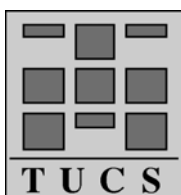


A CONTINUOUSLY PROCESSED BUSINESS GAME CONSTRUCTION FOR BUSINESS PROCESS TRAINING

Timo Lainema

Turku School of Economics and Business Administration,
Department of Information Systems Science,
Lemminkäisenkatu 14, FIN-20520 Turku, Finland



Turku Centre for Computer Science
TUCS Technical Report No 499
April 2003
ISBN 952-12-1101-6
ISSN 1239-1891

Abstract

This report represents some features of a computer-based learning tool (REALGAME) constructed as a licentiate thesis for the Turku Centre for Computer Science (TUCS) and Turku School of Economics and Business Administration (TSEBA) during 1997-2000. In this document we will concentrate on describing (a) the game development environment and game structure, (b) the business processes in the game, and (c) the configurability of the game. Game processes will be represented according to four principal types of economic utility, which add value to a product. The main result of this study is that continuous processing and the learning tool business environment configuration are technically feasible possibilities, although hardly at all applied in practice before this construction.

Keywords: Business games, Real-time processing, Continuous gaming, Business process training, Configurable learning environments.

1. Introduction

This research report is – as the name of the publication series suggests – mainly a technical representation of a business decision-making learning environment. The argumentation supporting the construction of the learning environment has been documented elsewhere (e.g. Lainema, 1999a, 1999b; 2000, 2001, 2003b; Lainema and Makkonen, 2003; Lainema and Nurmi, 2003) and we are not going to go through this quite well documented discussion any more. However, it is worth mentioning the research hypothesis underlying this report (based on the research hypothesis in the author's licentiate thesis; Lainema, 2000):

An interactive (real-time/continuously processed) business game describes the interactive nature of the present business environment and sudden shifts in it, and how processes unfold in a company's internal and external environment. This verisimilitude (real-world likeness) of the processing method of the learning tool creates an authentic and meaningful environment enhancing participant business perception through game playing.

In the new business game construction real-time/continuous processing has been the major guiding principle. This continuous nature of the learning tool is a relatively novel concept in the field of business gaming (see Lainema and Makkonen, 2003, for more information). Another construction principle represented in this report is the configurability of the learning tool. This is a property that is seldom effectively implemented in business games. The argumentation for the configurability of a learning environment and learning outcomes from it are presented in Lainema and Nurmi (2003).

What we are trying to achieve through the constructive research reported here is a holistic business game construction describing the main processes of a manufacturing company. Manufacturing in this context is not a central function in itself; we are only drawing a distinction between manufacturing organizations and service-oriented organizations.

As the changes in the business environment are extensive, it must mean some new demands for business training, too. Obviously the demand to understand the overall business process structure is becoming increasingly important. The workers in any business face changes in the world around them at a growing speed. This means also that the employees must be able to change their way of thinking and working more often than before. The general knowledge of how a company works helps the worker to adopt new behavior. Again, this is argued more deeply in the author's forthcoming doctoral dissertation and no more space will be used to argue this.

Lane (1992) states that it is hard for people to imagine the consequences of an action after it has worked through a very complex system, but this process of understanding can be helped with the use of models:

As humans create a world of increasingly complex systems, it becomes more important that we be able to formulate views on their operation which are coherent (and holistic). A tool which allows a client to grapple with some of his/her ideas can lead, for example, to the discovery that goals which seemed reasonable when only

part of the system was viewed are seen as inconsistent or impossible in the context of the whole system. Another benefit comes when a model reveals how a high-level result, opinion or consequence has been built up by small components. It is particularly interesting and useful to take two opposing positions and look at their component assumptions to isolate the exact point of disagreement. This is an excellent process for beginning to resolve such conflicts of opinion.

In general, what Lane describes above is something that the construction of this document is aiming at also. However, in this document the main emphasis is on the description of the following:

- (a) The game development environment and game structure,
- (b) The business processes in the game, and
- (c) Configurability of the game.

When describing the business processes of the construction we apply the four **principal types of economic utility**, which add value to a product (or service). The classification we are using is introduced in Coyle et al. (1996). The principal types of economic utility are: **form, time, place, and possession**. Generally, manufacturing activities provide form utility, logistics activities time and place utility, and marketing activities possession utility.

Before proceeding, we will introduce the context of use of the gaming environment. The essential idea of the game is to position the participants in a role where they have to manage a manufacturing business. This should be done in a profitable manner to keep the company alive. The companies are in continuous information exchange with their customers, suppliers, and indirectly with their competitors. The game company represents a total enterprise model of a manufacturing company in which decisions from one functional area interact with those made in other areas of the company. The game can be both industry specific and generic, depending on whether the game model is configured/tailored for the case organization.

The intended audience of the game varies. The game is suitable for participants from middle management position to foremen of production, and to business students. As the game clock speed and game complexity are variable, these parameters can be adjusted according to the audience. The participants adopt roles of decision-makers. They have to make decisions on production, supplies, sales, marketing, investments, transport, and so on. This report includes screen copies which further clarify the managerial role of the participants. The goals of the game can be varied, depending on the aims of the training. These may include efficient production, profitability, optimizing inventories, maintaining economical self-sufficiency, and so on.

As the game is complex, running a game training requires at least eight hours. Preferably the game session should last one and a half or two days to obtain all the benefits of the game. The required time depends of course on the selected complexity of the game model and the speed of the game clock.

The participants, after having learned the rules of the game, are mostly free to manage their companies. Thus, the facilitator intervenes only if the participants wish to have additional guidance.

2. WHAT IS CONTINUOUS SIMULATION?

We argue that for the present business environment to be described more realistically, the influence and importance of time should be embedded in the business game to reflect the cause-relationships in the business environment. Besides of this, the new business game construction (REALGAME) described in this paper includes a holistic view of business (all major business functions and stakeholders), and represents this for the game participants. Furthermore, the construction includes the ability to configure the business game according to different business environments.

The time argument means that the connection between players, supply market, customers and capital market needs to be interactively processed. The role of time in simulating time-bound business processes and decision-making is essential, as well as the communication between the companies and different stakeholders. What is suggested here is a real-time or continuously processed business game. Decision-making, and results from the decisions made, should occur in an interactive on-line mode, as they do in the real-world environment. In a continuously processed business game different business events and processes take place continuously and concurrently, and often at a varying frequency. The participants steering the company view all the market events and internal processes on-line. Whatever happens can be seen without a delay and action can take place instantly - providing the participants realize the need to do so. Thus, the game emulates the real world processes of business environments with the major exception that the internal simulation time is accelerated compared to the real world. In managing this kind of environment, the participants' ability to perceive processes and causal dependencies is essential.

The on-line processed game (REALGAME) was developed at the Turku Centre for Computer Science and Turku School of Economics and Business Administration (TSEBA) during 1997-2000. When writing this document, the new game construction has been used some 30 times on different teaching occasions, in company in-house training session, on MBA courses and on courses for undergraduate university students.

In REALGAME there are 6-8 companies competing against each other, with the market suppliers and funding organizations being common to all the companies (teams of participants). This game has been programmed with a Rapid Application Development (RAD) tool (Delphi) in Windows environment. Delphi enables the use of databases and supports user-defined inherited objects. Both of these resources are seen as essential for the creation of REALGAME. Databases are needed in order to record all the detailed business transactions taking place during the game run. The transaction specific representation mode of REALGAME is characteristic of it and this specificity can be seen in all of the processes of the game. Like the use of databases, the object-oriented development environment has been a major requirement for the work to be successful. Without the object oriented development environment it would have been considerably more difficult to construct a truly configurable business game. This document will later give some examples of the game configurability possibilities.

According to the terms introduced by Thavikulwat (1996), REALGAME is scaled flexibly (participants can select when to print reports), synchronized (all game participants are bound to the same period lengths), and clock-driven (the game time advances in concert with the computer's internal clock). Furthermore, REALGAME is

interactive in both the respects Chiesl (1990) mentions (first, continuous interaction between the game and the players; secondly, players interact with other participants). In continuous game processing the functions are executed in a continuous and iterative manner. Different steps can be enacted simultaneously or separately. Different iterative decision loops may occur at the same pace or at a different pace. It may be impossible for the decision-maker to decide beforehand what actions to take at particular moments. The process is never ending, with any easing off in decision-making resulting almost certainly in trouble. Furthermore, the participants are part of the game processes. This means that they see the changes in their environment evolve on-line. We feel that this is a major difference if we compare continuous processing to batch-processing: the participants are an organic part of the business processes and are able to witness and see them, and – most of all - are able to interact continuously with this process. As Huckfeldt et al. (1982) have noted, the models employed should not be judged on the extent to which they replicate a social process, but rather on the extent to which they help us understand a social process. Continuous processing makes the learning environment business processes transparent and according to constructivist learning principles (see e.g. Lainema, 2003a) this is useful from the learning point of view.

The impact of time in real-time and batch-processed business game decision-making in some business operations and processes is described in table 1. The examples in the table illustrate just some of the differences between batch-processed games and real-time processed games.

Operation / Process	Batch-processing	Real-time processing
Reacting to opportunities or threats in the market	The speed at which the participants react to opportunities does not have a realistic significance, because all the companies deliver their decisions (or at least their decisions are processed) at the same moment. The fastest decision-maker does not benefit from fast decision-making. E.g. no company has any advantage of adapting early to customer needs.	The true nature of decision speed is represented, because a faster decision maker always responds first to any external events. For example, the company that fastest adapts to changing customer needs can gain new market share.
Response to misleading production plans	A misleading production plan cannot be straightened until the decisions for the next season are being made.	A misleading production plan can be straightened as soon as it is discovered.
The advantage of faster product development	In some cases the speed of the product development process does not have any significance. E.g. consider two companies (A and B) developing similar novel products. Company A develops the new product during the season in half of the duration of the season. Company B develops the new product during the season but it takes the whole length of the season. In this case company A has minimal - or no - advantage of being faster in the development process.	The company being faster in development processes earns all the benefit it deserves from being faster. E.g. it can benefit from being the only provider of the novel product by demanding higher prices.

Table 1: Some examples of differences on how the participant decision-making may be treated differently between real-time and batch-processed business simulation games.

Actually, REALGAME does not operate in true real-time processing. True real-time processing would mean that each and every independent game object (e.g., a customer or a supplier) would have its own internal Timer object, which would activate independently of everything else in the game, and process all object specific tasks and processes. Very early in the development of the game, this kind of true real-time processing was tested but proved to be impossible to use. This was due to the Windows operating system, which allows only a limited number of active Timer objects to be used at a time. True real-time would have needed hundreds or even thousands of game internal Timer objects and this is not at all realizable in a Windows environment, as the operating system's resources run out with no more than some tens of timer objects. However, the processing is continuous in the sense that:

- (a) the game time is clock-driven - the smallest increment of time being one hour,
- (b) the participants are not tied to making decisions at specified points of time but can make them whenever they choose,
- (c) the decisions made at each point of time can be single decisions or several decisions but no decision batches are required,
- (d) the participants may choose to run reports at any point of time, and
- (e) the participants see the internal and external business processes evolving, e.g. hour by hour, depending on the game parameters (explained below).

The clock speed in REALGAME varies in three phases. The first phase is usually the slowest one and the third phase the fastest, although the speed is not fixed in any of the phases. In the beginning of the game the duration of one REALGAME hour might be, for example, 30 real world seconds. In the second phase, the game might proceed with the speed of 15 real world seconds equaling one game hour, and in the third phase one real world second might equal one REALGAME hour. These different processing speeds are parameters of the game, and may vary between different game sessions.

3. REALGAME CONSTRUCTION

3.1 Construction Development Environment

The rest of this document describes the new business game construction programmed as the main contribution of the author's licentiate thesis (Lainema, 2000). The documentation of the game construction in this report is mainly descriptive. The technical documentation is not included in this report. The main purpose of this descriptive approach is to introduce the nature of real-time/continuous processing to the reader.

The game applications were developed in the Windows environment. Selecting Windows as the platform ensured that it would be possible to implement the game to the computers of any company, thus allowing in-house business game training in companies. In the beginning of the project the operating system in use was Windows 3.1 but very early in the project the 32 bit Windows environment had to be selected because of resource demands. In future it is possible to transfer the game also to the Linux environment with the Delphi Linux version (Kylix).

Both the company application and the market application of the game are Windows based applications programmed with Delphi. Delphi was selected as the researcher had previously carried out programming projects with Delphi. The programming task was started with Delphi version 1. After that, the Delphi versions used were upgraded to version 2 and 4. At the moment, as companies have upgraded their systems to Windows 2000 or XP, REALGAME has been upgraded to Delphi 6.

Afterwards it can be concluded that the selection of Delphi as the development tool was successful because:

- Delphi allows rapid application development (RAD). This means that Delphi's programming environment and the language itself provide properties and tools which significantly speed up the application development work compared to traditional third generation programming languages.
- Delphi includes easy-to-use embedded support to the use of databases. Databases are the only data storage method of the new business game construction.
- Delphi allows user defined objects. Furthermore, it was the only RAD tool at the time of the start of the project to allow the user to develop their own objects through inheritance from standard source objects.
- Delphi compiles true executable code (EXE code), which can be run from the Windows operating system without the need of a code interpreter.

REALGAME is not client/server based. The main reason for this is the light installation process of Delphi Standard in any computer class. Delphi C/S demands a much more complicated installation process than Delphi Standard. The ease of the installation process is an important aspect as the training is usually carried out on the premises of the organization whose employees are to be trained. This advantage from the light installation process has become very evident, as the game has now been installed in more than 10 different computer classes in Southern Finland. Normally the game

installation process takes less than one hour and no microcomputer expertise support is needed.

By using simple file copying routines the network usage of the game has proven to be relatively reliable: short or even longer network failures do not cause any errors in the game execution. Simple file copying makes it also possible to run the game applications in environments with fewer resources than would be needed with the use of client/server technology.

3.2 Game General Structure

REALGAME as an Information System

Any information system can be thought of as comprising an infrastructure and the systems which make use of that infrastructure (Land, 1992). In REALGAME the infrastructure consists of the following artifacts (following Land's classification):

- **The organizational structure:** game participants under the guidance of the game supervisor.
- **Communication channels:** computer applications using network data interchange.
- **Facilities:** TCP/IP computer network, market server computer, plus the technical network components.
- **Apparatus:** the workstation computers running each of the participating company decision-making applications.
- **Software tools:** both the company decision-making and the market server application, the database engine supporting both of these applications, and the Windows operating system.
- **Training, advisory and help facilities** provided to support the information systems activities of the user community: the game supervisor operating the market application, training/guiding the participants and conducting the briefing and de-briefing of the game session.

Land (1992) states that most information systems have three major information sources: (a) the real world itself, (b) the designed information system – an artifact, and (c) an informal information system. *REALGAME is not a typical information system in that sense that it does not directly include the real world itself as a major information source.* The designed information system may be configured according to a real world organization but during the game session the participants can access the real world only through the informal information system, that is, through the knowledge of the other participants about the real world. This situation refers to the factor of the range of channels used for conveying messages or signals to the user (one of four generic factors Land mentions).

As Land (1992) notes, in practice most information users operate as a part of one or more groups. These groups may work in a cooperative manner, sharing information in order to carry out tasks or solve problems. This is also the case with REALGAME. However, REALGAME is *not a group decision support system* in that sense that it does not have specific characteristics to support the work of a group. In other words, the game can be used by a single user as well as by a group of users, although the

educational situation is expected to be more fruitful for a group of decision-makers discussing different decision alternatives.

Land (1992) argues that *an information system is a social system*, which has information technology embedded in it. This can be also said about REALGAME. On the other hand it could also be argued that REALGAME is firstly information technology and then the social system has been gathered around it. However, the *training session is firstly and definitely a social system* and it must also be borne in mind that learning through gaming is partly or even strongly a social process. Land expresses that the designers of the formal part of a system have to be aware of the contribution made by the other half, and seek to provide links which enable the information user to make the most effective use of all components of the information system.

Land (1992) furthermore notes that *even the most rigidly defined systems will be used by its users in ways which were neither planned for nor anticipated by its designers*. In the case of REALGAME the more ways the users find to exploit the game system the better. The participants are encouraged to interpret and collect the extensive available data in ways that best serve their decision-making needs. In fact, this kind of process of creating one's own procedures to interpret game organization data to receive a holistic picture of the organization and its environment is highly recommended.

REALGAME Technical Architecture

The game works in a TCP/IP network environment (figure 1). A critical area for the cooperation of the game applications is the transfer of data between the market application (MAP) and the company decision-making applications (CAP). As CAPs are continuously in connection to the MAP (and visa versa) the local computer network has to operate reliably.

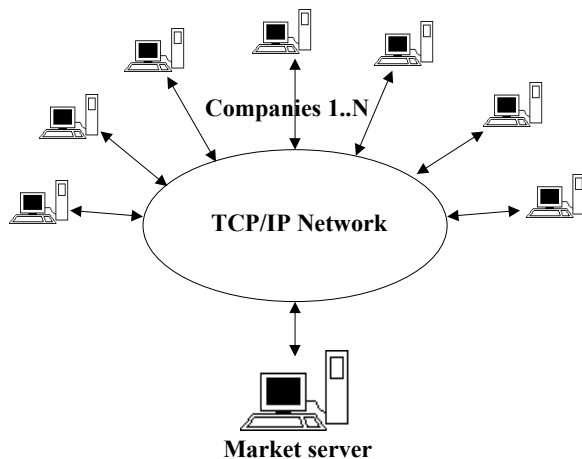


Figure 1: The game construction's general network structure.

The amount of companies in the game is not restricted by the game application itself. However, the maximum amount of simultaneously competing companies in the game event depends on the resources of the hardware environment and the Windows operating system in use. A standard Windows NT workstation allows a maximum of 10 workstations to be mapped simultaneously to it. Thus, using a Windows NT workstation

as a market server computer, there can be the maximum of 10 competing companies in one game. In practice we have allowed a maximum of 8 participating companies as Windows file sharing has proven to be somewhat unpredictable.

In the present version of the game the most restricting component for the game configuration is the speed of the MAP computer and its ability to handle the locking and releasing of the Market database alias subdirectories (figure 2, subdirectories 1...X...N). How well locking and releasing is handled depends on the Market server computer's processor clock speed and the type of Market server operating system. The network usage of the game construction requires at least a Windows NT operating system for the server, the clients can be Windows 97 workstations.

As client/server technology in data processing is not used, the different game applications can not access the same computer directories simultaneously. Thus, data processing has not been centralized but each application component of the game processes its own data locally. Local processing demands continuous data transfer between the MAP and CAPs. The data transfer from the market server computer to the company computers goes as follows:

- the market application processes the market events and transactions of one processing cycle and saves this data to the databases in the Market database alias (directory)
- the MAP locks each company specific Market alias subdirectory to be used exclusively, thus preventing CAPs from accessing the tables while it is accessing the subdirectories
- the MAP copies the databases needed in CAPs from the Market database alias directory to the company specific subdirectories
- the Market application releases the locks on the company specific Market alias subdirectories, thus allowing CAPs to access these subdirectories
- CAPs access the company specific Market alias subdirectories and copy the databases to their own Local database alias directories.

The data transfer from the companies to the market application goes as described above with the only exceptions that data is transferred to the opposite direction and the locking is done by the company application.

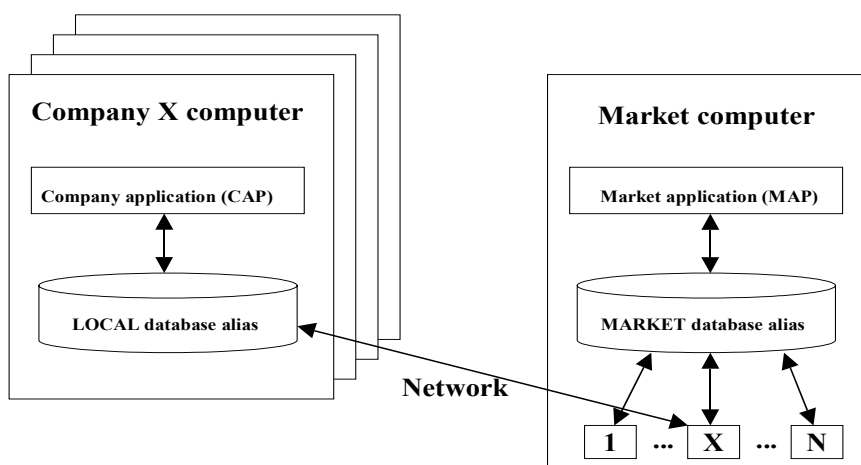


Figure 2: The game construction's general data transfer structure.

3.3 Description of the Game Processes

3.3.1 The Management of the Game Internal Time

The MAP controls the advancing of the game internal time. The MAP uses the time specific configuration the game administrator has defined in the CLOCK database table. The CLOCK table is also copied during each MAP processing cycle to CAPs to be used to guide the time handling in CAPs and to ensure that CAPs advance in the same phase as the MAP.

As described earlier the game does not operate in true real-time processing because the Windows environment allows only a limited amount of active Timer objects to be used at a time. The speed of the game time has three phases. The first phase is the slowest one and the third phase is the fastest. In the beginning of the game the duration of one game hour might be e.g. 30 seconds. This means that the MAP executes the events of one game hour every 30 seconds.

The motivation to use different game speed phases in the game arises from the need to train different kinds of business skills. The first (and slowest) game speed allows rehearsing short-term game company internal operations. These concern mainly operational decisions like raw material purchases, delivering, finding an optimal production capacity for production cells, and so on. As the speed is relatively slow the players have the possibility to see properly the material flows in the game but do not need to search frequently for external funding, invest on marketing operations, and so on. The first phase is also suitable for examining the dependencies between different materials transactions, accounts payable and receivable, and cash flow.

In the second phase the aim is to rehearse more tactical – mid-range - decision-making. Before turning on the second phase the players should have familiarized themselves with the production function. The second phase involves more decision-making concerning overall material chain management. In the first phase it might be difficult to get a holistic view of the material processes, as the game speed is too slow for that. This should be acquired in the second phase. Some first phase decisions are automated in the second phase in order not to block the decision-making capacity of the players. For example, the delivery process is automated in the second phase (an alternative for this would be a decision by the participants to invest in an information system producing automatic deliveries). As the second phase is usually considerably faster than the first phase the participants should be able to see a more holistic view of the dependencies between the functions taking part in the materials processes.

The third phase of the game is the long-term strategic decision-making level of the game. The speed of the game should be as fast as the MAP computer allows. At the best it has been possible to execute the game internal hours once in a real-world second but the MAP is not tuned up to be as fast as possible yet. Thus, in the future it will be possible to reduce the speed of the MAP third phase execution to fractions of a second on average. As the speed of the game increases during the game – depending on the computer capacity in use – the transaction processing of both the MAP and CAP will become too heavy for the computers to handle within the given cycle time. This is why the CAP can be set to process several hours during one cycle. For example, four game hours can be processed during one CAP processing loop.

To allow game speeds fast enough for game sessions where several game years can be simulated during one day, the speed of the MAP must also be accelerated in the same way (however this has not been done at the time of writing). For example, to be able to simulate three game years in one day training the execution of one hours must not take more than 0.833 seconds (the training time: 6 h x 60 min x 60 s, divided by the simulated game hours: 3 years x 12 months x 30 days x 24 h). The speed of the MAP to execute one game hour depends highly on the amount of participating companies and customers on the markets. Even with a relatively limited game configuration (some 8 competing companies and three customer markets, each including less than 20 customers) and a Pentium II/366 MHz computer MAP game hour processing takes some 1.5 seconds. Thus, to allow several years to be executed in a one-day training event the processing method of CAPs must be applied in the MAP also.

3.3.2 The Company Application (CAP)

The CAP includes several processes managed by the game participants. These processes are represented in the form of game computer screen copies in this chapter. Several of the company internal processes are also automatic and triggered by the game internal clock. The operation of the game clock is managed in the MAP but as the clock proceeds the time is transferred to each CAP, too. Thus, if the game internal time has proceeded, the CAPs will execute the company internal processes (like the production of goods within the manufacturing process).

The separate tasks in the CAP are all defined and saved as their own windows consisting of their own definitions and program code procedures. Altogether there are 32 independent window units plus one component (object) unit. The total amount of program code in these units is some 17.000 rows. This code includes only pure data processing procedures (plus, of course, comments within the code, and variable and type definitions) as each window's layout properties are saved in separate Delphi form files. These separate form files can contain up to 650 rows of property definitions of objects encapsulated in these windows.

The data of CAP transactions and operations is saved in database tables. First of all, this is because Delphi offers easy to use functions to manipulate data in databases. Secondly, relational databases offer a natural data structure to maintain transactional data. In the game construction the transactional databases are analogical to any enterprise software databases: the data is mainly structured according to the same normalization rules. Thirdly, databases provide database table indexes, which in many ways speed up the data processing and retrieval of the saved data compared to e.g. binary file routines. The database used in the game construction is Paradox database version 7.0, which comes with Delphi.

Figure 3 represents the REALGAME interface.

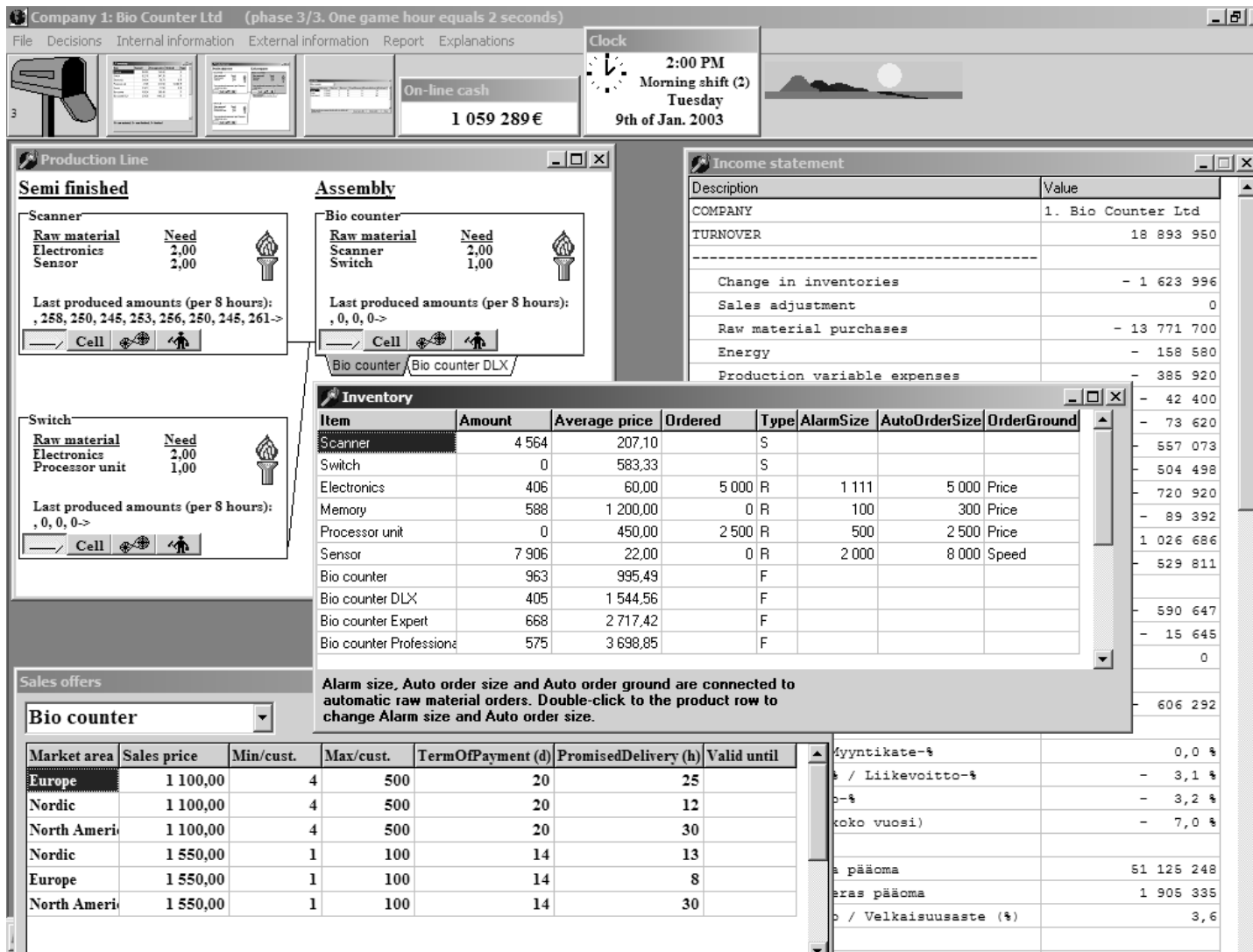


Figure 3: REALGAME interface.

The **Clock** window tells the player what time (game internal time) it is, what production shift is active, what day of the week it is, and what date it is. The game internal time proceeds with the same pace as the market program is proceeding. Thus, the company clock proceeds only when the market time proceeds first.

The **On-line cash** window tells the players the cash amount they have available in any moment. The cash amount is always in real-time and is updated after every cash payment and cash income. The menu items of the company application are shown in figure 4.

Company 3: Bio Counter Ltd [speed 3/3]				
File	Decisions	Internal information	External information	Report
Initialize	Suppliers (purchase) Continuous deliveries	Send public bulletin	Customers by market areas	Income statement
Exit	Production line Store (raw, semi, finished)	Cash flow Accounts payable and other depts Accounts receivable Forthcoming changes in capacity	Market area reports Quality report Market transactions from the last two weeks General Environmental Information	Send your income statement to the market Combined income statements of all companies (font 8) Combined income statements of all companies (font 7)
	Sales Marketing investments	Balance items Profit items		
	Order handling	All orders received Delivery terms in use		
	External funding	Sales reports Product reports		

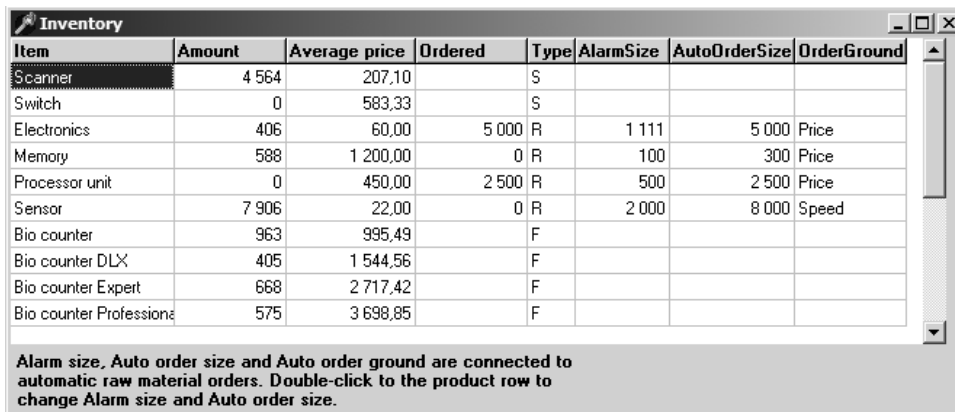
Figure 4: The menu items of the company application.

The User Operated Processes of CAP

When describing the user-operated processes of the CAP we apply the **four principal types of economic utility**, which add value to a product or service. The classification we are using is introduced in Coyle et al. (1996). The principal types of economic utility are: *form*, *time*, *place*, and *possession*. Generally, manufacturing activities provide form utility, logistics activities time and place utility, and marketing activities possession utility.

Form utility refers to the value added to goods through a manufacturing, production, or assembly process. E.g. this utility results when raw materials are combined to make finished products. This utility represents a change in product form that adds value to the product. Different material processing functions of the CAP together create the body of the CAP. Other business functions are more or less constructed upon the materials processing functions and support them. Following screen copies (figures 5-15) from the CAP represent the form utility processes in REALGAME. The following screen copies are from several different game sessions. Thus, the configuration (e.g. currency and interface language) and internal time vary between different screen copies.

During the game the production function uses the raw materials in the company inventory (*Store*) (figure 5). This inventory window includes finished goods, semi-finished goods and raw materials. As seen in figure 5 the company has run out of Electronics raw material.



Item	Amount	Average price	Ordered	Type	AlarmSize	AutoOrderSize	OrderGround
Scanner	4 564	207,10		S			
Switch	0	583,33		S			
Electronics	406	60,00	5 000	R	1 111	5 000	Price
Memory	588	1 200,00	0	R	100	300	Price
Processor unit	0	450,00	2 500	R	500	2 500	Price
Sensor	7 906	22,00	0	R	2 000	8 000	Speed
Bio counter	963	995,49		F			
Bio counter DLX	405	1 544,56		F			
Bio counter Expert	668	2 717,42		F			
Bio counter Professional	575	3 698,85		F			

Alarm size, Auto order size and Auto order ground are connected to automatic raw material orders. Double-click to the product row to change Alarm size and Auto order size.

Figure 5: The store window.

The materials process starts with *Raw materials purchases* (figure 6). In the third phase of the game the raw material purchases will be done automatically by the CAP. However, in the beginning of the game the participants themselves have to take care of purchases. The *Raw material purchases* window includes two tables. From the table on the left the player selects which raw material he/she wants to order. After selecting the raw material the table on the right will show the different suppliers of the selected raw material and their terms of delivery. Furthermore, each of these suppliers has a restricted inventory of each raw material. As the suppliers are common to all competing companies a certain raw material may become scarce. The raw material inventories of the suppliers are renewed in time but in some occasions the demand of raw materials may be greater than the renewal speed.

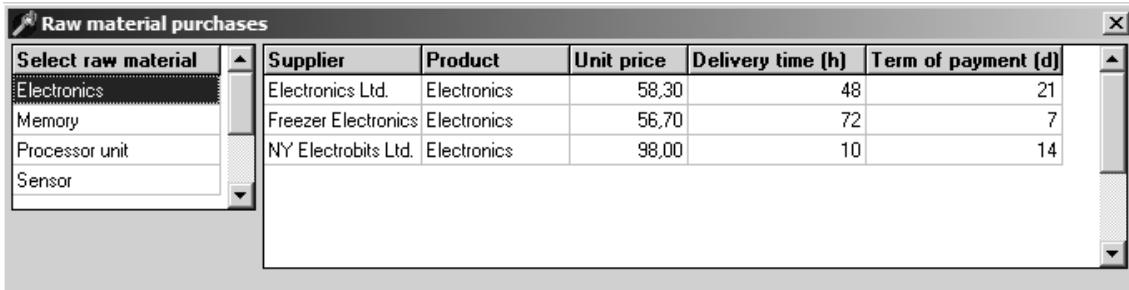


Figure 6: The raw material purchases window.

When the player has decided from whom to order he/she selects the *Buy* button. What follows is represented in figures 7 and 8.

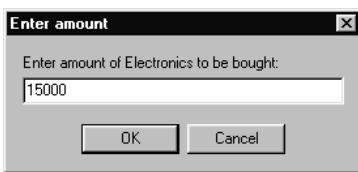


Figure 7: Entering the amount of raw materials to order



Figure 8: Confirming the order.

From this point forward only selected dialogs of the company application are presented as screen copies. Figures 7 and 8 show the principle in which the interaction between the game and the players is operated.

As the game describes time depended interactions within a manufacturing company most events and operations take certain time to materialize. E.g. the raw material ordered does not arrive at our inventory before the delivery time has passed. During this time the delivery can however be seen in the *Future capacity and material changes* window (figure 9).

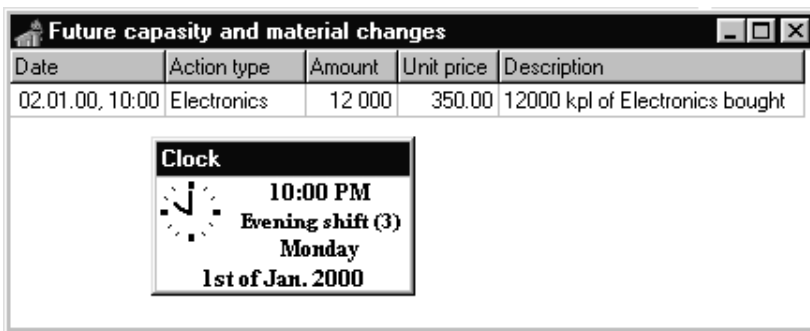


Figure 9: The future capacity and material changes window.

At the point of ordering the raw material we also become debtors to the supplier. The debt is entered to the *Accounts payable and other debts* table (figure 10). The debt will be paid automatically by the CAP by the due to date and time. From figure 10 can also be seen that the information included in the table is transaction specific information.

Date	Target	Action type	Explanation	Amount
08.01.00, 00:00	Nordic	Advertising	Advertising costs in Nordic	-8062
08.01.00, 00:00	Europe	Advertising	Advertising costs in Europe	-4017
08.01.00, 00:00	North America	Advertising	Advertising costs in North America	-8062
15.01.2000, 00:	Switch	Production salaries		-3780
08.01.2000, 00:	Switch	Production energy costs		-892.8
08.01.2000, 00:	Switch	Production var. cell costs		-7200
15.01.2000, 00:	Bio counter	Production salaries		-7560
08.01.2000, 00:	Bio counter	Production energy costs		-2352
08.01.2000, 00:	Bio counter	Production var. cell costs		-21600
15.01.00, 16:00	Bio counter	Delivery costs	Market area Nordic (1), customer 2	-54.5
15.01.00, 16:00	Bio counter	Delivery costs	Market area North America (3), customer 2	-1530.8
15.01.00, 16:00	Bio counter	Delivery costs	Market area North America (3), customer 2	-1522
15.01.00, 16:00	Bio counter	Delivery costs	Market area North America (3), customer 1	-1522
15.01.00, 22:00	Electronics	Raw material expenditures		-4200000
16.01.00, 05:00	Bio counter	Delivery costs	Market area North America (3), customer 1	-2538.5
16.01.00, 05:00	Bio counter	Delivery costs	Market area Nordic (1), customer 3	-56.3
16.01.00, 05:00	Bio counter	Delivery costs	Market area Europe (2), customer 3	-270.3
16.01.00, 05:00	Bio counter	Delivery costs	Market area North America (3), customer 3	-2538.5

Figure 10: The accounts payable and other debts window.

After the delivery time has passed the raw material appears in our inventory represented earlier (figure 5).

The raw materials are consumed in the company manufacturing function (figure 11). The production line of the company consists of a configurable amount of production phases (maximum amount of phases is five, two in the example) and cells (maximum nine per production phase, two in the first phase in the example). The flow of materials proceeds from the inventory to the first phase production cells, then the semi-finished products are transferred to the inventory, then the second phase cells take their materials from the inventory, and so on. The inventory always functions as a transfer container for the materials on their way through the production. Also the finished products end up in the inventory before they are delivered to the customers.

In the production line example of figure 11 the first phase of production consists of cells producing Scanners and Switches. In the second phase there is just one cell producing Bio counters, which is the final product in this model.

Production Line

Phase 1

SCANNER

Cell fixed cost/h: 0.00
 Cell var. prod. costs/h: 400
 Set up time (h): 2

Output/h: 15000
 Executed every 2. game hour
 Working shifts: 1, 2, 3

SWITCH

Raw material Need
 Electronics 2.00
 Processor unit 1.00
 Finished average price: 1978.14

Phase 2

BIO COUNTER

Raw material Need
 Scanner 2.00
 Switch 1.00
 Finished average price: 2716.81

Clock

9:00 AM
 Morning shift (2)
 Wednesday
 3rd of Jan. 2000

Store

Item	Amount	Average price
Bio counter	12	2 660.83
Electronics	11 328	350.00
Processor unit	864	1 249.63
Scanner	456	567.20
Sensor		32.90
Switch	864	1 683.50

Figure 11: The production window, the clock window and the store window.

Each of the production cells consists of four pages of cell information representing production cell specific information and capacity decisions. These pages are selected from the buttons (yellow arrow, Cell, wheels, and worker buttons) in the cells. From the production window the players can also turn the production cells on and off.

In figure 12 the player has selected the Workers button of Scanner cell. Now the player may either choose to hire new workers or sack old workers. In both cases there will be a delay before the workers quit or start working. Quite the same can also be done with production machines. In figure 13 the player has selected the Machines button of the Scanner cell. Again there will be a certain delay in getting the machines bought for production.

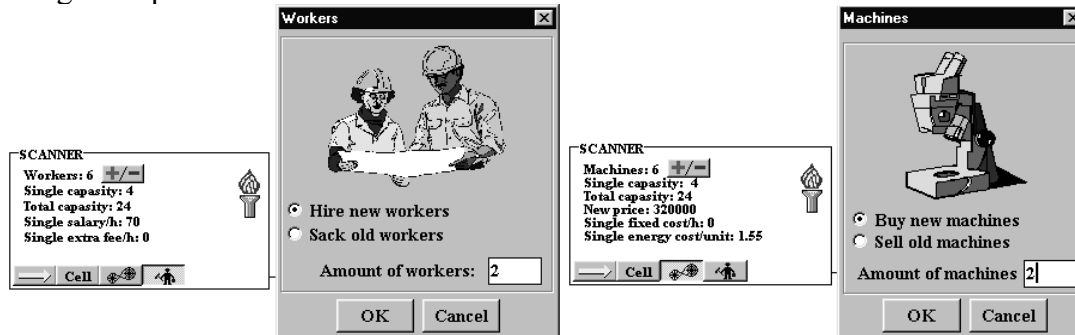


Figure 12: Hiring or sacking workers.

Figure 13: Buying and selling of machines.

Also the changes in worker and machine capacity and the delays these changes need to materialize can be seen from the *Future capacity and material changes* window (figure 14).

Date	Action type	Amount	Unit price	Description
22.01.00, 23:00	Electronics	1 000	350.00	1000 kpl of Electronics bought
23.01.2000, 07:	New worker	2	0.00	2 new worker(s) hired to cell Scanner
29.01.2000, 07:	New machine	2	320 000.00	2new machine(s) bought to cell Scanner
22.01.00, 23:00	Sensor	5 000	65.00	5000 kpl of Sensor bought

Figure 14: The future capacity and material changes window.

Last decision concerning production deals with the working shifts used in each cell. Figure 15 shows how the shifts can be turned on and off (the player has selected the Working shift button from the Scanner cell).

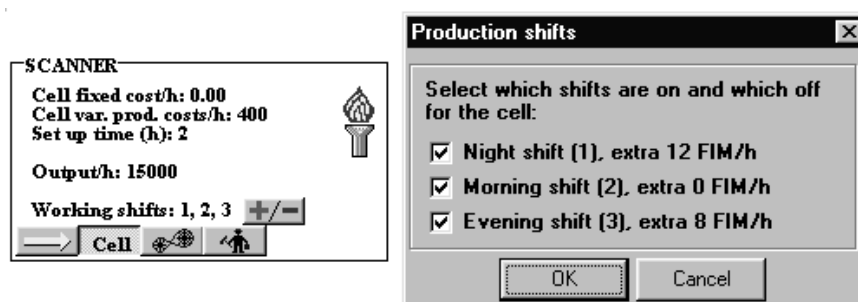


Figure 15: Selecting shifts in a production cell.

As the materials have flown through the production line the final products appear in the inventory and are ready to be sold/delivered.

We have now introduced most of the form utility functions of REALGAME. The next economic utility is **possession utility** (Coyle et al., 1996). Possession utility is primarily created through the basic marketing activities related to the promotion of products (or services). Coyle et al. define promotion as the effort - through direct and indirect contact with the customer - to increase the desire to possess a good (or to benefit from a service).

The customers in each market in any REALGAME configuration can be seen in the *Market area information* window (figure 16). From the selection box above the table the user can select any of the markets in the game. After the player selects an area the larger table shows the customers in the selected market. By viewing the customers the player can estimate the demand potential of the market. This information is calculated also in the smaller table at the bottom of the window. The maximum demand per month in the table tells the maximum demand per customer per product per one month. This is not necessarily the demand that will materialize in the game (e.g. offered products are not good enough for the customers) but it will give the players an estimate of the size of each market area. The cumulated purchases column informs the players the customer specific materialized purchases accumulated from the beginning of the game.

The customers in the game are passive in that sense that they wait passively for offers from the companies taking part in the game.

CustomerNr	Customer	Product	MaxDemand/Month	CumulatedPurchases
1	London UH	Bio counter	500	
1	London UH	Bio counter DLX	250	
2	Bristol UH	Bio counter	400	
2	Bristol UH	Bio counter DLX	225	
3	Amsterdam UH	Bio counter	725	
3	Amsterdam UH	Bio counter DLX	450	
4	Bonn UH	Bio counter	400	
4	Bonn UH	Bio counter DLX	150	
5	Berlin UH	Bio counter	950	
5	Berlin UH	Bio counter DLX	675	
6	Frankfurt	Bio counter	1 025	
6	Frankfurt	Bio counter DLX	725	
7	Zurich UH	Bio counter	325	
7	Zurich UH	Bio counter DLX	175	
8	Milan UH	Bio counter	725	
8	Milan UH	Bio counter DLX	400	
9	Madrid UH	Bio counter	750	
9	Madrid UH	Bio counter DLX	550	
10	Roma UH	Bio counter	925	
10	Roma UH	Bio counter DLX	650	
11	Athens UH	Bio counter	750	
11	Athens UH	Bio counter DLX	425	

ProductName	Total
Bio counter	7 475
Bio counter DLX	4 675

Figure 16: The market area information window

The customer buying decision is based on several facts. The customer purchase decision depends (at the moment) on the game company:

- product sales price
- the term of payment the company is offering
- the delivery time the company is promising
- the amount of marketing investments
- product quality

Sales offers (including sales prices, maximum amounts of products that are offered for one customer per one transaction, term of payment, promised delivery time, and the offer valid until date) are maintained in the *Sales offers* window (figure 17). The offers are either market specific or common to all markets.

Market area	Sales price	Min/cust.	Max/cust.	TermOfPayment (d)	PromisedDelivery (h)	Valid until
Nordic	3 000,00	5	3000	17	13	
Europe	3 000,00	5	300	17	35	
North America	3 000,00	5	300	17	30	
Europe	3 200,00	1	100	14	8	
Nordic	3 200,00	1	100	14	14	
North America	3 300,00	1	100	14	30	

Blank market area means that the offer is valid for all market areas.

Buttons: Insert new offer, Delete offer, Close

Figure 17: The sales offers window.

The rest of the factors affecting the sales potential of the company are determined in the *Marketing investments* window (figure 18) and R&D window (figure 19). Marketing investments are market specific. The monetary figures are amounts per month.

	Nordic	Europe	North America
Advertising	100000	50000	100000
Direct marketing	22223	20000	30000
Sales promotion	33333	0	0
Public relationships	44444	0	0
Total	200 000	70 000	130 000
Total of totals	400 000		
Representatives	1	2	2

All investments are monthly investments. Investments take force when you accept them by selecting the Confirm button.

Buttons: Confirm, Close

Figure 18: The marketing investments window.

Product development takes place also as a monetary investment (figure 19). The participants can select a Product quality report which tells the product qualities of all companies (figure 20).

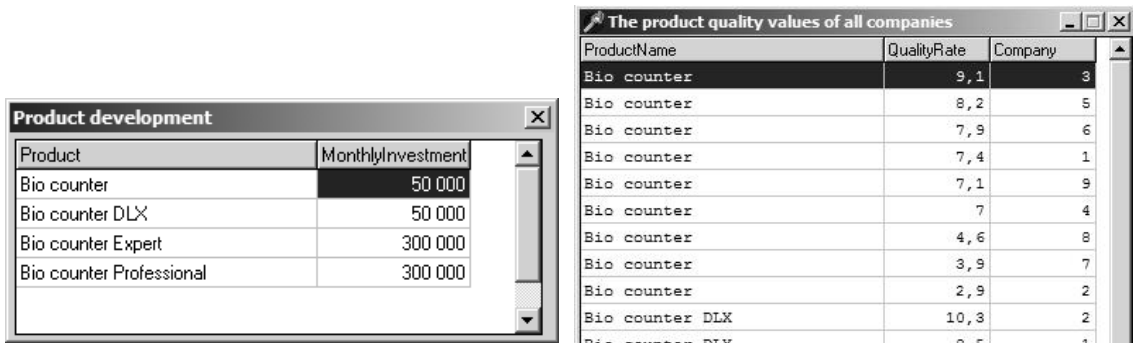


Figure 19: The product Development window. Figure 20: Part of the Product quality window.

As the customers order from a company the company receives an order. The *Handling of the open orders* window (figure 21) tells the players who has ordered, when the order took place, what are the sales terms our company has promised to the customer, and when the company has promised the delivery to be at the customer's (according to the offer). An essential feature in this window is that the information is transaction specific. Transaction specific information is seldom explicit in batch-processed business games because the algorithms in a batch-processed game are executed in a batch producing aggregate information about the transactions. Continuous processing produces explicit information about single transactions.

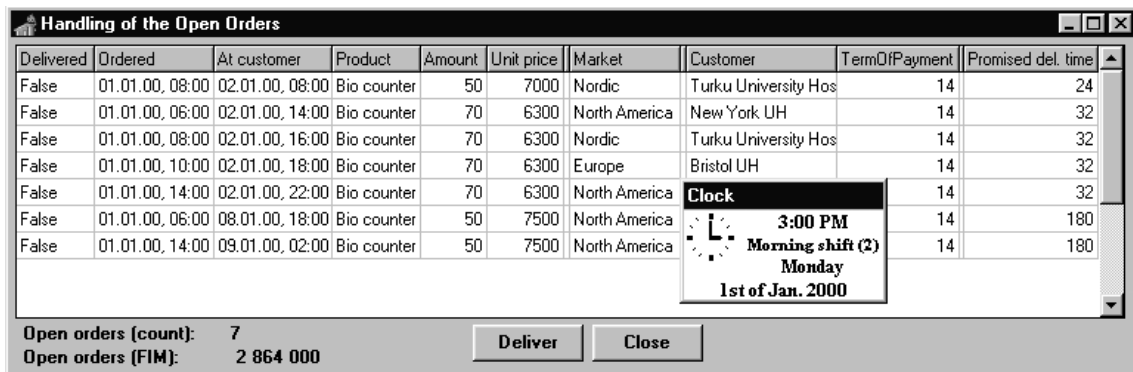


Figure 21: The handling of the open orders window.

The last two economic utilities (Coyle et al., 1996) are time utility and place utility. **Time utility** can be created through proper inventory maintenance. Not only must goods (and services) be available where customers need them, but they must also be at that point *when* customers demand them, i.e. economic value to goods (or services) by having them at demand point at a specific time. For example, having advertised products available for customers at precisely the time promised in the offer can create time utility. In REALGAME time utility is always present as all the transactions and operations are bound to the flow of time (e.g. customers expecting the goods to be delivered within the promised delivery time).

Place utility is provided by moving goods from production to points where demand exists. Extending physical boundaries adds economic value to the goods. In REALGAME place utility is created through transportation. The next screen copies demonstrate REALGAME functions that are related to time and place utilities.

REALGAME can proceed at three different speeds. We will now describe how the handling of open orders will be done when the game proceeds at the slowest speed (phase 1/3). In phases 2 and 3 the game handles open orders automatically, that is, the game delivers the orders automatically and sends the invoice to the customer whenever there are enough products in the inventory. However, in the slowest speed mode the players themselves have to deliver the orders: first they click the order to be delivered and then select the Deliver button. They will get the *Confirm delivery* window on the screen (figure 22).

<u>Deliver order</u>	
Product:	Bio counter
Unit price (/kpl):	7000.00
Market:	Nordic
Customer:	2, Turku University
<hr/>	
Ordered amount (kpl):	50
In store (kpl):	1136
Deliver amount (kpl):	<input type="text" value="50"/>
<hr/>	
Ordered date:	01.01.00, 08:00
Promised to customer:	02.01.00, 08:00
Promised delivery time:	24
Time to promised delivery (h):	16
<hr/>	
Select mode of transport:	<input type="text" value="Truck"/>
Estimated transfer duration: 12 hours.	
Transfer cost 50.00/haul + 0.09/kpl	
90% of transfers arrive in promised time.	
<input type="button" value="Deliver and invoice"/> <input type="button" value="Cancel"/>	

Figure 22: The confirm delivery window.

In the Confirm delivery window the players decide what is the amount of the product to be delivered and with what mode of transport to deliver. The different transport modes differ from each other by transport duration and cost. The delivery departs to the customer when the players select the Deliver and invoice button.

If the game is in the second or third phase then the deliveries will be done automatically by the CAP. However, the mode of transport selected by the CAP is still based on a decision by the players. The players may choose between the possible delivery modes in the *Available terms of delivery* window (figure 23). As the game delivers automatically it selects the transport mode according to the user chosen auto-delivery mode.

Market	MarketName	Delivery by	Duration (h)	Cost per delivery	Cost per unit
1	Nordic	Train	16	50	6
1	Nordic	Truck	12	50	9
2	Europe	Truck	28	150	19
2	Europe	Air freight	6	250	40
3	North America	Ship	168	1100	27
3	North America	Express air freight	18	2500	60
3	North America	Air freight	24	1000	44

ACTIVE DELIVERY METHODS:
1. Nordic: Duration
2. Europe: Fixed cost per delivery
3. North America: Duration

Change delivery method

Figure 23: The terms of delivery in use window.

We have now described the four principal types of economic utility in REALGAME, which add value to a product or service. The rest of the game functions are **support functions**, which do not create economic utility but are necessary for managing the company towards the desired direction. These support functions are related to *monetary flows, accounting procedures, and creating decision support* (reports) for managing the company. We will briefly introduce some of these functions.

As the order is delivered to the customer the customer becomes our debtor. The *Accounts receivable* window shows the players who their debtors are and how much the debtors owe them (figure 24). Furthermore, the window tells when the debts are to be paid to the companies. Once again, the participants see transaction specific information.

Date	Target	Action type	Explanation	Amount
16.01.00, 05:00	Bio counter	Sales earnings	Market area Nordic (1), customer 4	441000
16.01.00, 05:00	Bio counter	Sales earnings	Market area Europe (2), customer 4	441000
16.01.00, 05:00	Bio counter	Sales earnings	Market area North America (3), customer 3	375000
16.01.00, 07:00	Bio counter	Sales earnings	Market area North America (3), customer 4	441000
16.01.00, 07:00	Bio counter	Sales earnings	Market area North America (3), customer 4	375000
16.01.00, 09:00	Bio counter	Sales earnings	Market area Nordic (1), customer 5	350000
16.01.00, 11:00	Bio counter	Sales earnings	Market area Nordic (1), customer 5	441000
16.01.00, 13:00	Bio counter	Sales earnings	Market area Europe (2), customer 5	441000

Figure 24: The accounts receivable window.

If the players do not succeed to deliver the orders in time the deliveries may be late at the customer's. Customers will inform the companies about this with a message to the Bulletin board (figure 25).

01.01.00, 16:00: New orders received.
01.01.00, 18:00: New orders received.
01.01.00, 22:00: New orders received.
02.01.00, 00:00: New orders received.
02.01.00, 02:00: New orders received.
02.01.00, 17:00: customer 3 from market 1 informs us about a delivery (Bio counter) late for 0 days and 1 hours!
02.01.00, 17:00: customer 2 from market 1 informs us about a delivery (Bio counter) late for 0 days and 1 hours!

Figure 25: Customers informing company about late deliveries.

As time passes the debts of the customers will automatically be paid to our cash (*Online cash* window) and to the *Cash flow* window (figure 26).

Date	TargetObject	Brand	Sum
15.01.00, 16:00	Bio counter	Sales earnings	375 000
15.01.00, 16:00	Bio counter	Sales earnings	375 000
22.01.00, 19:00	Bio counter	Delivery costs	- 270
22.01.00, 19:00	Bio counter	Delivery costs	- 55
15.01.00, 22:00	Electronics	Raw material expenditures	-4 200 000
16.01.00, 05:00	Bio counter	Sales earnings	441 000
16.01.00, 05:00	Bio counter	Sales earnings	350 000
16.01.00, 05:00	Bio counter	Sales earnings	441 000
16.01.00, 05:00	Bio counter	Sales earnings	441 000
16.01.00, 05:00	Bio counter	Sales earnings	350 000
16.01.00, 05:00	Bio counter	Sales earnings	441 000
16.01.00, 17:00	Bio counter	Sales earnings	350 000
16.01.00, 19:00	Bio counter	Sales earnings	455 000
16.01.00, 19:00	Bank loan	Loan withdrawal, Merita	300 000
16.01.00, 23:00	Bio counter	Sales earnings	375 000
17.01.00, 03:00	Sensor	Raw material expenditures	- 650 000
17.01.00, 11:00	Bio counter	Sales earnings	350 000

Figure 26: The cash flow window.

During the game processes the companies may face cash deficit. If the companies do not collect the incomes fast enough from the sales they have to withdraw loans. Loan withdrawals are made in the *Funding* window (figure 27).

Funding X

Enter proper values for loan amount and loan term. When you are satisfied with the loan terms, select **Withdraw**.

Base interest

Loan amount

Loan term (months)

Monthly repayment (instalment + interest)

Cash limit (neg. cash) arrangement payment: 10 000

Cash limit interest per month (for neg. cash): 5,00

Confirm X

Do you want to withdraw the following loan: sum 1 000 000, interest 6,00% (per annum), loan time 12 months, monthly annuity (instalment and interest) 86 066.

Figure 27: The Funding window.

The CAP also updates the balance figures in real-time. Of course, this does not mean those accounting entries which are dealt with when the accounts are closed.

Assets		Liabilities	
	FIXED ASSETS	50 000 000	Capital
600 000	Intangible rights	201 208	Profit from previous periods
625 000	Land	872 640	Profit
27 000 000	Buildings	987 880	Provisions
21 801 863	Machinery	1 905 335	Bank loans
100 000	Other fixed assets		Accounts payable
200 000	Shares		
	CURRENT ASSETS		
3 390 200	Materials (Raw/Semi/Finished)		
	Accounts receivable		
250 000	Cash		

Figure 28: The balance sheet window.

The CAP also includes a possibility to calculate the profit on-line (figure 29). This calculation can be selected from the menu and the CAP calculates the financial statement according to the events materialized during the season.

Description	Value
TURNOVER	4 800 000

Change in inventories	6 648 005
Sales adjustment	0
Raw material purchases	- 8 660 000
Energy	- 12 980
Production variable expenses	- 899 600
Transportation	- 24 700
Advertising	- 336 606
Production fixed salaries	- 2 173 500
Production shift work additional expe	- 39 024
Administrative expenses	- 8 757 144
Depreciation	- 7 320 278

PROFIT BEFORE APPROPRIATIONS AND TAXES	- 16 775 827
Interests	- 1 042 059
Taxes (29%)	0

PROFIT	- 17 817 886
TotalEquity / Oma pääoma	536 640 000
Liabilities / Vieras pääoma	19 053 354
ROE (until today/tähän päivään asti)	- 3,3 %
ROE (whole year/koko vuosi)	- 29,7 %
ROI (until today/tähän päivään asti)	- 3,2 %
ROI (whole year/koko vuosi)	- 28,6 %
Profit-% / Voitto-%	- 371,2 %
Debt-equity ratio / Omavaraisuus	0,0
Open orders / Avoin tilauskanta (FIM)	0
Inventory / Varaston arvo (FIM)	23 701 531
Cash / Kassa (FIM)	- 19 247 891
Sales receivables / Myyntisaamiset (FIM)	4 000 000
Inv. turnover / Var. kiertonop. (per yea	5,8

Figure 29: The profit window.

Besides of standard financial statement the game offers diverse reporting possibilities. First, the reports describing our own company's sales efficiency can be seen in figure 30. The different reports describing market events and positions can be seen in figure 31. These reports are produced on-line (they will report the situation with a delay of a day or some days). Available reports of our company's efficiency in the materials functions can be seen in figure 32.

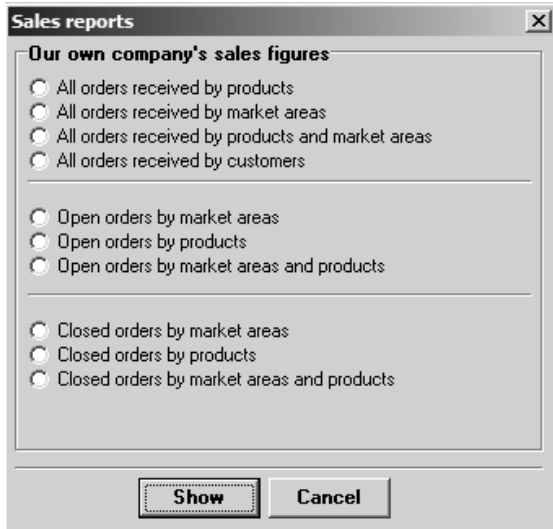


Figure 30: The sales reports window.



Figure 31: The market reports selection window.



Figure 32: The product and store reports window.

3.3.3 The Market Application (MAP)

The MAP interface is shown in figure 33.

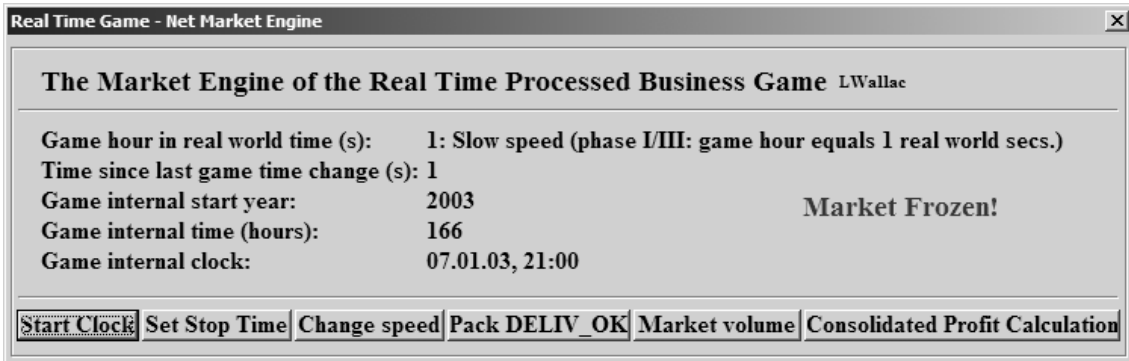


Figure 33: The user interface of MAP.

The execution of the MAP program code is straightforward. A Timer object in the MAP triggers the procedure including all the market routines. This is done in a batch-processed manner: the code is executed from the beginning to the end and then the MAP waits for the Timer to activate anew. In all, the MAP includes some 3,000 rows of Delphi programming code just to handle each game clock loop plus some 200 rows of window object definitions in a separate form file. In short, the processes the MAP executes in every game clock loop are the following:

- If the clock is on, increases the time with one hour.
- Reads all the decisions and other transactional information (offers, deliveries, market investments, R&D investments) of the participating companies.
- Browses through customer in each market and reads their demands. For every customer whose purchases are acute, browses through which company's offering (price, terms of payment and delivery, marketing investments, company image, and product quality) is most suitable and makes an order.
- Copies updated market information tables (market messages, time, new orders, general customer information, available delivery methods, company market images, environmental variables) to each company.

For the game administrator, the use of the MAP is quite simple. After the MAP is started the only thing the administrator has to do is either to stop or to start the execution of the MAP clock (the Stop Clock/Start Clock button in figure 33). The clock is stopped when the administrator needs the attention of the game participants to be focused on something else than the game, e.g. when some educational aspects of business decision-making are discussed together with all the participants.

If the administrator wants to change some environmental variables of the game this can be done easily by starting the Database Desktop application and accessing the needed database tables directly. This can only be done during the game if the MAP clock is stopped.

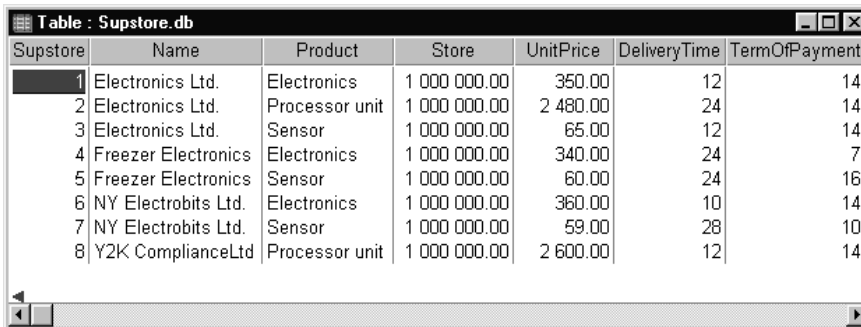
3.4 Configurability of the Game

The configurability of REALGAME concerns all the functions of the game. In this chapter we will clarify the meaning of configuration and how it will affect the user interface.

The configuration information is always saved and maintained in database tables. Maintaining the game configuration information is simple: the operator just opens the table including the configuration information with the Database Desktop and edits the information in the table.

3.4.1 Configuring the Supply Function

The configuration of the market raw material suppliers is done to a single Paradox table, which is maintained in the market server. The table layout and some example data are shown in figure 34. Besides of the fields in figure 34 the raw material renewal speed can be configured for each supplier/raw-material.



Supstore	Name	Product	Store	UnitPrice	DeliveryTime	TermOfPayment
1	Electronics Ltd.	Electronics	1 000 000.00	350.00	12	14
2	Electronics Ltd.	Processor unit	1 000 000.00	2 480.00	24	14
3	Electronics Ltd.	Sensor	1 000 000.00	65.00	12	14
4	Freezer Electronics	Electronics	1 000 000.00	340.00	24	7
5	Freezer Electronics	Sensor	1 000 000.00	60.00	24	16
6	NY Electrobites Ltd.	Electronics	1 000 000.00	360.00	10	14
7	NY Electrobites Ltd.	Sensor	1 000 000.00	59.00	28	10
8	Y2K ComplianceLtd	Processor unit	1 000 000.00	2 600.00	12	14

Figure 34: The SUPSTORE database table.

3.4.2 Configuring the Production Structure of Companies

The possibilities to configure the production line structure of the companies are:

- to determine the number of successive production cell phases and the number of parallel cells in each production phase
- to determine the internal structure of each production cell
- to determine which cells form a 'metacell': a group of cells in which only one cell at a time can be the active cell. This structure means that the metacell includes several parallel production recipes of which one at a time can be in use.

Configuring the Overall Production Line Cell Structure

The game company production line can include from 1 to 5 successive production phases. Each of these phases can consist from 1 to 9 parallel production cells. Figures 35 to 37 show some examples of the configuration of the manufacturing function. The first one is the game generic version (generic version used mainly to teach university students) model, the second one is a model configured for a glass manufacturer, and the third one is a production model for a high-tech instruments manufacturer.

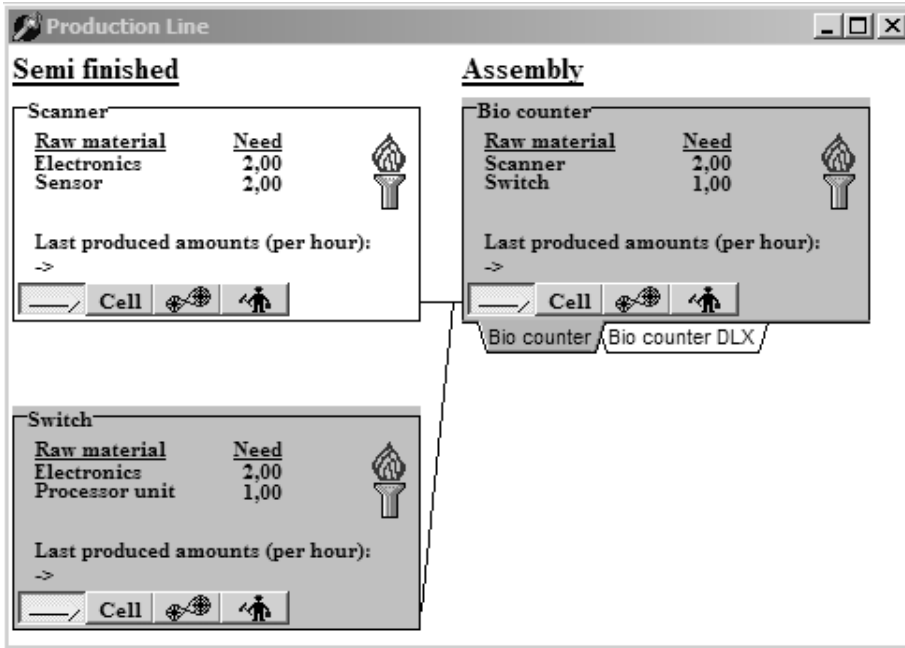


Figure 35: The Production line window in the game generic version.

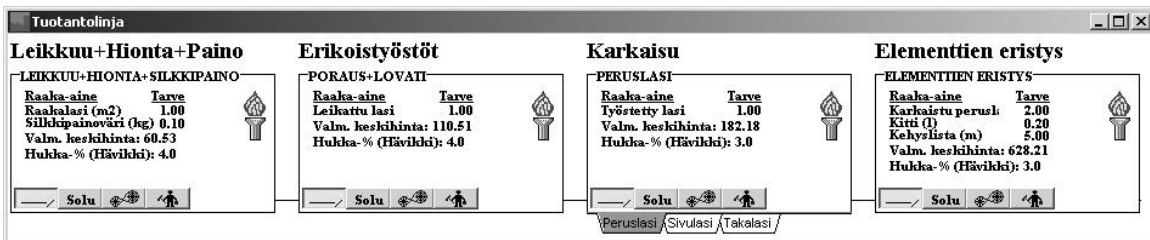


Figure 36: The Production line window of a glass manufacturer configuration.

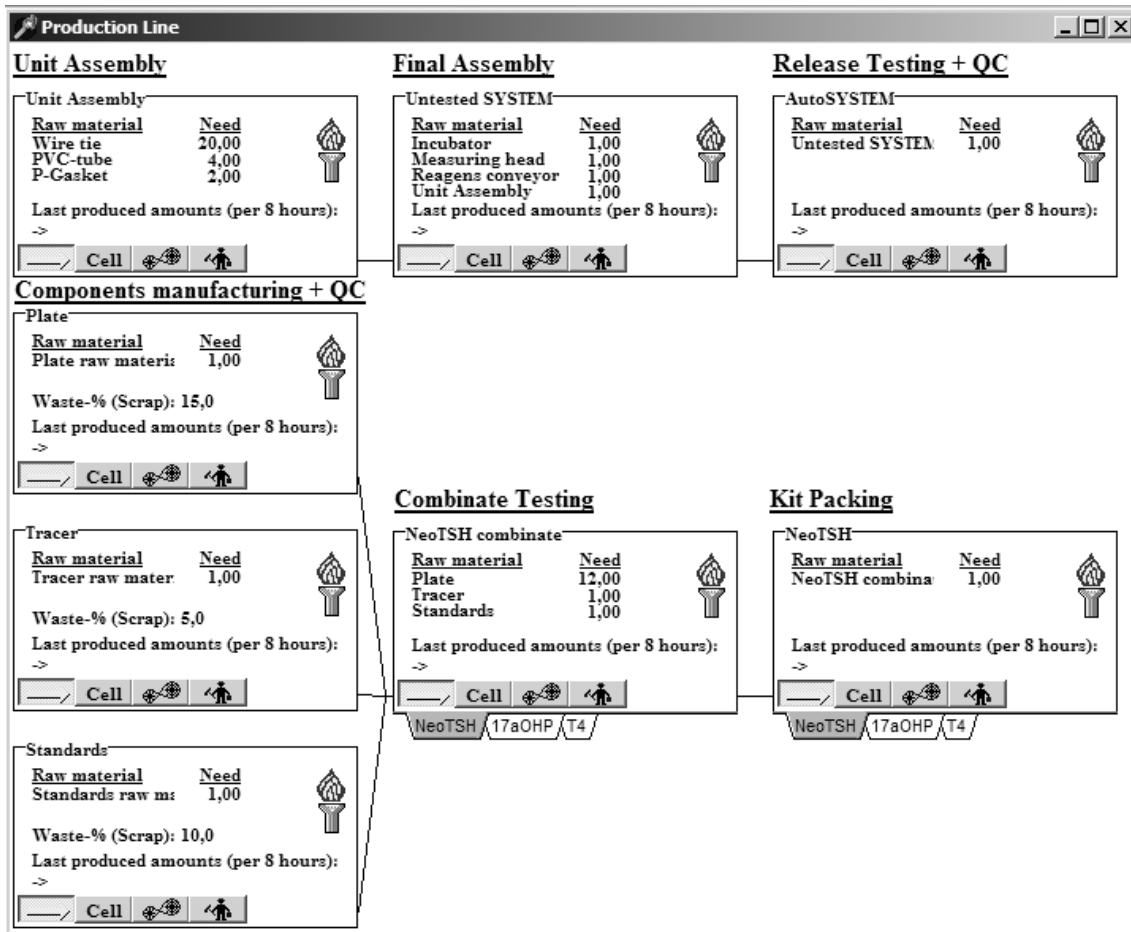


Figure 37: The Production line window of a configuration for a manufacturer of high-tech analysis systems.

The simplest possible configuration of the production line consists of a single production cell and the most complex one of 45 cells (5 phases times 9 cells). However, the present computer clock speeds limit the maximum amount of cells. The amount of production cells is one of the most significant factors affecting the processing time of one company application processing cycle.

Configuring the Internal Structure of Each Production Cell

The properties and functioning of a production cell is defined in a programmer defined Delphi object type. This object is inherited from Delphi standard objects. This object is the only significant programmer defined object in the game construction (though some database table rows, like customers in a market table, are often manipulated like objects). Other objects in the game applications are standard Delphi objects. The cell object consists of some 2,500 rows of program code. Each of the production cells in the production line window has its own table file. For example all the numerical information in figure 38 for the Scanner cell is saved in the cell's configuration database table.

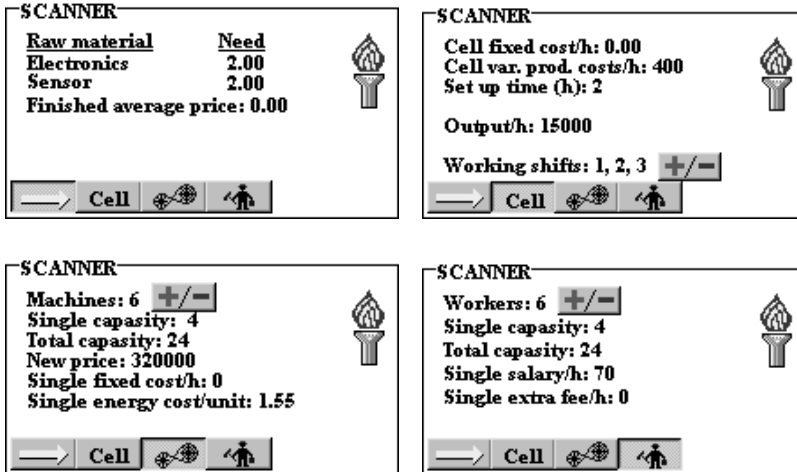


Figure 38: Four different views to a production cell. All the numerical cell info is saved as parameters in a database table.

Configuring a Production Metacell

In some production line cases the production cells need to have several alternative production recipes. Figure 39 represents a production metacell actually consisting of three separate production cells belonging to the same production cell group. Each of these three recipes can be turned on and at the same time as the recipe that has been on so far is turned off.

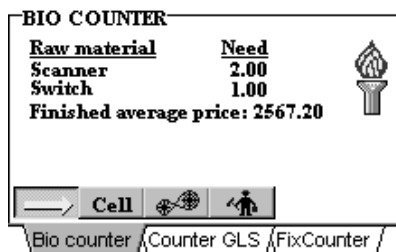
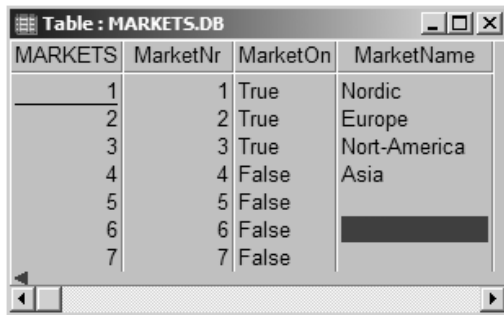


Figure 39: A production metacell consisting of three separate production cells.

3.4.3 Configuring the Customer Markets

The market structure of the game can be configured in two levels. First, the administrator can decide how many markets to be included in a game. This is done by entering the possible markets to the MARKETS database in figure 40.



MARKETS	MarketNr	MarketOn	MarketName
1	1	True	Nordic
2	2	True	Europe
3	3	True	North-America
4	4	False	Asia
5	5	False	
6	6	False	
7	7	False	

Figure 40: The MARKETS database table.

Column MarketOn specifies whether the customers in the market area are active or not. During the game it is possible to activate new market areas (by changing the value of MarketOn from False to True) or even to create totally new market areas by appending new records to the table. The amount of market areas is in practice limited only by computer processing capacity.

Secondly, the market area customers are defined in market specific customer tables (MARKET1, MARKET2, MARKET3,... MARKETN). Figure 41 describes the customer structure of market area 3, North America. The table rows are customer and product specific.

The figure consists of three screenshots of a database table named 'Table : MARKET3.DB'. Each screenshot shows a different view of the data.

Table 1: MARKET3.DB (Customer Purchase Data)

MARKET3	CustomerNr	ProductName	CustomerName	LastPurchaseTime	AvgVolumePerMonthIfGetIn
1	1	Bio counter	New York UH	-78,00	200,00
2	1	Bio counter DLX	New York UH	-78,00	175,00
3	2	Bio counter	Boston UH	-70,00	275,00
4	2	Bio counter DLX	Boston UH	-70,00	325,00
5	3	Bio counter	New Yersey UH	-62,00	425,00
6	3	Bio counter DLX	New Yersey UH	-62,00	300,00
7	4	Bio counter	Philadelphia UH	-54,00	300,00
8	4	Bio counter DLX	Philadelphia UH	-54,00	400,00
9	5	Bio counter	Pittsburgh UH	-46,00	425,00
10	6	Bio counter	Buffalo UH	-38,00	375,00

Table 2: MARKET3.DB (Purchase Weights)

HoursBetweenPurchases	MinGetInValue	CumulatedPurchase	WeightPrice	WeightTech	WeightAdvertising
84,00	0,00	0,00	70	100	80
84,00	0,00	0,00	70	100	80
84,00	0,00	0,00	70	100	100
84,00	0,00	0,00	70	100	100
84,00	0,00	0,00	100	100	60
84,00	0,00	0,00	100	100	60
84,00	0,00	0,00	30	100	50
84,00	0,00	0,00	30	100	50
84,00	0,00	0,00	100	100	100
84,00	0,00	0,00	70	100	30

Table 3: MARKET3.DB (Promotion Weights)

WeightSalesPromotion	WeightTermOfPayment	WeightTermOfDelivery
80	100	50
80	100	50
100	20	20
100	20	20
100	80	50
100	80	50
100	80	100
100	80	100
100	80	50
100	20	20

Figure 41: The MARKET3 database table.

The amount of customers in each market area is theoretically only limited by computer database storage capacity. However, the processing of customer purchases is clearly the most time consuming calculation task of the MAP. This means that increasing the amount of customers will slow down MAP execution fast and thus slow down the game internal time.

3.4.4 Configuring the Game Environmental Parameters

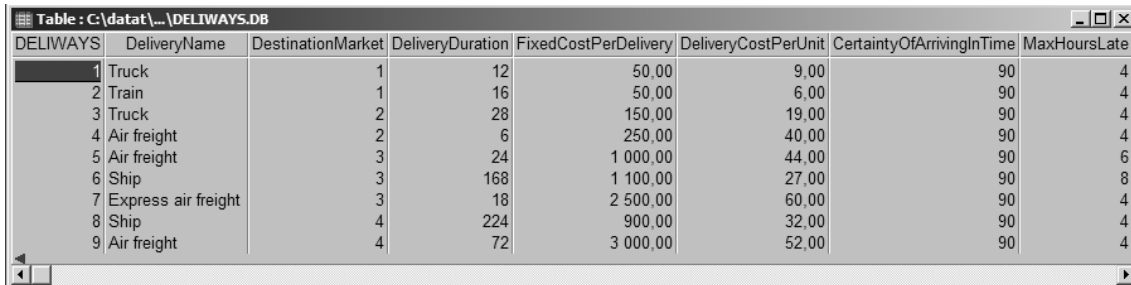
The game environment is configured in the ENVIRONM database table. As an example of what parameters there are, the fields of ENVIRONM are described in table 2.

Field	Type	Description
Language	A3	The language used in the game interface. At the moment the possibilities are Finnish and English.
Currency	A3	The abbreviation of the currency used in the game.
LineOfBusiness	A40	The line of business in the game.
ProdLine1	L	Not in use
ProdWorkerFeePerHour	I	The basic hourly salary of a production worker.
ProdWorkerHiringDelay	I	How many hours a production worker works after the hiring decision.
ProdWorkerSackingDelay	I	The delay (hours) from production worker sacking decision to the point the worker leaves.
ProdMachineBuyingDelay	I	How many hours it takes for the new production machine to start producing after the machine purchase decision has been made.
ProdMachineResaleValue%	I	The yearly depreciation (%) of production machine.
SalaryPayPeriodDays	I	The length (days) of salary pay period.
PenaltyLateDeliveryPerDay	N	How much the company customer specific image decreases (penalty per day) if the delivery is late.
PrizeForBetimesDelivery	N	How much the company customer specific image increases if the delivery arrives in time at the customer's.
MaxPenaltyShortDelivery	N	The maximum penalty of short deliveries (affecting the company customer specific image). The penalty is calculated: ((Ordered amount – Delivered amount) / Ordered amount) * MaxPenaltyShortDelivery
NrOfMarketsUsed	I	Number of markets in use in the game (the amount of active markets in MARKETS database).
CompaniesGaming	I	How many companies are taking part in the game event in question.
Shift1Name	A20	The name of the working shift from midnight to 8:00 AM.
Shift2Name	A20	The name of the working shift from 8:00 AM to 4:00 PM.
Shift3Name	A20	The name of the working shift from 4:00 PM to midnight.
AdvertisingInUse	L	Whether this marketing investment is in use and considered in the sales competition.
DirectMarketingInUse	L	As the previous.
SalesPromotionInUse	L	As the previous.
PublicRelationshipsInUse	L	As the previous.
RepresentativesInUse	L	As the previous.
ProductionUnit	A5	The abbreviation of the production material unit used in the game (unit, liter, kg,...).
RepoPricePerMarketProduct	N	Not in use.
HowManyGetsInToCustomer	I	How many companies can theoretically sell to one customer at a time.
NegCashOneTimeCost	N	How much the companies will pay (one time cost) if the cash balance becomes negative.
NegCashInterestPMonth	N	If the cash balance is negative how much interest (per month) will be charged for this negative cash balance.
FixedAdmExpenses	N	Fixed administrative expenses per month
AdmCostPerMachine	N	Administrative costs per production machine per month
CostOfCancellingOrder	N	Per cent costs (from the order total value) of cancelling an order
OrderLotSizesInUse	L	Whether supplier order lot sizes are in use in the game
BaseInterest	N	Base interest (rate) for loans
RnDinUse	L	Is R&D in use in the game
QualityErosion%PerDay	N	How much product quality will diminish per day
QualityInBeginning	I	Product quality value at the beginning of the game
RnDDecreaseAfter	N	The minimum amount that has to be invested in R&D per month in order to prevent the quality from deteriorating
MaxQualityRaiseInDay	N	How much product quality can maximally increase per day
InstallationCost	N	For immaterial products: how much a product's installation will cost for the company

Table 2: The fields of ENVIRONM database table.

3.4.5 Configuring the Market Specific Delivery Ways

The market area specific delivery methods and their costs are defined in the DELIWAYS database table (figure 42).



DELIWAYS	DeliveryName	DestinationMarket	DeliveryDuration	FixedCostPerDelivery	DeliveryCostPerUnit	CertaintyOfArrivingInTime	MaxHoursLate
1	Truck	1	12	50,00	9,00	90	4
2	Train	1	16	50,00	6,00	90	4
3	Truck	2	28	150,00	19,00	90	4
4	Air freight	2	6	250,00	40,00	90	4
5	Air freight	3	24	1 000,00	44,00	90	6
6	Ship	3	168	1 100,00	27,00	90	8
7	Express air freight	3	18	2 500,00	60,00	90	4
8	Ship	4	224	900,00	32,00	90	4
9	Air freight	4	72	3 000,00	52,00	90	4

Figure 42: The DELIWAYS database table.

4. CONCLUSIONS

Our original research hypothesis is stated:

An interactive (real-time/continuously processed) business game describes the interactive nature of the present business environment and sudden shifts in it, and how processes unfold in a company's internal and external environment. This verisimilitude (real-world likeness) of the processing method of the learning tool creates an authentic and meaningful environment enhancing participant business perception through game playing.

We have described some features of a new business game model. The practical construction building problems stemming from the hypothesis can be stated as:

- a) The business model should work in an interactive, real-time/continuously processed connection with the different markets.
- b) The business model should demonstrate the total business concept of a general manufacturing organization and its most significant stakeholders.
- c) It should be possible to configure the business model and the market model according to the requirements of the training event.

As the game model is now successfully constructed and some evidence exists about the successful use of the construction in training sessions (Lainema and Nurmi, 2002; Lainema, 2003b), we would like to say that the main research problem has been solved.

Naylor (1971) has stated about the development of business games:

Those readers who elect to develop their own game will find the experience an extremely rewarding one, since one cannot possibly write a computer program for a business game without a clear understanding of the complex interactions involved in the operation of a business firm. Such an experience is highly recommended for students in business administration and economics who want to test their real knowledge of the behaviour of business firms in a dynamic world.

We, as the developers of the new construction, feel that these words of Naylor are very true. During the construction process we have learned a lot about business processes and the mechanisms in the business environment. But what is even more important for us is whether this construction works as a successful learning tool.

Isaacs and Senge (1992) mention five directions of computer-based learning environment (CBLE) research, to enable CBLEs to be used both to discover more completely the nature of the manager's theories-in-use concerning complex situations, and as tools to produce changes in these theories-in-use:

- To examine what people take away from CBLEs and attempt to put into practice in their working environments.
- To study the extent and nature of changes in theories-in-use altered in work settings.
- What is the range and type of errors in learning behavior of which people are aware before and after the CBLE?
- What new behaviors are CBLEs able to produce?

We have studied these other research issues in our other works.

REFERENCES

- Chiesl, Newell E. (1990). Interactive Real Time Simulation. In Gentry (ed.) Guide to Business Gaming and Experiential Learning. ABSEL.
- Coyle, John J., Bardi, Edward J., and Langley, C. John (1996). The Management of Business Logistics. 6th edition, West Publishing Company.
- Huckfeldt, Robert R., Kohfeld, C. W., and Likens, Thomas W. (1982). Dynamic Modeling, An Introduction. Sage Publications, Inc.
- Isaacs, William and Senge, Peter (1992). Overcoming limits to learning in computer-based learning environments. *European Journal of Operational Research*, Vol. 59, No. 1, pp. 183-196.
- Lainema, Timo (1999a). Planning Guidelines for Next Generation Business Simulation. In Peter Juliff, Tsurayuki Kado, Ben-Zion Barta, editor, *Educating Professionals for Network-Centric Organisations*, Kluwer Academic Publishers, pp. 19-25.
- Lainema, Timo (1999b). What's Wrong with Business Games? Is Budget Based Decision-Making up to Date? In the proceedings of the Twenty Second IRIS Conference (Information Systems Research Seminar In Scandinavia), *Enterprise Architectures for Virtual Organisations*, Keuruu, Finland.
- Lainema, Timo (2000). Constructing a Real-Time Processed Business Game for Business Process Training. Turku School of Economics and Business Administration. An unpublished licentiate thesis.
- Lainema, Timo (2001). Enhancing Participant Business Process Perception through Business Gaming. Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS), IEEE.
- Lainema, Timo (2003a). Implications of Constructivism for Computer-Based Learning. Proceedings of 11th European Conference on Information Systems (ECIS), *New Paradigms in Organizations, Markets and Society*. Naples, Italy.
- Lainema, Timo (2003b). Reinforcing Information Systems Students' Perception on Business Processes. A working paper.
- Lainema, Timo, and Makkonen, Pekka (2003). Applying Constructivist Approach to Educational Business Games: Case REALGAME. *Simulation & Gaming: An Interdisciplinary Journal of Theory, Practice and Research*, Vol. 34, No. 1, pp. 131-149.
- Lainema, Timo, and Nurmi, Sami (2003). Applying an authentic, dynamic learning environment in real world business. A working paper.
- Lane, David C. (1992). Modelling as Learning: A consultancy methodology for enhancing learning in management teams. *European Journal of Operations Research*, Vol. 59, No. 1, pp. 64-84.
- Naylor, Thomas H. (1971). *Computer Simulation Experiments with Models of Economic Systems*. John Wiley & Sons, Inc.
- Thavikulwat, Precha (1996). Activity-Driven Time in Computerized Gaming Simulations. *Simulation & Gaming*, Vol. 27, Issue 1, pp. 110-122.

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

<http://www.tucs.fi/>



University of Turku

- Department of Information Technology
- Department of Mathematics



Åbo Akademi University

- Department of Computer Science
- Institute for Advanced Management Systems Research



Turku School of Economics and Business Administration

- Institute of Information Systems Science