Game Environment for Command and Control Operations (GECCO)
http://www.nada.kth.se/theory/gecco/

Joel Brynielsson and Klas Wallenius
Department of Numerical Analysis and Computer Science
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
{joel, klasw}@nada.kth.se

Abstract. In GECCO we have implemented a game platform specifically for the Command and Control research community. GECCO is a strategy game where one move units on a map. The game is generic in the sense that it stores all information regarding the scenario in files that are easy to adjust. The source code is distributed to the research community as open source, and it is well documented and well structured. The game is suited to all common computer environments.

1 Introduction

In the world of today there are many emergency organizations that rely on operation management from their staff. The situations that occur are different from time to time and are often very complex. Another type of organization that relies on operation management from their staff is the military, where Command and Control centers need to evaluate the current situation in order to give correct orders based on available information.

Although commonly used in the literature, there is no agreement on what “Command and Control” really means[27]. The definition given by Coakley[10] seems to be the most widely accepted one and is also the definition we adopt:

Command and Control is everything an executive uses in making decisions and seeing that they are carried out; it includes the authority accruing from his or her appointment to a position and involves people, information, procedures, equipment, and the executive’s own mind.

Decision making in emergency and military organizations is classified as distributed decision making[4], which means that the decision making is distributed among the actors in the organization. These ideas influence the ongoing research in Command and Control[1, 12, 19, 26, 28, 34], and also seem to be the objective of products currently under development[32, 33].

Computer simulations of real world phenomena are gaining importance in the Command and Control field, as well as in other sciences. As computer power increases, more detailed real world simulations can be accomplished. In this
paper we will study schematic simulated worlds called microworlds[3]. In such microworlds we study solely the questions of interest, and disregard other properties.

Many examples of realistic microworlds can be seen in the war games available for personal computers on the commercial market. The games available today are, however, not suited for research purposes[20, 25], mainly because the game play history cannot be registered for later analysis, and because the researcher does not have access to the source code in order to make changes “on the fly”. Moreover, researchers tend to develop a new microworld each time they want to investigate a new topic. In this work we have chosen to address these problems by creating an environment that serves as a generic platform for future game development.

2 Designing a Generic Game Environment

It is difficult to design artifacts general enough to be usable for many different purposes. A “game”, for instance, could be a lot of things. In a dictionary[22] several meanings are given to the word, of which the following seems to best fit what we mean:

\textbf{game} – a physical or mental competition conducted according to rules with the participants\(^1\) in direct opposition to each other

It is not, however, possible to design a game environment for all possible games according to such a broad definition. Hence, we have had to place restrictions on what kind of games we have in mind. The aim has thus been to design a generic computer based environment for such competitive activities that comply with the following criteria:

1. The game’s focus is on Command and Control operations, i.e., the players are in charge of various units or resources which can be utilized to fulfill the goals of certain operations.
2. The problem space is constituted by the ability of the units to act and observe. They are thus able to move in a geographical environment, and to interact with each other and with the environment, where interaction includes both observing and affecting each other’s states.
3. The ability to act and observe is determined by rules and models represented in the game environment.

We claim that GECCO can be customized to serve the large class of games that is defined only by these restrictions. In section 4 we will give some examples of currently developed game scenarios indicating that this is actually the case. The achievement of this generality has been possible by the use of object oriented design. Three main components have been implemented, where Automata represent the geographical environment, Units represent the different kinds of

\(^1\) In this work we will consider a participant being similar to a collaborating team.
resources that can be commanded and *actions* represent the interaction models, all with characteristics that can be specified for different games.

It should, however, be kept in mind that GECCO is a platform; designing a game is an art, not a science[25]. The nature of game design requires a unique blending of talents, as well as the use of developed game design principles such as Dunnigan’s ten steps[11].

## 3 Game Characteristics

### 3.1 General Properties

GECCO is a strategy game where one moves units on a map. It is implemented using a client/server model where the computers communicate via a computer network (see Figure 1). The server holds the simulation and acts as the game engine. A client communicates with the server to obtain information about what to present. A client is only allowed to maneuver units that it has at its disposal.

![Client/Server Model](image)

_Fig. 1._ GECCO uses a client/server model. A server holds the simulation and is the game engine. A client communicates solely with the server that tells the client what should be presented.

A GECCO session can take any number of clients, depending on the scenario and the requirements. A client may be a human that maneuvers his units using a graphical user interface or a client may be automated by a computer.

As of today, we have implemented a non-automatic client that gives a human player the possibility to command units. This client holds a graphical user interface displaying all units that the player controls along with units that he knows of. This client also provides insight in what areas one sees at the moment.
3.2 Geographical Environment

The geographical environment in GECCO consists of an automaton matrix. The automaton matrix is automatically constructed from an image which is read by the server. Each pixel in the image is classified as a predefined automaton\[24\] (i.e., forest, building, road, water, etc.). Different automata may have different properties; for example, forest automata might start to burn, and water automata may get polluted.

3.3 Units

Units are vector based to enable us to calculate what automata the unit affects. A unit holds a set of properties, specific to the unit type, that describe the state of the unit. Such properties may be amount of fuel, ammunition, water, food, and health status.

A client may have the right to observe or command a unit. If a client observes a unit, the client receives all information that the unit has in its possession. If a client commands a unit, the client is also able to send actions (see section 3.4) to the unit. Typically a player has command rights on his own units, observe rights on his friends units, and no rights at all on units that belong to the enemy.

3.4 Actions

Actions are used for interaction between automata and units. When a unit or an automaton wants to do something, like putting another automaton on fire, attacking another unit, or distinguishing fire, it sends an action to the unit or automaton that it wants to affect. Specific actions are specified for specific automata and specific units. An automaton may affect another automaton, a unit may affect another unit, a unit may affect an automaton, and an automaton may affect a unit.

3.5 Open Source Project

We provide GECCO as open source to the research community. It is distributed using Gnu Public License (GPL)[13], meaning all code is free for everyone to use. Today, version 1.0 has been released together with well documented source code, manuals[5, 6, 7, 8] and executable files.

4 Example Scenarios

As a start we have developed four example scenarios. The scenarios have been chosen for two reasons:

1. they are different from each other in the sense that they will make use of the GECCO architecture in different ways and therefore evaluate the implementation from many different perspectives,
2. they are, to some extent, implementations of scenarios used in current Command and Control research at the Swedish National Defence College[35].
4.1 World War II

The date is June 19, 1944. The outcome of World War II is still uncertain. The allied forces have established a bridgehead on the Cotentin peninsula in France in order to conquer Cherbourg, and the rest of Cotentin.\footnote{Today we know that the allied forces succeeded in seizing power of Cherbourg as of June 26, 1944.}

This scenario (see Figure 2) represents a large scale operation in Europe at the operational level. The map represents a large area. The time perspective concerns days rather than hours. We think that this scenario would be suitable for training of military commanders in Command and Control environments such as the one offered in ROLF\cite{26, 28, 29}.

Fig. 2. World War II scenario. In the picture we see that the user plays the role of Maj. Gen. J. Lawton Collins, commanding the U.S. VII Corps.

4.2 The Öresund Bridge Collapses

Among the twelve scenarios defined in \cite{35}, this is perhaps the most heavily discussed one, see for example \cite{2}. A tanker hits the Öresund bridge, trains
and cars fall into the freezing water, oil is leaking from the tanker threatening to pollute the coast of southern Sweden, oil is burning, etc. To strengthen the Swedish society in case of emergency is one of four main tasks for the Swedish Armed Forces [30, 31].

This scenario (see Figure 3) requires helicopters and boats to solve various tasks in the disaster area.

Fig. 3. Rescue mission. This picture shows a game where the Öresund bridge, connecting Denmark and Sweden, has collapsed. One are to search for, and rescue, people that are floating around using helicopters and boats. The picture shows the team leader’s view, namely all units and the contents within their sensor ranges.

4.3 Fire Fighting

Fire fighting is a well studied topic at the Swedish National Defence College where fire simulations are run on a regular basis. Players interact with the simulations via a decision support system specifically designed for forest fire fighting called C³FIRE [17, 18].

The fire fighting game developed for GECCO is based on the same ideas as C³FIRE and makes use of the same algorithms as C³FIRE when simulating the fire development. A number of new unit types have been added, such as a “bandit” car that ignites fires, and fire extinguishing helicopters.
4.4 Tactical War Game

When developing GECCO we needed a game on which to test our ideas. Therefore we developed a game with a scenario that looks like a simple version of the war games available commercial on the shelf. The scenario uses a map, covering a small part of Sweden. There are three types of units in the game, namely tanks, helicopters and radar stations, each type with a particular set of properties (i.e., the helicopters can travel across lakes, the tanks can take a lot of damage, the radar stations have a broad sensor range, etc.).

5 An Example Using GECCO for Command and Control Research

We have used the tactical scenario described in section 4.4 to create an example of how to use GECCO for Command and Control research. Imagine that Bob has developed a model for decision making in staffs. He now wants to see how his model works and therefore he wants two staffs to play a game where they are each other’s opponents. One staff will be using Bob’s model while the other one will not. Bob wants to play the game on two different Command levels and therefore defines the following roles:

The Staff has got an overview of all its units, but is not able to operate them. Each Company Commander operates units that belong to his Company, but he is only able to see what is within the field of vision of his own units.

Besides this, Bob wants to keep track of the situation himself while playing, and therefore he creates a God’s eye that covers everything. In Figures 4–6 we show the possible setup.

![Fig. 4. Company commander.](image1)

![Fig. 5. Staff.](image2)
6 Future Plans

6.1 Research

The primary purpose of GECCO is to supply the research community with a useful research environment. Primarily we think of research within the following areas:

- development of tools for decision support that can be connected to manual clients in the game, see for example [9],
- development of computer generated forces, CGF:s, i.e., “smart” clients that are controlled by the computer. Many interesting and stimulating techniques are available for this task[23], and several interesting projects exist[14, 36],
- development of intelligent units/agents that learn as they travel[15, 16],
- microworld studies, see for example [21],
- development of tools for statistical treatment of log files.

6.2 Command and Control Training

A common problem for military, as well as civilian, Command and Control centers is that they need to practice. In most of the situations (e.g., a forest fire or a war) it is infeasible to let the staff practice in a real world scenario. Although such training will still be required, some training objectives can be met by practicing in microworlds that simulate reality.

Within the Swedish National Defence College, Command and Control microworld simulations are run within the ROLF project[26, 28, 29]. We think that GECCO will be of value for this research.
7 Conclusions

In GECCO we provide the research community with a tool that has not been available before. With the help of GECCO both behavioral and technical questions in the Command and Control field can be answered. The main property that makes GECCO the main tool for Command and Control research is that it is able to represent a large genre of games. We describe this property by saying that GECCO is scenario generic. Besides being scenario generic, GECCO is:

- made especially for applied research purposes,
- open for everyone to use and modify,
- well documented,
- suited to all common computer environments.

We think that GECCO will be used and developed further by Command and Control researchers around the world.

Acknowledgments

We would like to gratefully acknowledge the help of our students, Henrik Bäärnhielm, Andreas Enblom, Jing Fu Zi, Niklas Hallenfur, Karl Hasselström, Henrik Hägerström, Oskar Linde, and Jon Åslund, for their invaluable help in designing and implementing GECCO.

References


