during the previously described decision making step.

The relation between these problems is shown in figure 4.13. It is based on RBDC [1986] on air defense mission planning.
Figure 4.13: The defensive decision making process
4.11.2 The offensive mission planning automaton

The offensive mission planning decision object concentrates on three main areas of interest, namely, targeting, resource allocation and mission routing. This structure is illustrated by figure 4.14 and is derived from RBDC [1988] on offensive air mission planning.

Figure 4.14: The offensive decision making process

An intelligent gaming environment
The targeting decision object contains the rules that specify the priority of a type of installation. This targeting process uses information about the current situation and certain priority accents set by the user to suggest a set of primary targets. This set is then used to define secondary targets. The destruction of these targets is instrumental in reaching primary targets. As an example, airbases can be evaluated as targets because they have a certain type of aircraft which is dangerous for the player’s own assets. Those airbases that can generate the highest sortie generation capability i.e. the amount of flights of that type of aircraft, and can reach the player’s own assets are chosen as primary targets. The surface to air missiles that are near to the airbase are designated as secondary targets.

The resource allocation decision object specifies the most suited available aircraft/weapon combination for each target. Targets of the same level of priority are discriminated further and missions are identified e.g. the airbase with highest sortie generation capability is allocated a mission first and so on. For such a mission an aircraft-weapon combination is taken that is in range of the target, is available and has more than a 50% chance of closing the airbase for several hours.

Finally, the routing decision object specifies the best route for each mission taking into account the necessary airspace management constraints e.g. a mission is preferably not routed through a suspected enemy corridor to avoid fratricide.

4.12 Development approach and knowledge acquisition

An iterative design approach is adopted based on the design process defined in chapter 3. The approach is called iterative because the development cycle is kept short, 3-4 weeks, and is repeated 3-4 times. An essential part of this approach is the participation of experts in designing and validating the prototypes that are built.

We go through the phases A.1 to A.3 that are defined in chapter 3, in two steps.

In a first step, we describe the military decision process for the echelons referred to above using the RBDC’s course ware. We apply the knowledge structuring principle to determine the independent knowledge fields. We then roughly identify the information requirements and related decision making rules for each specific problem area. The result of this phase is a decision making process diagram and accompanying explanation as shown in the previous paragraphs.
The next step is to discuss our conceptual model of the decision making process with experts e.g. the teacher responsible for the specific problem area. This discussion typically takes 4 to 6 hours and focuses on the decision making process and rules. Information is not addressed explicitly. Enhancements are made to the conceptual model and an agreement is made about the capabilities of a first prototype.

During phase A.4, we develop a prototype of the decision automaton within 4 weeks and present its results to the expert and his or her colleagues. A presentation tool must be used that allows an expert to make an effective judgement about the quality of the automaton's reasoning process. In our case, a map based display capability is developed to validate the decision object's conclusions. This interface presents information in a way that can easily and rapidly be understood by users. Experts evaluate plans on map backgrounds and represent resources by specific icons e.g. two crossed lines represent an airbase. So, the interface displays a map background with the corresponding icons. This is illustrated in great detail in paragraph 4.13.

Also shown is the capability to check each step of the automaton's decision making process by providing a selective presentation option. Relevant information is displayed consistently with every step in the problem solving process e.g. critical terrain sections can be displayed selectively to illustrate the first step in the terrain analysis process. The result of this discussion is a set of enhancements to the existing decision automaton. These changes are documented and a second development cycle is started. Meanwhile, the first prototype is submitted to the cooperating expert and his or her colleagues for further testing.

This process is repeated until the decision automaton's results are considered valid by the expert and a team of colleagues. Validity is measured by the degree of military realism that is displayed in the automaton's conclusions. A set of characteristic problem situations, typically exercise problems that are used at the RBDC, must be solved by the decision automaton. Exercise problems are well known by experts and a consensus exists about the set of valid solutions. Each solution is presented to a group of experts and discussed. The automaton is considered valid when the group of experts agree about the validity of its solutions to all characteristic problem situations.

The attribute part of the classes that are defined in the decision automata determine the data that must be stored in the relational data base that contains all the data about the IALTA objects. The attribute part of the classes already displays a

An intelligent gaming environment
relational structure e.g. the squadron class contains a unique reference to the airbase class since a squadron is always located on a single airbase. It is however necessary to go through a normalisation step to ensure the relational structure of the data base. So, translating the data structure that is used within the classes of a decision automaton to a relational data base structure is a relatively straightforward operation. The necessary interfaces between automaton and interface, namely data base selection functions and the corresponding file structure, are also defined.

The thresholds in the decision making process are identified by determining the constraints that a decision automaton may put on the entire decision making process e.g. a shortage of transport capacity may force a brigade to reduce the number of attacking units. An important aspect of this threshold analysis is the definition of resources that are critical for the accomplishment of a mission. If a decision automaton decides about the use or allocation of such a critical resource, it is necessary to determine the minimum amount of resources that are required to carry out a mission. The following example illustrates that the value of a threshold is set dynamically. On the access used by an attacking unit there are two rivers that must be crossed, the rivers can be crossed with a tank bridge. If the rivers are sufficiently distant from each other, a single tank bridge is enough. Otherwise two tank bridges may be required or a certain delay in the progression of the unit must be accepted. In this example the number of available tank bridges is the critical factor and the threshold may be 1 or 2 depending on the distance between the rivers and the required speed of the attack. So, thresholds are related to critical resources and their value is situation dependent. Furthermore, the criticality of resources varies with the mission e.g. in a static defense the number of available tank bridges is irrelevant.

The decision automaton’s decision making action is called by the overall decision object’s decision making action or by execution function of an event as described in the following chapter. The action call returns whether a threshold has been reached. Also specified is the constraint that must be imposed on the overall object’s decision making process. Consequently the decision making action calls a related threshold resolution action which contains the knowledge to resolve the situation e.g. reduce the expected advance speed or change the access selection criteria.

An intelligent gaming environment
4.13 Implementation

4.13.1 Choice of tools

The tools used to support this phase are:

- Excelerator and MS-Word, to document the interviews and the results;

- KES, a knowledge engineering shell to develop the prototypes;

- Turbo Pascal, to develop a graphical interface.

These tools allow us to concentrate on the methodological aspects of the development problem because they support powerful development primitives. Furthermore, development is constrained to a Personal Computer (PC) based environment and must provide a high level of portability of the decision automata. Automata must be rapidly adaptable and must be used on a wide variety of hardware platforms. Therefore a tool is necessary that combines maximum portability with a sufficient support for rapid development. KES provides this combination. Turbo Pascal has been chosen for the graphical interfaces because it provides a great deal of support for this sort of application and it is foreseen to provide players in the war game with a PC. Therefore, the graphical interfaces do not have to be ported to an other hardware environment. So, emphasis can be put on the rapid development and structure of the application. This is supported by the Turbo Pascal graphical library and the Pascal language constructs.

The development of the decision automata represents a total effort of 74 man/months. Eight developers that are managed by myself, participated in this work over a period of three years. The decision automata are implemented in KES and C. The source code amounts to 18000 lines of KES and 4000 lines of C. This corresponds to 120 classes, 600 rules and 500 demons.

4.13.2 The brigade graphical interface

The brigade graphical interface is shown in figures 4.15 through 4.19.

Figure 4.15 illustrates the map background that can be generated from the terrain data base that is defined to support the tactical terrain analysis described in 4.7.

An intelligent gaming environment
Figure 4.15: The brigade validation interface

An intelligent gaming environment
Figure 4.16

An intelligent gaming environment
An intelligent gaming environment
Figure 4.18

An intelligent gaming environment
This representation contains all relevant military terrain data which combines data from ordinary military staff maps, road and bridge maps and cross-country movement maps. So, all the terrain data is shown which provides an accurate basis for the evaluation of the tactical decision automaton because it determines the trafficability of the terrain i.e. how the terrain can be used for movement by units. Figure 4.16 displays the data that is relevant in the terrain analysis process. Map backgrounds can be created selectively.

As discussed previously, the graphical interfaces can display each step of the decision making process separately. Figure 4.17 illustrates the first step of the tactical decision making process, namely the identification of critical terrain sections, dotted black lines, i.e. terrain sections which can provide a major tactical advantage. Furthermore, this figure shows that the analysis process is constrained and influenced by the sector that has been allocated to the brigade. The starting line, the objective, the left and the right limit of the sector determine the main movement direction. Also shown are the potential defensive positions, full black lines, that can be used by the opponent.

Figure 4.18 illustrates the second step in the tactical decision making process, namely, the definition of accesses between critical terrain sections. All accesses between 2 distinct critical terrain sections are determined and can be distinguished by their width and type. The width of an access indicates the amount of forces that can move as a front line. We consider three types of access, namely tank, infantry and combined with a predominance of tanks. Types are indicated by a different colour. This feature allows the experts who evaluate the results to clearly see the options that are generated by the decision automaton.

Figure 4.19 displays one of the five manoeuvres that are generated by the decision automaton. The thick red arrows indicate the suggested main thrust which is a sequence of accesses. Thick black lines indicate the other accesses that are used in the manoeuvre. They may be complete i.e. from start to objective or just partial. Also shown are the units that are ordered to move on each access. An access may be allocated to several units i.e. only unit is actually performing a tactical mission, the other units are reserve units that move on the same axis. So, the experts may evaluate the different options and generate comments about the quality of the reasoning process.

An intelligent gaming environment
Figure 4.19

An intelligent gaming environment
The previous figures emphasize the familiarity with the expert's usual working environment that we achieve in this interface. It mirrors a brigade officer's normal working environment with a similar presentation of facts.

4.13.3 The air mission planning graphical interface

Figure 4.20 shows the overall airforce situation as it is presented to experts to evaluate the threat and to plan their missions. All relevant information is displayed and this figure emphasizes the need for a clearer presentation of data. This is supported by the selectiveness with which data can be shown as discussed above. Furthermore, the expert may zoom in on a given area as illustrated in figure 4.21. It must be noted that each icon can be queried for data about its position, nature and resources e.g. an airbase provides information about its full name, coordinates, squadrons and logistical status.

Figure 4.22 displays the conclusions of the air mission planning decision automaton. Again relevant data can be selected and shown separately in order to illustrate each step of the decision making process. In this figure the aircraft symbols represent Combat Air Patrols. When queried these icons provide data about their location, CAP duration and start, number of aircraft and squadrons that execute the mission. Also displayed is the position of the AWACS, the aircraft with a "parachute" on top of it and the expected offensive corridor i.e. that part of airspace which an opponent might use for offensive missions. In a corridor, the interface also calculates the line of first interception i.e. an imaginary boundary where ground based interceptors could engage enemy aircraft based on the air defense's reaction time and flight duration. This line is an indication as to the need to define a CAP, i.e. if the line of first interception is too far within one's own airspace then it is necessary to protect that airspace with a CAP. So, this interface illustrates the data that are used by the air mission planning decision automaton to assess the situation and define the appropriate defensive posture.

Furthermore, figure 4.22 shows the results of the offensive mission planning by drawing circles around targets and providing information about the missions that will attack the target. Each mission describes the aircraft type, number and payload i.e. the weapon combination that is used, and the airbase and squadron that carry it out. So, experts have all the necessary data to evaluate the conclusions of the decision automata.
Figure 4.20

An intelligent gaming environment
Figure 4.21

An intelligent gaming environment
An intelligent gaming environment

Figure 4.22
Summarizing, this chapter describes the development and implementation of the decision making actions of the decision objects. We focus on the principle that can be used to structure decision automata, namely identify independent resource related knowledge domains, and on the coordination of decision automata. In this context, determining a decision making precedence and decision making thresholds is shown to be feasible and applicable.

The following chapter discusses the static and dynamic object structure of the IALTA simulation environment. It describes how decision objects use their decision making actions to drive the actions of execution and control objects and the constructs that are required to simulate the execution of orders realistically.
Chapter 5: The dynamics of the IALTA gaming environment

This chapter describes the structure of the simulator, control, execution and environment objects of the IALTA gaming environment. We discuss the constructs that are necessary to support the representation of the dynamics of military reality in order to form a multi-purpose gaming environment as discussed in chapter 1.

5.1 Modelling and structuring

The general conceptual framework that is applied to build the simulation model is introduced in chapter 2. Figure 2.3 presents the overall objects that compose the simulation model and describes the relations that exist between them. Chapter 2 also defines objects as consisting of an attribute and action part. Chapter 3 gives an overview of the part of military reality that must be modelled in the IALTA gaming environment. Chapter 4 describes the action part of decision and decision support objects that are developed in the context of the IALTA gaming environment. As discussed in chapters 1 and 2, a model of military reality must contain objects that mirror real objects in their description and behaviour. So, a first step in the development of the IALTA Model of Military Reality (MMR) is to identify the relevant objects and describe their characteristics. A second step is to identify their behaviour and the mechanisms which ensure that these behaviours are properly coordinated, synchronized and evaluated to achieve a dynamic model that has a good correspondence with reality.

We start by discussing the objects that are part of the IALTA MMR and the object hierarchy that they constitute.

5.1.1 IALTA object hierarchy

The IALTA object hierarchy is described in the following diagram and represents a reduction of reality.

During the development of the decision automata, we identify independent knowledge domains by concentrating on the resources that are the subject of this
Figure 5.1: The IALTA object hierarchy

An intelligent gaming environment
knowledge. So, during the development process of a decision automaton, the resources are defined that are relevant to the decision making process which we want to support in the IALTA gaming environment. A change in the state of any of the resources may have an impact on the decision making process of a decision object or may require a decision making process to start. Therefore, we suggest that these resources are described by execution objects in the simulation model. Furthermore, the attribute part of the classes that model resources in the decision automata must be included in the attribute part of the corresponding execution object.

Let us then look more closely at the nature of these resources and how we describe them in the simulation model. We start by focussing on the attribute part of the objects. We discuss each of the decision automata described in chapter 4 and identify the resources and their attributes that must be modelled by execution objects. Within each decision making process, we identify resources which may cause a problem and resources with which the problem can be solved. We also determine the description of the environment in which the problem must be solved.

The staff decision making process described in chapter 3 addresses the movement of a brigade which is constituted by subordinate units at battalion and company echelon. Within these units, the number of vehicles is a relevant attribute. The type of unit is also important in case all units cannot be moved e.g. for an attack, one will move only the combat units if there is insufficient time. Therefore, we specialize battalion and company units according to their type. These are the resources that form the problem. The resource with which to solve the problem is the available road network. We need to know the amount of time a route may take and the wooded areas that can provide cover against detection and attack as described in the example in chapter 4. Roads are divided into three classes with specific speeds. A road object is designed with a type and speed attribute. It is not specialized into three types because there are no structural differences between the three types of road. In this case, the resource with which to solve the problem is at the same time its environment namely the terrain.

The intelligence decision object seeks to gather information about enemy units and define their possibilities. So the resource that causes the problem is the unit. A unit can be of several echelons and within an echelon of different types. So, we specialize units according to their echelon and within the echelon according to their type. Information gathering resources are scouting platoons, drones i.e. unmanned

An intelligent gaming environment
aircraft, aircraft etc. A scouting platoon is a type of unit which has not been identified yet. However, we do not model it as a unit because it receives orders from a battalion or company and these units are not decision objects, i.e. objects that give orders. So, it is modelled as a land based object with specific attributes relating to depth of information gathering i.e. range in kilometres, reporting and operational delay. Drones and aircraft are airborne objects with similar attributes. Intelligence gathering takes place on the terrain. Terrain trafficability and cover opportunities are important characteristics which must be addressed in the model of the terrain. This model is completely described below when we describe the resources that are involved in the tactical decision making process.

The second part of the intelligence decision making process i.e. defining the enemy’s possibilities is a tactical decision making problem. So, we apply the enemy’s tactical concepts to foresee his plans and manoeuvres. The resources that are used in this process correspond to those of the tactical decision making process.

The tactical decision making process starts with a terrain analysis. So, terrain is modelled as an object and spawns two sub-objects namely point and line objects. A point object represents a surface and is further specialised into sub-objects based on the surface trafficability characteristics like wood, lake, town, village and crossing objects. A line object consists of a list of points and is specialised into road, river, railway and uncrossable sub-objects based on their different trafficability characteristics. Furthermore, the point object is specialised into weather sensitive and difficult sub-objects to include the influence of weather and slope on trafficability. A point is weather sensitive when weather conditions may alter its trafficability e.g. a small river may be easily crossable in summer but not in spring because melting snow increases the current. So, weather conditions can be defined for each individual point of the terrain. Difficult point objects can be instanciated and indicate that the trafficability that is linked to the point’s surface e.g. wood is further degraded.

The tactical decision making process analyses the brigade’s combat resources i.e. combat battalions and companies and allocates them to certain missions. A decision is made about the composition of these units. So, attributes must be defined to describe the assets of units. A linked list of weapons is added as an attribute to the unit object which contains the weapon identifier and the corresponding number of weapons.

An intelligent gaming environment
The logistical decision making process uses data about each unit’s number of damaged vehicles, wounded personnel and consumption as discussed in chapter 4. So, the corresponding attributes are added to the unit object. These attributes relate to the resources that cause the logistical problem. The resources that are used to solve the logistical problem are tow trucks, tank transporters, repair installations, ambulances, sorting stations, hospitals, trucks and depots. The corresponding objects are created and are land based mobile objects. Repair and sorting installations, hospitals and depots move in time. The environment of the logistical decision making process is the road network and the terrain features that enable the installation of logistical units e.g. installations are preferably placed in villages. The necessary terrain objects are already modelled for the solution of the tactical problem as described previously.

The brigade communication decision automaton requires the definition of a communications network to function properly. So, a network node object is defined to model each part of the network. This object is identified by its coordinates and has a second attribute that describes the list of nodes to which it is connected and its opening hours. It is a land based mobile object. The brigade object must contain data about its communication resources i.e. number of communication stations, to complete the decision making process. No specific object type is created because a brigade’s communication stations do not operate independently. They always accompany the brigade’s headquarters. The environment is the terrain which must contain altitude data to determine whether communication is possible. We do not include this data in our model of the terrain because we suppose that there are sufficient elevated points to ensure that communication is always feasible. This assumption is corroborated by cooperating military communications experts who state that line of sight is a very minor consideration at brigade level.

The artillery decision making process must support the tactical decision. In doing so, it requires data about available artillery battalions, their composition and their stock levels. The number of available batteries attribute is added to the corresponding unit sub-objects. The logistical status is inherited from the unit object. The environment is the road network and the tactical model of terrain which allow the simulation of movements and of the installation of artillery positions.

The engineering decision making process supports the tactical decision. It distributes engineering resources i.e. engineering platoons, tank bridges etc. to combat battalions. So, the engineering company object must describe the number
of available engineering resources for each resource type e.g. the number of tank bridges. The environment is the terrain as it is modelled for the tactical problem.

Our choice of echelons that must be modelled to simulate army operations is also based on the generally accepted command paradigm that a commander can control and order two command echelons below his own. The lowest echelon that is played by decision objects is brigade. So, companies are the lowest echelon that is modelled as an object and platoons do not constitute sub-objects of units.

For the airforce, the ATAF, ATOC and ASOC hierarchical decision making process results in the definition of an Air Task Message (ATM) for offensive operations. An ATM specifies the squadron that must carry out a mission, the number of aircraft and their payload i.e. the number and type of weapons. The ATAF and SOC decision making process may define CAP’s, move SAM’s and/or schedule a NAEW mission. It also requires radars to warn its interceptors of possible enemy raids. The Wing decision making process consists of the implementation of ATM’s by assigning individual aircraft, plotting their route and determining their take-off time. We make a reduction of reality at this point by not modelling individual aircraft. Indeed, at ATAF or ATOC level, it is irrelevant to know whether the aircraft with tailnumber AC111 flew a mission rather than AC112. So, we must model the different types of aircraft, weapons, radars, Surface to Air Missiles (SAM), airbases and squadrons.

We notice that some decision objects e.g. the brigade tactical decision object, do not solve problem situations that stem from a change in the state of the resources they manage. Rather, the decision making process is initiated by an order from the unit that commands the brigade. In order to model this flow of information between objects, we create a message object which is the ancestor of orders, reports and free text messages.

So, by defining the resources that generate problem situations, those that are instrumental in solving them, the environment that influences the solution and the information flow between decision and execution objects, we can determine the objects that must at least be represented in the simulation model.

Figure 5.1 illustrates the IALTA object hierarchy. Multiple inheritance i.e. an object inherits attributes and behaviours from several objects, is used to structure our objects.
This paragraph discussed the attribute part of the IALTA object hierarchy. The action part of the objects and the related problems are presented in the following paragraphs. We start by discussing the way time is modelled in the IALTA gaming environment.

5.1.2 The IALTA time sequencing set

As introduced in chapter 2, objects take actions in the simulation to fulfil their mission. We can distinguish between intentional and unintentional actions e.g. moving and being seen. In order to model reality, actions must be handled according to the moment at which they occur and the duration they take to complete. Therefore we define a queue or list of actions which represents time. However, the following problems must be solved to model the dynamic behaviour of objects realistically:

1. should actions that last for a certain duration be handled in the same way as actions that are instantaneous?

2. how should actions be inserted into the queue?

3. how should concurrent actions be handled?

An object’s actions may be discrete in time e.g. the dropping of a bomb or may have a more continuous form e.g. a movement. We distinguish between actions that have a discrete character which we shall call events and actions that have a continuous character. In the latter case, we speak about activities. This distinction is necessary because an activity i.e. an action which is continuous, may be influenced by other activities or by events e.g. a bomb being dropped on a moving column of vehicles, as will be discussed in the following paragraph. Therefore we require some sort of mechanism that handles these different type of actions and evaluates the state of activities after every interruption, namely, an action that occurs before the completion of the activity.
We have defined two types of action, discrete and continuous actions, that are structured as follows:

- **time**: time at which the action takes place;
- **elapsed time**: amount of time that the activity has been simulated;
- **execution function**: function that describes the actions that should be taken when the action takes place.

Actions are handled in two specific manners.

Events, discrete actions, are placed in the event list and are always inserted before activities, continuous actions, that occur at that moment. This is preferable because events can be handled completely at this point in time whereas activities require a period of time to execute e.g. a movement or combat. Furthermore, events may cause activities to be cancelled e.g. the start of a unit's movement can be cancelled because the unit has been destroyed by a bomb dropped by an air raid.

Activities are simulated for a period of time that is equal to the difference between the first next action time and the current time e.g. if the current time is 1000 and the next action occurs at 1020, activities can be simulated for a maximum of 20 minutes. If the activity has not reached its final state, all events occurring at the first next action time are simulated and the interrupted activity is re-inserted in the list. The amount of time that the activity has been executing is stored in the elapsed time attribute. The value of this attribute is important because our map of the terrain is discrete i.e. it is a set of points that have a certain surface as described earlier. The period that an activity is simulated may not be sufficient to pass from one point to another in the case of a movement. However, we do not want the moving object to start its movement all over again. So, the elapsed time attribute allows us to keep track of the amount of time that the object has been moving within the point and insures that the object's change in position will occur at the right moment.

In the previous discussion, it is assumed that the first next action time is known e.g. it occurs at 1020. However, unintentional actions may take place e.g. a moving attacking unit is detected by an enemy unit. So, we introduce a new concept namely the Action Map object. In an object-oriented model all actions are initiated by objects. It is obvious that there must be an object that generates unintentional
actions. As discussed in chapter 2, the appropriate decomposition principle prohibits the object that causes the unintentional action from generating this action. A good example of an unintentional action is the detection of an attacking unit by a defending unit. The attacking unit does not want to be detected and the defending unit observes the terrain for any movement, thus which object specifies that a detection has effectively taken place? The attacking unit object cannot tell the defending unit that it has entered its detection area because this is unrealistic. The action map object fulfils this role.

The action map object manages a map of the terrain and the air which specifies all possible zones of interaction. Clearly, unintentional actions are the consequence of intentional actions. Therefore, the action map object must use those actions that may cause unintentional actions and make a prediction when these actions may occur. Basically, all actions that initiate an activity may generate unintentional actions. Each activity that a unit may execute e.g. movement, detection or combat, has an activity zone linked to it. So, the action map object's task is to simulate all intended activities without actually changing the object's state and determine the earliest possible interaction i.e. the action map object predicts future outcomes of activities as illustrated in figure 5.2. This figure shows the state of the simulation at T0. Unit A starts a movement along the given axis, unit B is an enemy unit that does not move. The detection area is drawn around each unit. The action map object simulates the movement of unit A without actually changing unit A's coordinates. At time T1, unit A enters the detection area of unit B which constitutes a possible detection action. This action is evaluated by the environment object which can reduce the size and change the shape of the detection area due to the terrain e.g. a wooded area impedes detection. The detection may be confirmed or not. A similar process is carried out for all activities and the earliest T1 is determined.

Any possible interaction is evaluated by specific environment objects which decide whether an event actually occurs e.g. detection or unit destruction. Whenever an action occurs which may trigger an unintentional action, the action map object registers it and updates its maps. The intersection between two activity zones signals the possibility of an interaction. The earliest interaction possibility is submitted to the environment object for further examination if it takes place at a time prior to the next action time e.g. an attacking unit enters a detection zone at 1012. The action map object uses a simple interaction mechanism e.g. a detection zone is represented as a circle around the unit, to define candidate actions. In order to evaluate this interaction possibility, the action map object calls upon specific

An intelligent gaming environment
environment objects e.g. when a detection may occur the action map object calls upon the terrain ridge object to define the limit of a unit's detection area which may cause part of the circle cut off. As described in chapter 2, environment objects incorporate specific evaluation functions that determine the result of an action. In this manner, the action map object is able to generate unintentional actions. The combination of objects that define their actions and an action map object that generates unintentional actions provides us with a good correspondence with reality since we can effectively determine the first next action time.

Finally, we introduce an action manager object which encapsulates all the functions that manage the list of actions, ensure the correct queueing of events and activities and solves concurrence problems.

An intelligent gaming environment
Concurrent actions are placed in the action list in a random way. However when they are handled, actions that influence the same objects are considered jointly and are grouped to form a single action. So, the dependence of actions is evaluated when concurrence occurs. For example, if unit A has decided to engage unit B, it generates a combat event at time T. Unit B may also have detected unit A and decided to start a combat at time T. Both combat actions are grouped to form a single exchange combat namely unit A versus unit B.

Furthermore, the action manager ensures that the required time contraction is enforced i.e. the simulation time evolves at the specified pace e.g. real time or 3 times as fast as real time. A decision cycle, i.e. a period of time that is allocated to players to make decisions, is defined for the lowest echelon that is carried out by human players. After each decision cycle e.g. 1 hour, the simulation model is run for a period corresponding to the length of the decision cycle multiplied by the specified time contraction factor. The decision maker is then presented with the new situation e.g. 1 hour or 3 hours of simulation time later. The simulation passes to real time mode during the decision making process of players i.e. during the period that players make decisions, the simulation advances at the same rate as real time. The simulation may also pass to real time mode when an action occurs that requires players to make decisions, e.g. a certain unit can no longer carry out its mission.

5.1.3 IALTA object interaction model

Adding the action map object to figure 2.3 and defining the relevant decision objects for the IALTA gaming environment allows us to define the interactions within the IALTA model which is illustrated in figure 5.2. It shows that decision objects, corps, division (Div) and brigade (Bde) generate orders for execution objects and/or decision objects at subordinate echelons. Execution and decision objects report their status and may send requests to higher echelons.

The orders are processed by execution objects, battalion (Bn) or company (Cie), which generate the actions that they intend to start at a given time. These actions are placed in the action list that models time by a controller and sorted in an increasing order of time.

Figure 5.3 shows that the environment objects evaluate each action, define its result and confirm candidate unintentional actions. As described in the previous paragraph, the action map object simulates activities until the next action time is reached.

An intelligent gaming environment
or an unintentional is determined and confirmed by the environment. It updates the action list through the action controller object according to the actions that the environment has confirmed.

Figure 5.3: The dynamic structure of the IALTA model

An intelligent gaming environment
The earliest action in the action list is processed by the environment object which determines the result and updates the execution objects. After every update, execution objects report the change in their situation to the decision object that manages them.

So, each action of execution and decision objects results in the actual creation of a corresponding action object. These actions are evaluated by the environment which defines their result. This construct ensures that all interactions between objects in the MMR are evaluated. Thus, describing the entire command, control and communication process explicitly. Indeed, the command and control process is modelled by the decision objects as shown in chapter 4. The communication process is simulated by message actions e.g. orders, status reports or requests, drawn as thick arrows which are evaluated by the environment and are only successful when the communications network permits it.

5.1.4 The prototype of the IALTA model

The implemented simulation model is illustrated in figure 5.4.

In order to provide a manageable basis for the simulation model, the attribute values of objects are stored in a relational data base. This ensures that the attribute of objects can be changed easily and translated into a consistent data structure. Furthermore, the data management tools provided by the DBMS allow us to prepare experiments rapidly. So, we can emphasize the functional validity of the prototype IALTA gaming environment. The actual storage of the data is handled by the data base management system. It enhances the correspondence with reality of the model by providing the necessary support to define three distinct views of the world. These views mirror the perception of players and are essential in the training of the decision making process because they force players to reason with uncertain information.

The first view is reality which is accessible to the action map, environment and simulator objects and to one type of decision object namely the controllers of the game.

The second and third view is reality as it is perceived by either side in the game. The views are populated by the player staffs who must define the certainty of data and who have to maintain the data fresh. They must plan the necessary information gathering missions to update their view or experience the effects of making

An intelligent gaming environment
Figure 5.4: The ILTA prototype simulation model

An intelligent gaming environment
decisions on the basis of incorrect data. This ensures that decision and execution objects only have access to the data which they could have available in the real world.

Figure 5.4 also illustrates the simulator object's role to create the instances of the decision and execution objects. Furthermore, the simulator object defines the length of the game and determines the interaction mode. As described previously, the game can be played at different speeds. So, when specifying the required time contraction, the game designer must also define the intervals at which players will be asked to view the situation and make decisions. Certain actions may also trigger user decisions e.g. a unit that attacks on the player's main axis is stopped. Whenever a decision is required from a user, the simulation mode reverts to real time.

Decision objects can be performed by players through the user interface or by the decision automata that are described in chapter 4. The aim of decision objects is to produce orders for execution objects i.e. units. The communication process between units results in the creation of message actions. These message actions are evaluated and sent to their destination if the communication network and the electronic warfare conditions permit it.

Units generate messages and other actions based on the orders they receive and the actions they take part in.

All actions are passed to the action controller object which ensures that the action list is structured according to the rules described above. Each time, an activity occurs, the action controller passes this to the action map object. Also specified is the maximum amount of time that the action map object may simulate activities. This amount of time is based on the action in the list that follows the activity.

The action map object finds the earliest possible action using all current activities and the maximum activity time. It simulates the activities to this time and then returns control to the action controller which calls upon the environment to evaluate the activities e.g. determine the fuel consumption for a movement or the losses for a combat.

As above, the environment object is responsible for the evaluation of each action and updates the execution objects.
5.1.5 Integrating ILTA and IATA

As discussed in chapter 3, the IALTA system is composed of the ILTA and IATA models. These models can be used separately or in combination. In order to simulate an air-land battle, we define a number of air force decision objects that generate mission actions. Those mission actions that initiate an interaction between air force and army resources are placed in the ILTA action list e.g. a Close Air Support mission that is directed at the a battalion. Whenever such an action occurs, the IATA simulation is run for a period equal to the current time minus the end time of the last IATA simulation. If the period exceeds a certain amount of time e.g. 30 minutes, the IATA simulation is run to ensure that the air force situation remains synchronised with the land situation. This approach allows us to present the current situation to the air force model when it is run i.e. the maximum deviation between the real situation and the situation presented to the air force model is T, with T being the maximum time that no interaction occurs between the air and land models. Figure 5.4 shows that when these actions take place, each simulation model is presented with that view of reality that is relevant. The current situation is described by the status of objects in the common simulation data base.

5.2 Phasing and management

We refine here the implementation approach described in chapter 3. The development of the simulation model uses an iterative approach to determine the three aspects of an object-oriented simulation namely the object’s attributes, the object’s actions and the evaluation actions. Therefore the phasing has been defined as follows:

During phase 1, we combine phases B.1 and B.3 introduced in chapter 3. We determine the complete description of the military hierarchical structure of units. We define the necessary object structures by reducing the complete structure of objects to relevant objects and attributes as discussed previously.

During phase 2, we address the first part of phase B.2 and determine the behaviour of objects.

In phase 3, the second step of phase B.2 is carried out. We focus on the specification of the evaluation actions of the environment objects.
Phase B.4 entails the development of a prototype which is used to test whether all necessary aspects of reality are represented in the model. The development of the prototype consists of two main steps namely:

1. implementation of the prototype;

2. design and execution of test experiments.

The prototype is developed from the model aspect perspective. This means that actions or evaluation actions are added to the objects in the model according to the aspect of the model that is being addressed.

Three main applications compose the prototype, namely, the model of military reality, the user interface and finally the integration with decision automata.

The prototype is developed in 2 phases with a different emphasis. The first prototype centres around the user interface to ensure that the model contains all relevant objects that are handled by players. The second prototype focuses on the simulation model and the integration of decision objects to ensure the validity of the model.

The integration of the simulation and decision automata is carried out in two steps:

1. logical integration: both systems work independently but are symmetrical as far as information exchange is concerned;

2. complete integration: both systems exchange information without the developer's interference

As appears from the priorities set for the prototypes, prototype 1 emphasises the completeness of the information flows and the user friendliness which is essential for a highly interactive system. The simulation results are less important although all necessary objects must be represented. In contrast, prototype 2 concentrates on the validity of the model and its correspondence with reality. Furthermore it will test the effectiveness of the decision automata which have to plan and manage battles.

An intelligent gaming environment
5.3 Implementation

The tools used to support this phase are:

- Excelerator and MS-Word, to document the interviews and the results.
- SuperProject Expert, to plan the different phases.
- Simulation language: C++, an object oriented extension of C.
- DBMS: Ingres.
- Graphical language: Turbo Pascal.

Ingres and C++ are chosen because they are implemented in C and can therefore be integrated using their inherent interfaces to C. We expect that the common implementation layer of C supports the achievement of the required performance of the simulation model.

The development is carried out as described in 5.2. The object hierarchy is described using the specialisation tree diagramming technique as illustrated in figure 5.1. The tree is drawn upside-down and the root corresponds to the basic object. Objects that inherit attributes and behaviours from more general objects are drawn below their parents and linked to them by a line. The action part of objects is illustrated using the data flow diagramming technique. Function symbols correspond to actions of objects, external entity symbols represent data that is passed between actions and data stores indicate the interaction with the data base. Figure 5.5 illustrates this diagramming technique. It shows the description of the combat battalion object. The action manager or controller object passes a message action to the battalion object which retrieves the contents from the data base. The order may specify that a movement must be carried out. The Mov action of the battalion receives the necessary data from the order analysis action and defines a movement action which is passed to the action manager. This movement may create a detection action. The detection action is processed by the corresponding battalion action which may decide that a combat should be engaged. Therefore data about the detected enemy unit is passed to the combat action which generates a corresponding action. All the objects identified in figure 5.1 are described similarly.
This provides us with a complete description of the action part of objects and of their interaction.

![Diagram showing the ILTA combat battalion object]

Figure 5.5: The ILTA combat battalion object

A team of seven people have worked for 18 months to develop the necessary descriptions and prototypes. The team is organised according to the object structure, e.g. 1 person works on the Action Controller, 1 on the combat Bn object, one on the environment object and 1 on the Action Map object. An average of two people have worked on the same object type. The work is initiated by a work
statement that describes the role of the object in the overall model. This document also suggests an attribute and action part for the object type. Furthermore, the phasing, the required documentation and the tests that must be performed are presented. The document is written by myself in the role of the project leader who manages the integration of all the object types and ensures that the data base structure is adapted to support the entire simulation model. The analyst/programmers develop the objects which are integrated in the overall simulation model when they are tested and validated by the cooperating military expert. This officer participates continuously in the development process. He clarifies the structure of the objects and verifies the interactions with other object types. This process results in the IALTA documentation as illustrated in RBDC [1989], RBDC [1990]. The source code represents some 19000 lines of C and C++ and a data base that contains 132 tables and 1591 attributes.

Summarising, this chapter describes the way of modelling and building a simulation model which displays a behaviour that has a good correspondence with reality. The following chapter concentrates on the support that is provided for decision objects which are humans.

An intelligent gaming environment
Chapter 6: IALTA decision support objects

As discussed in chapter 1, a war gaming environment presents three nested levels of IS-RS combinations. At each of these levels, we can identify human interaction within the gaming environment, namely, through the game designer, the game controller and the players at each echelon. This chapter describes how each of these types of interaction can be supported and which specific decision support objects are developed to facilitate this.

6.1 A game designer support environment

The main war game design activities comprise the definition of the setting of the game and the issuing of orders to players. These activities result in a design of a war game which pursues a specific set of objectives. Therefore, the game designer must also be supported in the definition of a set of control variables to manage the game and ensure that its goals are met.

Defining a setting entails the choice of the environment in which the war game will take place e.g. part of the terrain or general weather conditions. The game designer must also decide which force structure either side has and specify what their composition and strength is. The brigade decision object can be used to analyse the options that could be selected by player staffs. So, the fact that our war gaming environment includes decision automata allows the game designer to make a prediction about what may happen during the war game by simulating a set of foreseeable corps and division orders before the actual game. These orders are derived from the orders that he or she will give to the player staffs.

This should allow him or her to adjust the forces so as to ensure that the game's objectives are met within the chosen environment that is described by the terrain data.

The game outcome prediction flow is illustrated in figure 6.1.

It shows that the game designer may change three main variables to steer the outcome of the war game namely, the initial orders, the terrain and the unit situation. The simulation model is used in a fully automated mode i.e. brigade and regiment
are automated decision objects. The evolution of the battle can be monitored through a graphical interface which displays a map with the relevant data. As described in chapter 5, specific significant actions, e.g. unit destruction, can be defined that cause the simulation to report to the decision maker whenever such an action occurs. The time contraction then changes to real time mode. The game designer may analyse the history of the war game since all events are stored in the simulation data base. The war game can be re-started with a modified unit situation in order to evaluate the influence of reinforcements on the outcome of the battle.

Figure 6.1: The outcome prediction flow

An intelligent gaming environment
So, the game designer requires support for his information gathering process. The tools that are designed to manipulate terrain, unit and order data are described in the following paragraphs. Situation assessment is facilitated by a data presentation and explanation utility. The simulation model supports the game designer’s option generation and decision making processes.

6.1.1 The IALTA terrain management module

An army war game is performed in a specific area of terrain which conditions the movement of units and therefore the operations that can be carried out. The terrain management module allows a game designer to define this area or select it from an existing data base.

The terrain is modelled as described in the previous chapter. So, our model of the terrain is a discrete representation i.e. it is a set of points. Points may have only one surface characteristic, may contain a crossing, indicate a difficult area and display weather sensitivity. They may also be part of one or more line terrain features e.g. good and/or bad road. This reduction of reality allows the definition of terrain at any required resolution e.g. 1:50.000 or 1:250.000. These resolutions are supported because they are relevant to the level of detail required by the echelons of the decision objects. Planning at corps and division echelon is carried out on 1:250.000 maps, whereas brigade staffs mostly use 1:50.000 maps.

The game designer uses a dialogue interface to select the resolution of terrain, he or she wishes to employ. Terrain data may be added, updated or simply selected. Data is stored in a relational data base which contains a table for each terrain characteristic. A general data model has been designed that describes point data by their UTM coordinates and lines by their unique identifier, point sequence number and UTM coordinates. For line types that have supplementary attributes, e.g. roads, an extra table exists which contains the line’s identification and other attributes. Terrain data can be added or updated through a specific graphical interface which is both mouse and digitizer driven. Figure 6.2 shows the structure of terrain management module. Figure 6.3 illustrates the user environment that is designed for the terrain management module. The terrain data base is stored on a server workstation because it must be widely available on the IALTA Local Area Network (LAN). The user has a Personal Computer (PC) that executes the dialogue and graphical interface. Within the graphical interface, data can be manipulated with a
mouse or with a digitizer that has the same menu as appears on the computer screen.

Figure 6.2: The terrain management module

An intelligent gaming environment
Selected terrain data can be used to produce a map background to display unit data that can be managed through the unit data base management system as shown in figure 4.15.

![Diagram of TMM user environment]

Figure 6.3: The TMM user environment

6.1.2 The IALTA unit management module

The unit management module is shown in figure 6.5. This type of interface presents a user with data in a format that corresponds to a form as illustrated in the following figure. Unit descriptions and hierarchy are stored in a relational data base through form based interfaces as shown in figure 6.4. The game designer may handle this data through a dialogue interface. So, units can be handled separately and in detail through this interface.

An intelligent gaming environment
A more general view of the situation can be provided through the specific graphical interfaces that are discussed later.

<table>
<thead>
<tr>
<th>Commanding Unit</th>
<th>Unit Id</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (BE) Corps</td>
<td>15 Div</td>
<td>Inf Bl</td>
</tr>
<tr>
<td>15 Div</td>
<td>15 Bde</td>
<td>Bl</td>
</tr>
<tr>
<td>15 Bde</td>
<td>15 G</td>
<td>Bl</td>
</tr>
</tbody>
</table>

*Figure 6.4: A form-based interface*

Each army unit has a map background associated to it which allows the user to display the unit's entire hierarchy of command i.e. a tree which describes the subordination of units to the one being examined. In displaying this hierarchy zooming in on terrain sections is automatic when the selected unit has another map associated to it than is currently displayed.

The airforce situation can be presented in a selective manner as discussed in chapter 4 and illustrated in figure 4.20 through 4.22. It allows the game designer to assess the capacity of both forces to carry out their missions. Furthermore, the army units and important terrain features, e.g. bridges, can be displayed which supports the integration of air and land operations.
Both graphical map based representations can be output on a hardcopy facility which enables the game designer to include the situation overview in the order that is given to the player staffs. This facility is also used to verify the data that is entered through the dialogue interface since errors in coordinates are much clearer on a printed map. The terrain data is plotted on a transparent sheet that can be overlaid on the real map. In this manner, distortions or mistakes can be traced. It appears
that distortions are very limited, a maximum distortion of 2 mm has been noticed on 40 by 60 km area that was checked.

6.1.3 An order definition support environment

An important part of the work of the game designer and in fact of any staff officer is the definition of orders which entails the production of text in a pre-formatted document and the drawing of situations and intentions on maps that are generated with the terrain data base. Therefore, an order definition support environment is developed which contains the description of orders and a graphical library to produce all military symbols. Its structure is shown in figure 6.6.

![Diagram](image)

*Figure 6.6: The order management module*

An intelligent gaming environment
6.2 A game controller support environment

The game controller's task is to ensure that during a war game the situation evolves according to given objectives. We state as a general rule that any intervention of the game control staff should be realistic. Moving a unit from one position to another in one instant is considered unrealistic and falsifies the correspondence with reality that the war game pursues. So, when a deviation between the situation of the war game and the expected situation occurs that is too great, we consider that the war game should be stopped and re-started on a modified basis. The player staffs are informed of the reasons for reviewing the game and are placed in a new situation. A good degree of correspondence with reality can thus be achieved.

The information gathering and situation assessment of game controllers is supported by the already described unit management module. It should be noted that game controllers have access to the real situation data, blue's perception of red and vice versa. Decision automata and data on the perception about the opponent can be used to predict the courses of action that are possible and the simulation model can be employed to predict the short term evolution of the situation. This foresight may be used to adapt perceptions of player staffs or change the orders or support they had received previously.

6.3 A player support environment

6.3.1 An army player support environment

Players at any echelon must be capable of monitoring the evolution of the situation i.e. they must be able to gather information. A dialogue interface with the unit data base is provided. It should be noted that neither side has access to perfect information about their enemy. Specific views on the data base are created which represent the perception of each side. Enemy units are described with a certain degree of uncertainty that is decided on by players who are responsible for the intelligence analysis.

Situation assessment is supported by a graphical interface which can display both forces and include the measure of uncertainty of data. Furthermore, the hierarchy of units can be illustrated by pointing on a unit and selecting the hierarchy option as shown in the following plot of the display.
The selected unit is split up in its composing units and may be displayed on a more detailed map. Each unit has a map associated to it. Maps are generated from the terrain data base. A unit has a tactical, logistical and intelligence map. The tactical map shows a unit’s sector with all its depending units and perceived enemy units. The map background contains those features of the terrain that are tactically relevant. The logistical map displays other terrain data e.g. roads than the tactical map, and covers a wider area to show the overall logistical installations which support the unit. The intelligence map is similar to the tactical map but is extended towards the enemy sector and the neighbouring units to show more data about the enemy situation and reserves. This should support the player’s assessment of enemy tactical possibilities.

Option generation of manoeuvre plans is supported by providing the players with the terrain analysis modules that are developed for the brigade decision object. The result of these models is integrated with the graphical interfaces that were previously discussed. Furthermore, the players have the ability to draw military symbols on the map backgrounds as discussed in paragraph 6.1.3.

It is envisaged to support decision making by providing the players with the ability to simulate their plans against their perceived enemy. This is currently not supported.

6.3.2 An airforce player support environment

As described previously, the airforce situation can be monitored through a graphical interface. The important aspect is that it supports both situation assessment and option generation.

For the air defense aspect, the impact of losing detection capability in the form of radars can be evaluated by the drawing of a radar coverage diagram as shown in the following diagram. Coverage can be shown at three heights and gaps may be identified which can be used to penetrate the airspace without being detected.

Offensive mission planning is also supported by a specific option which highlights all airbases which have squadrons that are capable of reaching the target the player is pointing on.
Figure 6.7: The IALTA airforce radar coverage diagram

An intelligent gaming environment
Furthermore, the airforce decision automata are integrated in this environment. So, decision automata are used as decision support objects that advise players in their situation assessment, option generation and decision making process. Again these results can be displayed with the graphical interface. Decisions can be accepted or changed and recorded in the data base.

6.4 Implementation

The decision support objects discussed previously are developed in an iterative manner in very close cooperation with military experts.

Military symbology is used to display the simulation objects in order to provide a familiar environment. Most operations are mouse-driven and require a very limited use of the keyboard in order to facilitate the interaction with the interface. Maps are used in all the interfaces because they constitute the officer's decision making environment i.e. a map with relevant data is the main tool that supports an officer's decision making process. This is complemented by data base interfaces that present forms with the necessary detailed data. So, the map based representations are the tool to form a global view of the situation, devise a set of solutions to the decision making problem, evaluate them and choose between them. The data base interfaces provide the complementary detailed view of the situation.

Objects that are displayed are mostly derived from the decision objects except for the order structure that is defined in chapter 5. This dialogue between decision and execution object has been formalised to support order generation by human decision objects i.e. players staffs.
The overall implementation concept centres around a common data repository namely a data base. All applications e.g. graphical interfaces, simulation model, receive and send data to it. The graphical interfaces represent some 35000 lines of Turbo Pascal code and the terrain data base contains 72 tables with 351 attributes. A total effort of 60 man/months are spent on this development that has been carried by myself as project leader and 8 analyst/programmers over a period of 2.5 years.

Our model is implemented on a distributed hardware architecture. From a cost-effectiveness, performance and robustness standpoint, a distributed architecture is preferable to a centralised architecture. Centralisation of processing capability requires a major infrastructure investment e.g. special facility with cooling. Distribution of processing capability enables the players to access and manipulate data more freely and independently e.g. a failure of the main computer does not influence the player’s working environment. An important consideration in choosing a distributed architecture is the linking of the processing capability with the software tools that are used in a given place and function. A schematic of the distributed architecture is shown in the following diagram. The cost of the architecture is also relevant in our choice. Various architectures are evaluated on the basis of the technical requirements that we defined in RBDC [1988]. It is concluded that a centralised architecture is more expensive in its initial cost and in its life cycle cost. Furthermore, all the required functionality cannot be provided, e.g. many basic user support tools like word processors or spreadsheets are not as user friendly on large computer systems as they are on PC’s. Also, most users already have some experience with PC’s from their other professional activities. So, when they participate in a war game, the player environment is more familiar if it is based on a PC.

So, players have a PC-based environment and are connected in a Local Area Network (LAN). This LAN provides them with a staff decision support system which is based on a staff common data and tools server. Furthermore it allows players to communicate with each other. It should be noted here that all communications are simulated explicitly. All message actions that are defined in chapter 5 are passed through environment’s evaluation actions. So, the simulation model regulates the available communications across the LAN based on the conditions of the battlefield e.g. due to jamming a division commander may lose communication with a subordinate brigade, the message actions between both staffs are unsuccessful and therefore not entered on the LAN.
Figure 6.8: The IALTA architecture

An intelligent gaming environment
Chapter 7: Experiments

7.1 Experimental design

In order to test our hypotheses, we have to model a part of reality which is considered complex. Furthermore, we want to be able to illustrate the modelling of individual and group decision-making processes. The military environment combines a complex sub-system of reality e.g. terrain, airspace and all the different types of unit and weapon systems, with ill-structured problems that must be solved as discussed in chapters 1 and 2 and illustrated in chapters 3, 4 and 5.

As already discussed, we have chosen a case-oriented approach in the military domain.

The first step in our experiment is to develop a simulation model of an integrated air-land battle at a tactical level of decision making. It is an open model with 2 or 3 levels of decision making enacted by players. The subordinate command echelons are represented in the model using the automated decision and execution object concepts defined in the hypotheses of chapter 2. The development approach and process are described in chapters 4, 5 and 6.

The second step entails the evaluation of the correspondence with reality of the gaming environment that is developed. A group of experts in the field of war gaming and a group of military experts participate in the evaluation process and are asked to analyse the IALTA gaming environment using the factors of correspondence defined earlier in chapter 2, namely, experimental validity, military realism, credibility, flexibility, resources required, responsiveness, sensitivity of the model, visibility to the user and transparency.

They are also encouraged to make a general statement about the gaming environment. The military experts are from the RBDC and from visiting NATO member countries. The war gaming experts are from different NATO and national scientific agencies.

In order to perform the necessary tests, we design four war games that use the gaming environment according to the objectives that are described in chapter 3.
7.2 The ILTA game

The first type of war game that the RBDC wants to perform is an army war game that is directed at the students of the senior staff course. In this type of game, the emphasis is put on the quality of the decision making at the corps and division echelons. A second type of war game is more procedure oriented and requires human players at the division and brigade echelons.

So, firstly, the war gaming environment must support the command and control process of division staffs by providing them with brigade staffs that execute the orders they receive correctly and that generate the necessary feedback and dialogue. It is also required that all support staff function be carried out by a computer-based model to reduce the cost of war games and increase their frequency and availability. This means that the brigade decision automata are called to generate manoeuvre plans and conduct the battle for the brigade decision objects as described in chapter 5 figure 5.3.

Secondly, the war gaming environment must support the work of brigade staffs as they would perform it in reality. So, the evolution of the situation must be presented in a familiar manner to players and communication between staffs must apply current procedures and message formats.

The following paragraph describes the war game that is designed to test these requirements.

7.2.1 Organisation of the game

Figure 7.1 shows the organisation of the war game.

The blue team consists of one corps, division and brigade staff which is enacted by one person. The red team is also played at these echelons and with the same number of people. Brigade decision automata are called by the corresponding decision objects to emulate this echelon’s decision making process. Concurrently, the same brigade is enacted by a player who carries out the equivalent decision making process supported by the tools and data that are contained in the IALTA environment. So, this organisation enables us to combine the testing of the brigade user interface and the evaluation of the decision automata’s results.

An intelligent gaming environment
The staff, logistical and part of the tactical decision automata are also tested during a war game of the RBDC that is intended to practice procedures and that is played manually. Within this division-brigade game, a specific brigade staff played by officers is chosen to compare the results of its decision making with the results of

![Diagram](image)

*Figure 7.1: The organisation of the ILTA game*

An intelligent gaming environment
the automata that attempt to replicate the decision making.

7.2.2 Testing characteristic scenarios

A typical scenario is played which opposes two divisions. Each division consists of a brigade. The blue brigade is composed of three combat battalions, two artillery battalions, an engineering company, a supply company, a transport company and a medical company. The red brigade is composed by two combat battalions. Since blue is the attacking force, a greater number of tank assets are allocated to it than the red force. An overall force ratio of 2:1 in favour of blue is implemented which enables blue to attack but will probably cause it some problems as time progresses.

In this scenario, blue attacks red which fights a delaying manoeuvre. This defensive posture combines static defense and delaying combat which makes it representative for the evaluation of defensive actions.

The game is conducted as follows:

a. Players give an order to the brigade decision object which calls the different decision automata and coordinates their results. In parallel, experts make a brigade plan. The plan generated by the decision automata is presented for evaluation to the military experts.

b. Both plans are transformed into orders for the execution objects and combat is simulated for 24 hours.

c. The history of the simulation is traced through the player's graphical interface which allows the experts to understand and evaluate the dynamics of the simulation.

This type of scenario is played through in a 40 by 60 kilometre area which is part of the 1 (BE) Corps' sector in the Federal Republic of Germany which includes all the different terrain types defined in chapter 5. Several different sectors are chosen in different directions to test the flexibility of the decision automata. This area is chosen because it is very well known by the military experts. They have a close knowledge of the terrain which enables them to make an evaluation of the terrain representation that is used in the TALTA war gaming environment. Furthermore, they have a good experience in the planning and conducting of operations in this area because they
have participated in many exercises. So, they are able to test the conclusions of the
decision automata very critically and make a judgement about the dynamic evolu-
tion of the situation.

This war game combines both types of game described in chapter 3, namely, the
corps-division game that is aimed at training decision making and the division-
brigade game that focuses on the training of procedures. For the corps-division
game, the decision automata carry out the brigade decision-making process. For
the division-brigade game, brigade decision making is carried out manually and the
interaction with the execution objects is supported by the user interface.

7.2.3 Experimental results

The decision automata are tested in two games. The first test centres around the
staff and logistical decision automata. The ability to emulate a brigade’s staff decision
making process is evaluated.

The staff decision automaton handles the orders from the division correctly although
the orders contain certain errors, namely, insufficient time is allocated to carry out
the necessary movement. The decision automaton responds that it can not solve
the problem and that it expects a delay in the starting time of the brigade’s
manoeuvre. Human decision makers concur with this finding and find the solution
realistic. A problem has appeared on the input side because the automaton can not
accept certain constraints. The new type of constraint and the rules to process it
are added and further tests are performed. The staff decision automaton behaves
realistically in these conditions. Specific features are added to the graphical inter-
face to make the rhythm of the movement more understandable, i.e. a graph that is
commonly used by staff planners is added.

The logistical decision automata are tested in the same exercise and appear to
behave realistically. Some differences with the player’s solution occur when
shortages in evacuation capacity are encountered in the maintenance domain.
Discussion with the RBDC’s teachers show that the decision automaton’s solution is
acceptable and equivalent with the player’s. In this context, the explanation facility
of the knowledge base shell proves very helpful because the decision making
process can be traced easily.

The other decision automata are tested in the game described above.

An intelligent gaming environment
The intelligence decision automaton behaves realistically in the definition of the information gathering plan and related missions. The capability of identifying the enemy is still insufficient. Too much corroborating data is required for the automaton to identify an enemy unit. The knowledge in this field must be defined further to cope better with uncertainty.

The communication decision automaton is found realistic by experts.

The artillery decision automaton plans fire allocation and movements correctly.

The engineering decision automaton plans the allocation of resources in an acceptable manner but requires more detail in the terrain analysis. Engineering requirements are too frequent because all the terrain details are taken into account.

The tactical decision automaton is extensively tested in different sectors. Five manoeuvre options are generated for offensive operations. A best manoeuvre is generated for the delaying manoeuvre within the same sector. The terrain analysis and axis finding processes are found acceptable. However, the automaton does not use partial side accesses sufficiently. The resource allocation process is acceptable in most cases but lacks some rules when the number of battalion commands - number of available companies ratio exceeds a 1:2 ratio.

The unit creation, battalion order analysis, event handling, combat and detection processes of the objects in the simulation are tested and found realistic. They provide a good view for the player of what is happening on the battlefield. The interaction process with the simulation requires more attention because players find that they cannot interact sufficiently and especially frequently enough. It is suggested to use shorter decision cycles or increase the actions that trigger user interaction.

The order definition interface is found satisfactory.

The form-based data base interfaces are acceptable but require more simplicity. An increased "help" functionality is required. Text files that provide this support are added.

An evaluation of the IALTA environment using the criteria defined in chapter 2, is described in 7.4.
7.3 The IATA game

7.3.1 Organisation of the game

Figure 7.2 shows the organisation of the IATA game.

![Diagram showing the organisation of the IATA game](image)

*Figure 7.2: The organisation of the IATA game*

An intelligent gaming environment
The IATA game is aimed at the airforce players who have to practice their procedures and decision making capabilities at ATAF, ATOC, SOC and ASOC echelons. The opponent decision making is entrusted to the corresponding decision objects which apply their decision automata.

7.3.2 Testing characteristic scenarios

Two different war games are played with specific scenario’s.

The first type of war game is oriented towards the training of procedures. This concerns the definition of plans and orders according to NATO procedures. During the game, the emphasis is mostly on offensive operations and more particularly on the support that the airforce can provide to the ground troops because it is played concurrently with the army division-brigade game. The game is played for 3 consecutive days in real time mode with a varying number of resources. Different types of operation must be carried out, e.g. OAS, OCA, Recce, during this period within the usual area of operations.

The participants have some experience in the planning of air operations in a non-automated manner. The test with increased automated support is interesting because it illustrates the use of different tools and provides insight in the physical organisation of the player environment, e.g. the number of PC’s that are necessary. About forty officers whose grade ranges from captain to colonel, participate in this game.

The second type of war game is oriented towards decision making. This game puts an increased emphasis on air operations that pursue the air superiority, e.g. OCA and air defense. Experienced decision makers participate in the evaluation of the planning tools. Furthermore, they are able to generate realistic plans and compare the results of the simulation with the results of exercises in which they have taken part. They are also able to evaluate the quality of the decision automata’s planning. Three teachers of the RBDC participated in this exercise which requires a limited amount of hardware support.

Both war games are played in the familiar area of operations, namely, the central region. The resources that are employed are known and the disposition of resources on airbases mirrors reality. Quantities of resources are varied based on the objectives of the war game.

An intelligent gaming environment
7.3.3 Experimental results

As discussed previously, the first type of war game corresponds to a procedural game. Operations are planned and carried out as prescribed by current procedures, the necessary orders and briefings are given.

A support system for order definition which combines an access to the resource data base and word processing capabilities and produces a completely formatted order, is tested. It is noticed that users are insufficiently familiar with the support system which causes unacceptable delays. Too much time is invested in the learning of the support system because users perceive the advantages of formulating the contents of the order in relation with the resource data base that is also accessible to the subordinate echelons. So, the users revert to a more commonly used support tool, namely, the word processor and the allocation of resources is entered in text format and sent as a message to the subordinate echelons. We conclude that the support system should be used in support of courses in order to provide insight in the advantages of such systems and to familiarize users with the system structure and capabilities.

A commercial electronic mail system that is provided with the LAN is also tested. It allows users to send, receive, edit and update messages in an integrated environment. It appears that the learning process is rapid and that the general functionality is sufficient. However, it has become clear that users require an explicit acknowledgment of the receipt of their messages. Given the technical solution that supports the system, the fact that the message can be sent successfully is sufficient to know that it is actually received. The user’s mail box is namely on the electronic mail server. So, when the server is accessible, the service is available and all mail boxes are accessible. This is not clear to the user who is unaware of the underlying mechanisms, so it is decided to add this acknowledgment message to the electronic mail system by using a send command that provides it.

Also evaluated is the data base dialogue system which includes on-screen form-based interfaces and hardcopy output of data in report format. Users conclude that reports must have the same format and sorting as the on-screen interfaces. Furthermore, reports are required that provide more general overviews, e.g. the percentage of OCA missions that are successful.
The graphical interface that is described in chapters 4 and 6 is tested in this game. The complementarity between the graphical presentation of the situation and the form-based data base interfaces is thought very useful. Also appreciated are the mission planning support functions like the drawing of aircraft ranges. However, increased selectiveness is required to support mission planning. Planners use specific types of aircraft to carry out certain missions, so only those assets should be displayed. The assets that are in range of the requested target can easily be determined if this feature is supported. A selection option is added to the graphical interface which enables the user to select on the type, role and subordination of squadrons. This version of the interface is tested during the decision-making oriented game described below.

Concerning the physical organisation of the user environment, it is noticed that the number of PC's that is provided, 1 or 2 per staff, is too limited. Each main function, e.g. OCA or OAS, requires a separate PC and even two because a combination of the form-based and the graphical interface is considered very beneficial by players. A multi-windowed environment which allows the display of several full sized screens would be more suitable because it only requires the manipulation of one machine.

The decision making oriented war game uses the same tools to plan operations as in the game that is aimed at training procedures. However procedures are not applied rigourously, e.g. the required order formats are not used. So, most operations are planned directly through the data base dialogue interface without the help of the order definition support system. The objective of this game is to test the capability of the IATA war gaming environment to plan air operations effectively and rapidly, to simulate an air battle realistically and to present the results comprehensively. Tested are the form-based dialogue interface, the enhanced version of the graphical map based interface, the simulation model and the decision automata for the red side.

The military experts conclude that the form based dialogue interface is sufficient for planning and decision making. However, it is felt that still too much time is spent on detailed planning due to the mission level resolution of the simulation model. The tasking decision making process should receive more automated support in order to be able to dedicate more time to the analysis of major command decisions, e.g. which resources are allocated to which mission types and for how long. It is suggested to use the air mission planning decision automata to execute the tasking functions.

An intelligent gaming environment
The map based display's mission planning enhancements, namely, the extended selection capability described above and the identification of the squadrons that are in range of a given target are considered satisfactory and sufficient to support the player's decision making process. Combined with the form-based data base interface and the decision automata, the graphical interface can support each step of the mission planner's decision making process. Information gathering is supported by the form-based interfaces, situation assessment by the form-based and the graphical interfaces, option generation by the graphical interface and the decision automata, decision making by the decision automata.

The drawback of using a PC-based display is the resolution of the screen, namely, 640*480 pixels, which distorts the presentation of data and especially of the radar coverage. This disrupts the players situation assessment, i.e. what is always displayed as a circular area on a map is drawn elliptically on the screen.

The validity of the simulation model is tested and is considered satisfactory by the military experts. The integration of an existing simulation model, the Theatre Air Wargaming System, in the IALTA gaming environment causes certain interfacing delays which slows down the execution of the game.

The decision automata are tested in the red team which is manned by 1 person who determines the roles of the red assets and the priorities of target types. The results of the defensive and offensive planning decision automata are judged acceptable. The defensive resource allocation process is lengthy and a performance improvement is found necessary.

The formal evaluation of the IALTA war gaming environment is presented in the following paragraph.

7.4 The evaluation of IALTA

The following questionnaire is submitted to four military experts with a background in war gaming to evaluate the IALTA gaming environment. The questions are derived from the war gaming evaluation criteria defined in chapter 2 and can be marked on a scale from -- to ++ with the following meaning:

An intelligent gaming environment
-- : no
- : not likely
-/+: don't know
+ : in most cases
++ : yes

The definitions of the previously mentioned criteria are provided with the questionnaire.

Questions:

1. Flexibility

Can the war gaming environment be used for different applications, e.g. training or plan analysis?

2. Experimental validity

- Can the war games be repeated a sufficient number of times to support training purposes?

- Can the war games be repeated a sufficient number of times to support analysis purposes, namely to evaluate different cases and provide an acceptable degree of uncertainty?

3. Military realism

- Are the characteristics that describe the simulation objects sufficient to model their counterparts in reality?

- Is the behaviour of simulation objects sufficiently close to its counterpart in reality?

- Are actions generated realistically?

An intelligent gaming environment
- Are actions processed realistically?
- Are actions evaluated realistically?
- Is military decision making sufficiently modelled?

4. Responsiveness
   Is there sufficient possibility to interact with simulation objects compared to the interaction that is possible in reality?

5. Sensitivity
   Is the model sufficiently sensitive to variations in the initial situation?

6. Resources required
   - Is the required hardware configuration well related to the amount and types of war games that must be supported?
   - Is the required software configuration well related to the amount and types of war games that must be supported?
   - Is the required software engineering personnel well related to the amount and types of war games that must be supported?
   - Is the required support staff personnel well related to the amount and types of war games that must be supported?
   - Is the required player staff personnel well related to the amount and types of war games that must be supported?
   - Is the required game control personnel well related to the amount and types of war games that must be supported?

An intelligent gaming environment
7. Openness

Is there sufficient flexibility to choose the echelons that require personnel based on the purpose of a war game?

The answers to the questionnaire are shown in figure 7.3.

| Question | 1   | 2a  | 2b  | 3a  | 3b  | 3c  | 3d  | 3e  | 3f  | 4   | 5   | 6a  | 6b  | 6c  | 6d  | 6e  | 6f  | 7   |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1        | ++  | +   | +   | ++  | +   | ++  | +   | ++  | +   | -/+ | +   | ++  | +   | ++  | +   | +   | +   |
| 2        | +   | +   | +   | ++  | -/+ | +   | +   | +   | ++  | -/+ | +   | ++  | +   | ++  | +   | +   | +   |
| 3        | ++  | +   | +   | +   | +   | +   | -/+ | +   | +   | -/+ | +   | +   | -   | ++  | +   | +   | +   |
| 4        | +   | +   | +   | +   | -/+ | +   | +   | +   | -/+ | +   | +   | -   | ++  | -/+ | +   | +   | +   |

* : Good   .*: Fair   _ : Negative

Figure 7.3: IALTA questionnaire results and conclusions

Results are aggregated on a scale of good, fair and negative for each separate question. Results are further aggregated for each criterion.

Comparing these results with the evaluation of the TELEBATTLE system discussed in chapter 3, we can state that an overall improvement is achieved and that globally an intelligent gaming environment
the IALTA environments meets its requirements since the situation is simulated in an acceptable manner and personnel is reduced to the actual players. Especially positive are the remarks about the provided level of flexibility, experimental validity and openness which are directly related to the objectives of our research.

Military realism and responsiveness are considered less positive. The problem with the evaluation of military realism centers around the lack of actual experience of experts with a large conflict that involves the echelons played in this type of war game. The behaviour of individual units is considered sufficient and good but the overall behaviour of a corps or ATAF is much more subjective and less founded on experience since experts have not conducted actual operations at this echelon.

The required resources are considered well related to the complexity of the gaming environment except for the amount of personnel. It is noted here that development is still on-going and that a war gaming environment in production mode requires only half of the current amount of people.

An analysis that we carried out showed that a production environment of the IALTA war gaming environment could be supported by a team of seven people which is about half of the current 15 man team.

This chapter presented the tests that are carried out with the IALTA war gaming environment. They have shown that the concepts introduced in chapter 2 allow us to support several types of war game as required by the RBDC. The following chapter discusses the general conclusions that can be drawn from the development of an intelligent gaming environment and highlights possibilities for further research.
Chapter 8: Epilogue

In chapter 1, we stated that we intended to define a method of thinking, modelling, working and managing for the development of intelligent gaming models.

Having chosen the paradigm of bounded rationality as our way of thinking and interpreting reality, this dissertation shows that a complex part of reality can be simulated with a good degree of correspondence and flexibility. By encouraging us to consider decision making as a process that is bounded by human cognitive capabilities, this paradigm ensures that we model decision making as an explicit activity in the simulation of reality. This enhances the correspondence with reality of our model as illustrated in chapters 4 and 7. Considering the different activities that occur in reality, we develop a way of modelling that emphasizes a functional specialisation of an object oriented model of reality as discussed in chapter 2. It has demonstrated its value in the simulation of a complex part of military reality as discussed in chapter 5. Chapter 7 shows that this modelling approach enhances the flexibility of the war gaming environment because decision objects can either be human players or decision automata. So, the war gaming environment can be configured as required for the purpose of the war game. The cost effectiveness of war games is also improved because players for which the game is not intended are replaced by decision automata i.e. an entire staff or part of it may be replaced by decision automata. As discussed in chapter 7, decision automata can match human decision making in a war gaming environment. So, the same level of correspondence with reality can be obtained by using decision automata instead of players. Furthermore, since decision making is modelled in a deterministic manner and decision automata are able to replace human players, war games can be configured as closed simulations and be repeated a sufficient number of times to achieve the necessary statistical validity in order to draw conclusions from the game outcomes. So, we may say that our modelling approach enables us to build a war gaming environment that can be used for training and for analysis purposes. As discussed in chapter 1, a multi-purpose war gaming environment must allow variation on strategy, force structure and weapon mix. The following example demonstrates that the concepts used for the development of the IALTA environment support this variation.

*Variation on strategy is achieved by the decision objects which are decision automata or human actors as described in chapter 4. These objects execute*
the rules of strategy that are defined at the beginning of the exercise. Thus, testing other strategies can be achieved by providing the decision objects with new decision rules and/or structures or by providing these same rules to players. In the simulation these objects interact with all other objects in the same manner independently from their implementation. Therefore flexibility is ensured. Suppose a tactical rule which says that when a battalion requires combat support the commanding brigade tries:

a. to provide artillery support;

b. to provide Close Air Support from fighters;

c. to commit available reserves.

With the advent of very effective combat helicopters, brigade combat support tactics are adapted to use these resources as soon as possible. This can easily be achieved by inserting this rule in the previous one with helicopters receiving first priority. Chapter 4 shows our ability to model this reasoning process. Chapter 5 illustrates how these tactical rules can be translated into concrete actions by inserting them in the description of the corresponding execution object.

Variation on force structure can be achieved by providing sufficient flexibility of the hierarchical command structure. The experiment’s designer can decide upon the subordination of execution objects to the different command echelons. Furthermore decision objects can be provided with the necessary intelligence i.e. decision rules, to command their assets i.e. execution objects. Using the same example, the helicopter resources might be distributed from the Corps to the brigades which become responsible for their use. Therefore the brigade decision object must no longer generate a request for combat helicopter support from the Corps resources but can plan their missions using the combat helicopter decision automaton. The knowledge base contains the necessary rules to plan the use of the available resources in the context of a brigade’s mission.

Variation on weapon mix is achieved through the combined use of the execution and environment objects. When adding a new weapon system it suffices to describe its characteristics and behaviour and its influence on other execution
objects in a manner that is usable by the environment. This allows the environment to evaluate its effect within the overall combat simulation. Again using the previous example, adding combat helicopters to the model would result in adding an airborne object which has certain behavioural rules and whose actions are evaluated by the environment, e.g. terrain, weather.

So, our modelling approach supports the development of multi-purpose war gaming environments. Looking at the development of decision automata, this thesis confirms the knowledge structuring principle as a useful guideline in the modelling of complex decision making processes as discussed in chapters 4 and 7. Furthermore, the communication principle is shown to be a sufficient guideline to coordinate the different independent decision making processes that manage the use of resources. The advantage of designing separate decision automata for independent knowledge domains is threefold:

- it offers maximum game flexibility, because each problem area can be solved by either a human player or a decision automaton as said previously e.g. an entire staff can be replaced by a decision automaton or just certain specific functions within that staff; this can apply to one or several command echelons;

- it permits a more effective evolution of the decision automaton because for every problem domain the concerned expert or group of experts can directly be identified and queried;

- it makes for a clearer design of the decision automaton that coordinates the separate automata. This is more easily understood by experts as well as new participants in the development because staffs are organised and operate in a similar way.

Concerning our way of working, the development of the IALTA war gaming environment shows that achieving a constructive dialogue between designer and domain expert is essential. Speaking the expert's language or jargon and acquiring a basic understanding of the problem domain proves important in starting this dialogue.

It appears that representing decision making processes or object attributes and behaviours in the form of diagrams as illustrated in chapters 4, 5 and 6 does not constitute the correct vehicle for experts to validate our models. It seems that the reason for the ineffectiveness of diagrams to carry the dialogue between designer
and expert is the fact that experts do not model their decision making processes in a clearly defined stepwise iterative manner. The object construct corresponds more closely to the expert's view of reality but the diagrams are too unfamiliar to be useful.

This dissertation illustrates that a prototype does provide the necessary vehicle to accommodate the dialogue between expert and designer. For decision automata, the prototypes must have an explanation facility to describe their reasoning path clearly and the conclusions must be presented in a form that experts are accustomed to. In the case of military officers, a graphical representation of units in the terrain with appropriate symbols as shown in chapter 4 is appropriate. This feature which we call the familiarity aspect of the user interface, is essential for the validation process. Furthermore it encourages participation and it is paramount in the overall acceptation of decision automata and of the dynamic model of reality by personnel that has not participated in the development. In this context, we have noticed that the completeness of the set objects that can be shown and the selectiveness with which objects can be displayed are more important than the actual ease of use of the graphical interface.

Having defined the necessary vehicle to develop a gaming environment successfully, it has appeared that development cycles of prototypes tend to become shorter as the problem domain that is being modelled becomes more familiar to the modeller and the possibilities of the tools become more apparent to the experts. Experts go through a learning process that is illustrated in figure 8.1. Experts tend to oversimplify the problem domain in the early stages of the development because they consider the complexity of the problem to be too great for the designer to model. As the modelling possibilities become apparent through the prototypes that are developed, experts introduce an increasing degree of complexity. As mentioned above, development cycles shorten and good progress is made in covering the problem area. However, a point is reached where problem situations are addressed that virtually never occur and that require a lot of effort to model. Increased prototype development cycles are an indication that such a point has been reached. At this stage, it is important to refer to the goal of the prototype that is being developed and evaluate the need to model that part of the problem. This process is noticed during the development of every prototype regardless of the level of complexity of the problem being modelled. It constitutes an important aspect of the management of such a project. As stated earlier, participation is increased by building prototypes. However, users and experts must be made aware that prototypes are there to be criticised and thrown away if necessary. Once this has been achieved, overall
acceptation of the prototype is best obtained when the experts present the developments and the developers only act as technical consultants.

The prototypes of the IALTA war gaming environment are developed with the continuous support of military experts. This means that interaction occurs on a

An intelligent gaming environment
frequent almost daily basis. Therefore a strong commitment must be made by the management of the experts to ensure that sufficient time is available to cooperate in the development. The tools that are used in the development of the IALTA war gaming environment support this way of working. However, the integration of the different tools requires a considerable amount of effort and a more integrated development environment would probably decrease development cycles significantly.

As discussed in chapter 4, 5 and 6 such a development represents a considerable software engineering effort and involves a medium-sized development team. Therefore, it requires a non-trivial amount of management to ensure the continuity of the work and the documentation of completed prototypes. A stable project management structure is essential in achieving this.

More generally, chapter 7 illustrates the level of user satisfaction with the war gaming environment and its flexibility to accommodate new requirements. Based on the limited survey that was held at the RBDC, we can conclude that the approach presented in this thesis supports the development of war gaming environments that can be used for training and analysis purposes at different echelons. Furthermore, analysis can be supported with the required amount of experimental validity since different types of interaction with the model are supported.

Although an increase in military realism is achieved, it is difficult to judge whether the required level of correspondence with reality is obtained since the reality that is being simulated has not been experienced by military experts. Therefore a validation problem for this aspect will continue to exist for the field of war gaming. Applying our concepts for the development of a gaming environment for a verifiable part of reality could provide us with a clearer insight into this aspect.

Concerning the organisational problem discussed in chapter 2, we are capable of replacing support staffs by decision automata which increases the cost-effectiveness of war games. However, experts consider the increased amount of model development and support personnel as less positive. Further study on the amount and type of personnel required for the development of comparable gaming environments could determine whether the team used for the IALTA development is representative.

An intelligent gaming environment
Other fields of application for the framework that is presented in this thesis could provide insight in the previous statement. Our modelling approach is not restricted to the modelling of military reality. Decision making, controlling and executing activities occur in every organisation. The organisation's behaviour is also influenced by its environment. Therefore, we suggest that our modelling approach can be applied to a more general class of modelling problems that addresses complex sub-systems of reality which entail complex decision making processes at different levels. Gaming environments which support training and analysis can be developed for these domains. Further research that applies the framework presented in our dissertation could test this hypothesis.
References


Bartlett G.T. LTGen (Ret.), Preparing the Army's Surrogate Battlefields, Signal, July 1989.


An intelligent gaming environment


IRSD, Les phases essentielles de la Def Aé, Septembre 86.

IRSD, Les forces aériennes tactiques, Juin 88.

IRSD, Introduction à la logistique operationelle et signes conventionels, Janvier 84.

IRSD, La logistique à l’échelon unité, Septembre 83.
IRSD, Le ravitaillement dans la brigade et la compagnie ravitaillement et transport, Janvier 82.

IRSD, Le service médical dans la brigade et la compagnie médicale de la brigade, Février 84.

IRSD, Le déploiement des moyens logistiques de la brigade et les règles d’emploi de ces moyens dans les différentes situations tactiques, Mars 82.

IRSD, Organisation des unités logistiques du Corps, Novembre 83.

IRSD, La technique d’estimation des besoins dans la Bde, Octobre 84.

IRSD, La transmission et les troupes de transmission, Septembre 88.

IRSD, Les transmissions à l’échelon brigade, Septembre 88.

IRSD, Fractionnement, Septembre 86.

IRSD, Les mouvements routiers, Avril 82.

IRSD, L’appui feu, Mai 88.

IRSD, Emploi et mise en oeuvre du génie, Septembre 87.

IRSD, L’appui génie, Septembre 86.

IRSD, Terrain analysis, Unit Movement, IALTA project, Brussels, 1990.

IRSD, Delaying Manoeuvre work statement, IALTA project, Brussels, 1990.

IRSD, Object environment, IALTA project, Brussels, 1989.


IRSD, ILTA simulation interaction specification, IALTA project, Brussels, 1990.
IRSD, IALTA Bn functions, IALTA project, Brussels, 1990.
IRSD, IALTA simulation environment order structure, IALTA project, Brussels, 1990.
IRSD, WING KB work statement, IALTA project, Brussels, 1990.
IRSD, SAM KB work statement, IALTA project, Brussels, 1990.
IRSD, SOJ KB work statement, IALTA project, Brussels, 1990.
IRSD, Modules IATA existants, IALTA project, Brussels, 1989.

Joint Directors of Laboratories, Assessment of C³I in Military Games and Simulations, Technology Panel for C³, C³ Research and Technology Program, National Defense University, March 1987.

KHID, Dossier Luchtverdediging, Oktober 87.

KHID, Dossier Offensive Ops, Oktober 87.

KHID, AI Ops, September 88.

KHID, De Offensieve middelen, September 88.

KHID, OCA Ops, September 88.

KHID, Offensieve Luchtsteunoperaties, September 88.

KHID, De operationele logistiek op de echelon eenheid, Juni 89.

KHID, Operationele logistiek, September 87.

KHID, De logistieke steun op de echelon eenheid.

KHID, De logistiek op de echelon eenheid, September 83.

KHID, De ravitaillering in de brigade en de Cie Rav/Tpt/Bde, Januari 82.

An intelligent gaming environment
KHID, De maintenance in de brigade en de Cie Mat/Bde, Januari 82.

KHID, De medische dienst in de brigade en de medische compagnie van de brigade, Maart 82.

KHID, Organisatie van de logistieke eenheden van het Corps, November 83.

KHID, Techniek van het schatten der behoeften in de Bde, October 84.

KHID, De taktische inlichting, April 85.

KHID, De taktische inlichting, Juli 87.

KHID, Conventionele vijand, September 85.

KHID, De beoordeling door de stafchef, September 87.

KHID, Aanpassing van het stafwerk in offensieve operaties, September 85.

KHID, De vuursteun, Mei 88.

KHID, Organisatie en uitrusting van de genie, September 87.

KHID, Gebruik en inwerkingstelling van de genie, September 87.


An intelligent gaming environment


Trotha von T., Zum Kriegsspiel, in Jahrbücher für die Deutsche Armee und Marine, Band 10, 1874.


An intelligent gaming environment
List of abbreviations

Aie : Artillery
ALO : Air Liaison Officer
ASOC : Air Support Operations Centre
ATAF : Allied Tactical Air Force
ATk : Anti-Tank
ATM : Air Task Message
ATOC : Air Tactical Operations Centre
Bde : Brigade
Bn : Battalion
CAP : Combat Air Patrol
C² : Command and Control
CCIS : Command and Control Information System
CFE : Conventional Forces Europe
C³I : Command, Control, Communications and Intelligence
Cie : Company
DBMS : Data Base Management System
Div : Division
DOO : Daily Operations Order

An intelligent gaming environment
Eni : Enemy
Gn : Engineering
IATA : Intelligent Air Training Application
IALTA : Intelligent Air Land Training Application
ILTA : Intelligent Land Training Application
INF : Intermediate Nuclear Forces
IS : Information System
LAN : Local Area Network
Log : Logistic
Lt Avn : Light Aviation (Helicopter)
Mat : Material
Med : Medical
MMR : Model of Military Reality
NAEW : NATO Airborne Early Warning
NATO : North Atlantic Treaty Organisation
NORTHAG : Northern Army Group
OCA : Offensive Counter Air
OAS : Offensive Air Support
PC : Personal Computer

An intelligent gaming environment
Ptn : Platoon
Rav Tpt : Supply and Transport
RBDC : Royal Belgian Defence College
Recce : Reconnaissance
RLS : Real Life System
RMA : Royal Military Academy
RS : Real System
SAM : Surface to Air Missile
SOC : Sector Operations Centre
TTr : Transmission Troops

An intelligent gaming environment
SAMENVATTING

War-gaming kan voor vijf verschillende toepassingsgebieden gebruikt worden, namelijk, training van procedures en van beslissingsprocessen, analyse van operationele plannen, analyse van de doeltreffendheid van wapen-systemen, analyse van combinaties van wapen systemen en analyse van wapen-echelon combinaties. Tot op heden is meestal een aanpak gevolgd waarbij per toepassingsgebied specifieke modellen voor bepaalde vormen van interactie en resolutie ontwikkeld worden.

Het doel van ons onderzoek is een manier van denken, modelleren, werken en beheren aan te geven om modellen te ontwerpen die bij een gegeven resolutie voor meerdere toepassingen bruikbaar zijn en meerdere vormen van interactie toelaten. Een goede correspondentie met de realiteit is dan ook een eerste doelstelling.

Huidige aanpakken voor het ontwikkelen van war-gaming modellen zijn onvoldoende om ons doel te bereiken. De meeste bestaande modellen vertonen namelijk een tekort aan experimentele validiteit omdat analytische modellen de menselijke beslissingsprocessen van lagere echelons niet weer geven. Daar tegenover staat dat modellen die deze beslissingsprocessen daadwerkelijk willen weer geven, een groot aantal deelnemers nodig hebben waardoor het herhalen van experimenten moeilijk en duur wordt. Verder treedt een leer-effect op waardoor de experimentele validiteit van herhaalde games nadelig beïnvloed wordt.

Uitgaand van het paradigma van beperkte rationaliteit en kijken naar de activiteiten binnen een war-gaming omgeving kan men onderscheid maken tussen het nemen van beslissingen, het ondersteunen van het beslissingsproces, het uitvoeren van beslissingen, het controleren van de uitvoering, de invloed van de omgeving op de uitvoering en het beheren van al deze acties. Op grond hiervan komen wij tot een object georiënteerd model waarin objecttypes volgens deze activiteiten gespecificeerd zijn. Wij veronderstellen dat hierdoor de correspondentie met de realiteit verbeterd wordt.

Teneinde het aantal deelnemers aan een war-game te verminderen en meerdere vormen van interactie met de war-gaming omgeving toe te laten, stellen wij voor beslissingsobjecten te ontwikkelen die menselijke beslissingsprocessen kunnen nabootsen. Hierbij wordt een structurerings- en communicatie principe gebruikt.

An intelligent gaming environment
Onafhankelijke beslissingsprocessen worden in aparte objecten gescheiden. De beslissingen van verschillende objecten worden gecoördineerd op grond van de precedentie regels en de iteratie drempels binnen het overkoepelende beslissingsproces.

Een geïntegreerd lucht-land model dat zich op de hoogste natieale en de overeenkomstige NATO echelons richt wordt met behulp van deze concepten ontwikkeld. Het model wordt gebruikt voor training en analytische doeleinden. Menselijke interactie kan op verschillende echelons plaatsvinden en het aantal deelnemers is beperkt tot de doelgroep van het spel. Het project heeft drie en een half jaar in beslag genomen. In deze tijd zijn 20 complexe beslissingsobjecten, een dynamische simulatie die uit een honderdtal objecten bestaat en de gebruikersomgeving voor spelers en spel-ontwerpers ontwikkeld. Een architectuur is toegepast waarbij alle apparatuur op een lokaal netwerk is aangesloten. Individuele spelers beschikken over een personal computer waarop de programmatuur is geïnstalleerd die alle persoonlijke activiteiten ondersteunt. De gemeenschappelijke gegevens van staven zijn op specifieke werkstations toegankelijk. Het dynamische model wordt op een cluster van mini-computers uitgevoerd en past de verspreide gegevensbanken aan.

Uit de experimenten met het resulterende JALTA (Intelligent Air Land Training Application) systeem blijkt dat de ontwikkelde beslissingsautomaten in staat zijn spelers te vervangen. Hiermee is de gewenste flexibiliteit van de gaming omgeving bereikt en kunnen meerdere toepassingen ondersteund worden. Wat betreft de correspondentie met de realiteit wordt vastgesteld dat over de aparte objecten tevredenheid bestaat, maar over het gehele gedrag moeilijk een uitspraak valt te doen. Dit is waarschijnlijk te wijten aan het feit dat militaire experts geen werkelijke directe ervaring meer hebben met grootscheepse conflicten.

Wij concluderen dat de voorgestelde aanpak de nodige concepten naar voren brengt om war-gaming omgevingen te ontwikkelen waarmee op verschillende echelons de gewenste interactievorm gerealiseerd kan worden. De toepassing van ons raamwerk is niet beperkt tot het militaire domein en een beter inzicht in het vraagstuk van de correspondentie met de realiteit zou wellicht bereikt kunnen worden door de ontwikkeling van een omgeving voor gaming op commercieel gebied.