Game Development Toolkit for Business People in Japan

Hiroaki Shirai, Motonari Tanabu, Takao Terano, Yasushi Kuno, Hisatoshi Suzuki and Kazuhiko Tsuda

Simulation Gaming 2003; 34; 437
DOI: 10.1177/1046878103255793

Game development toolkit for business people in Japan

Hiroaki Shirai
Motonari Tanabu
Yokohama National University
Takao Terano
Yasushi Kuno
Hisatoshi Suzuki
Kazuhiko Tsuda
University of Tsukuba

This article describes the authors' experience of developing a business gaming course for business people in Japan. The course that was developed consists of (a) simple gaming experiments among multiple students using ALEXANDER ISLANDS, a tiny business game on the World Wide Web; (b) lectures to make the students understand the core concepts of business management through the simulation; and (c) a business game developed by the students themselves using business model description language (BMDL), business model development system (BMDS), and automated agent players (AAPs). The course is distinctive in the sense that, through the business gaming, the authors aim to develop students' skills (a) to implement their own models for specific business firms and (b) to understand business processes among companies. This article describes the background and motivation, basic principles, the architecture and implementation of BMDL/BMDS/AAPs, and the experimental results.

KEYWORDS: *business game; development toolkit; software agent; WWW environment*

This article reports an unconventional approach to a business gaming course at the Graduate School of Systems Management (GSSM) of Tsukuba University in Japan. The goals of the course are to develop students' skills (a) to implement their own models for specific business firms and (b) to understand business processes among companies. To achieve the goals, the distinctive point of the course is to let the students develop their own business simulators using a simple Business Model Description Language (BMDL), Business Model Development System (BMDS), and Automated Agent Players (AAPs). The developed simulators are run and evaluated by the other members of the class, as well as lecturers on the WWW environment. Through the experience, the students will understand the concepts and skills of business management. This article describes the background and motivation, basic principles, the architecture and implementation of BMDL/BMDS/AAPs, and some results of our school's 4 years of course experience.

SIMULATION & GAMING, Vol. 34 No. 3, September 2003 437-446
DOI: 10.1177/1046878103255793

Background and motivation

Experiential learning is a method to teach principles or facts to students not only through giving lectures but by letting students do some actions by themselves. Learning by doing is an effective means for various fields of education (Burns & Gentry, 1998; Herz & Merz, 1998; Ruben, 1999; Shirai, 2001). Business games are powerful tools for learning business structures and principles by making decisions. The conventional approach to business gaming courses, at least in Japan, is to educate inexperienced students to understand the concepts of management, accounting, business processes, and/or the basic techniques for business analysis. There are many conventional gaming simulations in the literature that aim to attain these goals (Elgood, 1993; Greenblat, 1989; Rye, 1994). However, at GSSM, such approaches are insufficient. The students we have are all businesspeople from various industries. They have different levels of expertise and different career backgrounds. Therefore, although the academic levels of the students are so divergent, playing-only simulators will not satisfy them. They want to know how to make sound management decisions. To meet the requirements and based on our previous experience (Fujimori, 1993a, 1993b; Shirai, 1994, 1996), we set the following goals for the design of the business gaming course:

1. The game we use is simple enough that business people with various backgrounds, especially those who have limited accounting knowledge, can understand the basic process of business simulations.
2. However, it is also complex enough that the students who have strong practical business experience can learn the advanced concepts of decision-making and business processes.
3. The students should be motivated to study advanced courses at GSSM through using the simulators. These courses include operations research, information systems, decision theory, accounting, marketing, computer programming, artificial intelligence, and/or computer networks.
4. A dozen students can execute the simulators at the same time in different places. Therefore, the simulator should be able to run on a computer network or on the World Wide Web.
5. The students should be able to develop their own business models with ease. The BMDL should be simple enough so that the students with few computer-programming skills can understand it and write their own models.
6. A typical business game should be developed to guide the students. The game should be executable as the introductory tool for the course and readable so the students can understand how the model is built and how the simulation is executed.
7. The business game simulator should include software agent models. The agents can be used as a substitute for human players. Using the agents, developers can debug their own models with ease. Also, we can model many kinds of business decision rules using the agents, and we can evaluate these rules.

Based on the above, the course we have developed consists of (a) a simple gaming experiment among multiple students using ALEXANDER ISLANDS, a business game on the WWW; (b) lectures to teach students the core concepts of business management; and (c) a business game development by the students themselves using BMDL, BMDS, and AAPs.

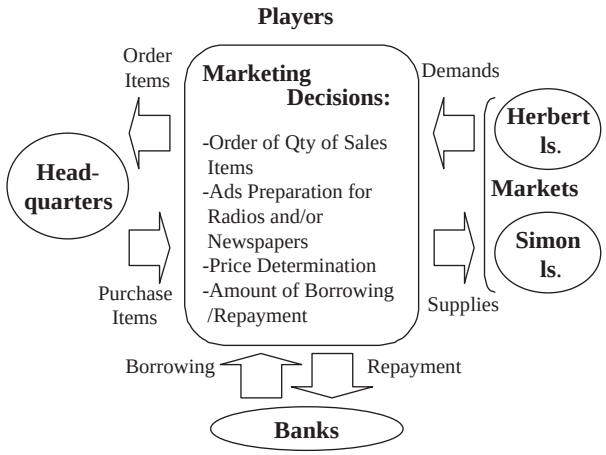

FIGURE 1 Business Model of ALEXANDER ISLANDS

ALEXANDER ISLANDS: A tiny business game

ALEXANDER ISLANDS (Fujimori, Kuno, Shirai, Suzuki, & Terano, 1999; Terano et al., 1999) is an introductory business game developed to meet Items 1, 2, and 6 of the requirements in the previous section. This game models a business process of (a) purchasing consumer items (e.g., personal computers) from the headquarters of a fictitious company, (b) selling them to two different markets (Herbert Island and Simon Island), and (c) making marketing decisions on ordering, advertising, pricing, and accounting.

The business model of ALEXANDER ISLANDS is summarized in Figure 1. During each turn of the game, each user can order any amount of a sales item from the headquarters, prepare advertisements for radio and/or newspapers, determine the sales prices for the two markets, and make accounting decisions on borrowing and/or repaying money. The model implicitly includes a stock management problem from operations research, demand forecasting from decision making, and advertising strategy from marketing.

Although the simulator does not include explicit decision support functions, it is desirable for the students to be aware of the existence of such practical problems in the simple gaming environment. Therefore, the players should have some decision aids on a PC (e.g., a spreadsheet program) to properly make decisions during the game. As the game is simple to use, it takes 10 minutes even for inexperienced users to make decisions. Usually the time to complete one game round is about 70 minutes, or seven to nine terms.

ALEXANDER ISLANDS contains the basic mechanisms of the business process and user interfaces for 10 players/teams and one manager. The user interfaces run on a

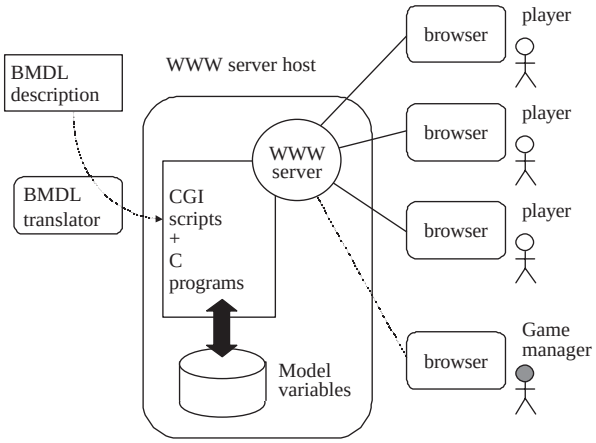

FIGURE 2 System Configuration

Web browser. The first version of the simulator was implemented in the C language, and reimplemented in BMDL in 1997. The source codes of ALEXANDER ISLANDS currently consist of about 450 steps of BMDL instructions including detailed comments. The length is short enough that the students can read the contents and understand how they work.

Framework of BMDL/BMDS

This section describes the architecture of BMDL and BMDS. BMDL/BMDS framework was designed to meet Items 3, 4, and 5 of the requirements in the Background and Motivation section. Figure 2 shows the system configuration. Sample codes of BMDL are shown in Figure 3. The example in Figure 3 is the code for a business game designed for three players and one manager for 10 game rounds. The game is executed as follows: At the beginning, each player inputs the sales price; the cheaper the price is, the more goods he or she can sell; and a decision is made to supply new goods to keep the same stock level.

The interfaces for players to give prices and the ones for the game manager to browse the status of all players are also included in the BMDL code. In the original codes, the variables of the model are described using Japanese characters. However, in Figure 3, they are translated to capital ASCII characters for this article. The mechanism enables the users to easily develop a homemade business game.

A business game written in BMDL by a model developer is translated into CGI script and C programs by a BMDL translator. The programs run on a host computer with a WWW server and the model variable data in the form of spreadsheets. Finally, players execute the simulation through browsers in the WWW environment. The simulation is executed with the architecture shown in Figure 4. The game manager

```
#
# NOTE: Original codes includes KANJI characters to denote variables.# Size of the Game#
def game-name Price Onlydef max-team 3def max-round 10password ctr t1 t2 t3
# Series constants definition
scon DEMANDS 497,1195 2447 4037 5406 6626 8177 9451 9945 10713
# Global constants definition
gcon ORDER_PRICE 90
gcon MINIMUM_PRICE 50
#
# Input Variables and Input Window definition
ipage price INPUT_PRICES
<H1>Input Price</H1>
<P>Input Sales Price</P>ivar SALES_PRICE range 0 1000 120# Model variables and their
initial values of each team.tvar NUMBER_OF_SALESvar PRICE_OF_PROCEEDtvar
PRICE_OF_ORDERtvar AMOUNT_OF_PROFITtvar AMOUNT_OF_DEPOSIT150000
# Computation Model Description
pinv NUMBER_OF_SALES = DEMANDS by SALES_PRICE - MINIMUM_PRICE;
tlet NUMBER_OF_SALES = rint(NUMBER_OF_SALES);
tlet PRICE_OF_PROCEED = NUMBER_OF_SALES * SALES_PRICE;
tlet PRICE_OF_ORDER_ = NUMBER_OF_SALES * ORDER_PRICE;
tlet AMOUNT_OF_PROFIT = PRICE_OF_PROCEED - PRICE_OF_ORDER;
tlet AMOUNT_OF_DEPOSIT = AMOUNT_OF_DEPOSIT@1 + AMOUNT_OF_PROFIT;
# Output description
ooption fmt %1.0lf
opage sales SALES STATUS_public
  <H1>SALES_STATUS_REPORT</H1>
  <P> $_ROUND-th: Demands: $ DEMANDS</P\>
begintable
out teams
out teams-vars SALES_PRICE::%5.3lf NUMBER_OF_SALES
PRICE_OF_PROCEED PRICE_OF_ORDER AMOUNT_OF_PROFIT
endtable
#
opage balance PROFIT STATUS teamspec
  <H1>PROFIT_STATUS REPORT</H1>
  <P> $_ROUND-th, TEAM: $ TEAM, TOTAL_DEMANDS: $ DEMANDS</P>
begintable
out values 'ITEM 'INCOME 'OUTGO
out values 'LAST_TERM_DEPOSIT AMOUNT_OF_DEPOSIT@1 -
out values 'AMOUNT_OF_INCOME PRICE_OF_PROCEED -
out values 'AMOUNT_OF_OUTGO - PRICE_OF_ORDER
out values 'AMOUNT_OF_PROFIT - AMOUNT_OF_PROFIT
out values 'THIS_TERM_DEPOSIT - AMOUNT_OF_DEPOSIT
endtable
#
opage allvteam ALL_VARIABLES_THROUGH_TEAMS control
  <H1>$-ROUND ROUND: ALL_VARIABLES_THROUGH_TEAMS </H1>
begintable
out teams
out teams-allvars
endtable
#
opage allvteam ALL_VARIABLES_THROUGH_TEAMS control
  <H1>TEAMS-TEAM: ALL_VARIABLES_THROUGH_TEAMS </H1>
begintable
out rounds
out rounds-allvars
endtable
#
# end
#
```

FIGURE 3 Sample Codes of BMDL

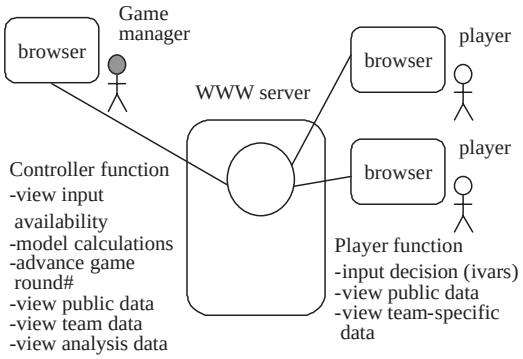

FIGURE 4 Simulation Architecture

controls game rounds, calculations, and analysis data. The game players can input and modify the input variables based on public and team-specific information. Figure 5 shows a sample entry window of the game.

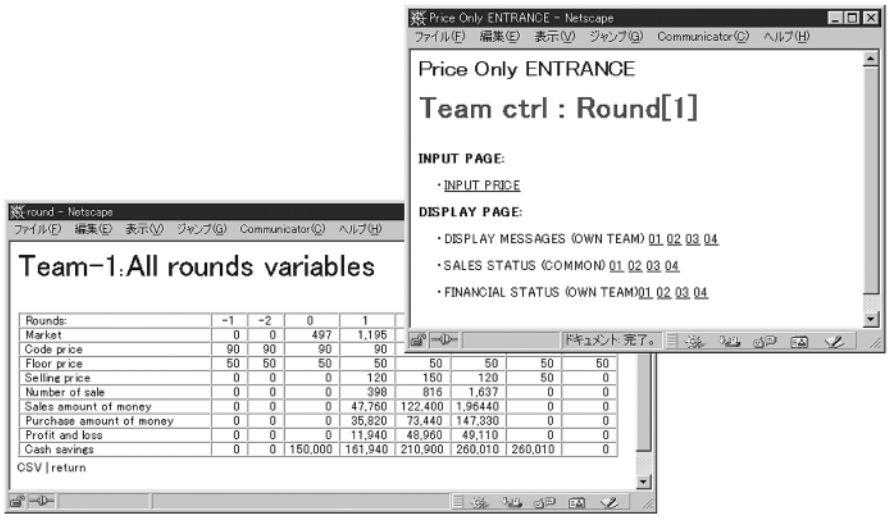

FIGURE 5 Sample Entry Window of Game

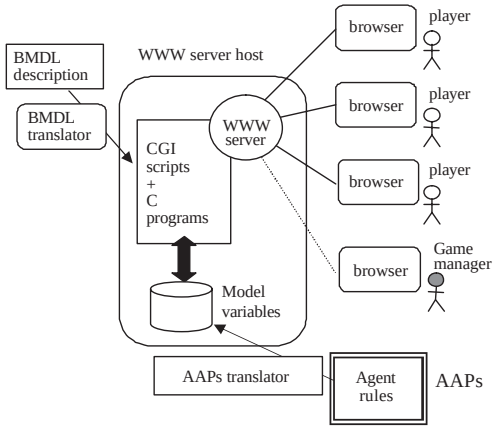

FIGURE 6 System Configuration

Framework of AAPs

This section describes the architecture of the AAPs. AAPs were designed to meet the requirements of Item 7 in the Background and Motivation section. Figure 6 shows the system configuration. Automated software agents participate in the business game as virtual players. Model variables generated by AAPs are managed in the corresponding model database. AAPs in the form of production rules are executed by the translator written in Java. Sample code of the AAPs is shown in Figure 7. Model developers

```
#Agent for Sales Model
#
# Valuables definition
iftrue  print  SALES_PRICE  print AMOUNT_OF_PROFIT
#
#Production Rules  :   @n means past period
#Discount
if  (AMOUNT_OF_PROFIT>10000)
 SALES_PRICE = SALES_PRICE@1  *  0.9
#Copy Top Team
iftrue  SALES_PRICE = t_val(SALES_PRICE@1  max_t(AMOUNT_OF_PROFUT@1))
#
```

FIGURE 7 Sample Code of the AAPs

can describe their software agent with AAPs, and they can debug their own models easily. Also, model developers can model many kinds of business decision rules using the agents. Then, they can evaluate those rules through business gaming.

Experience and discussion

Outline of the course experience

The course was started in 1996. At the time, we had no specific development toolkit. ALEXANDER ISLANDS in the C language was used in the introductory business gaming course. The students were not pleased because there were too many supplementary tasks needed to understand the business management principles.

In 1997, the BMDL version was introduced for an advanced course to develop the students' own business models. The program was improved so that the students could understand the codes of ALEXANDER ISLANDS easier and could develop their own games. To improve the program, questionnaire data on the course were gathered. Among the 23 respondents to the questionnaire, (a) 22 students replied that the course using ALEXANDER ISLANDS was interesting, (b) 17 students thought that the difficulty level of the course was moderate, and (c) 18 students were comfortable with the WWW environment. The models that students developed cover a wide range of business domains, such as production, research and development, logistics, and accounting.

Mini-P model in the 1998 program

In the 1998 program, we provided students with a simple business model: Mini-P (small production company) that runs on BMDS. The purpose was to let students use it as a prototype or a sample source code for their own projects. The model was originally

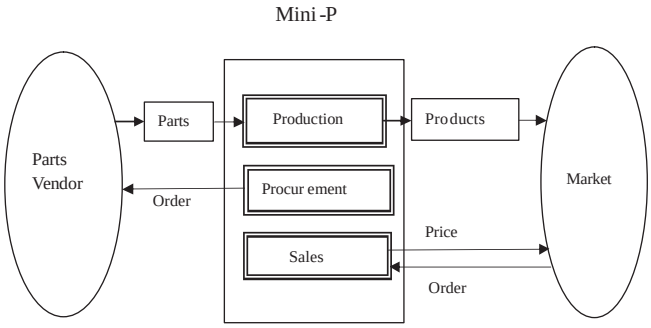

FIGURE 8 Business Model of Mini-P

designed to be used as a small case study in our course that simulates a production firm. Although it is simple, it can be extended to meet the student's complex requirements, for example, multiproducts and research and development. The business model of the Mini-P is summarized in Figure 8, and specifications are summarized as below. The corresponding source code in BMDL contains about 100 lines of code, including comments on the case. The specifications of the Mini-P model are as follows:

1. kind of products: only one product, A.
2. kind of material parts: only one part, a.
3. decision variables: (a) sales view: price of product A; (b) production view: number of product A; and (c) number of order a.
4. how to order parts: (a) parts price: constant; (b) number of parts: provided number exactly required; and (c) lead-time: one term.
5. how to make a product: (a) the product consists of 50 parts; and (b) lead-time: one term.

Evaluation

In the 1998 program, we think we attained most design goals described in the Background and Motivation section. We have received the following responses from questionnaire data from 15 respondents:

1. All of the respondents replied that the course with the ALEXANDER ISLANDS was interesting and that they recommended to the course to new students.
2. Most of the respondents replied that BMDL/BMDS is simple enough that students with few computer programming skills can understand it and write their own models.
3. Eleven respondents thought the level of the course was moderate.

In the 1998 program, 18 students developed their own models based on the Mini-P model, and 13 models were developed in the same way in the 1999 program.

In the 1999 program, we implemented the first version of the AAPs. Our early software agent players achieved good results of debugging models, but they did not achieve good results in the game playing with human players. The reason was caused

by poor production rules implemented in AAPs. More sophisticated rules will improve the result. We have been working on improving the functions of AAPs.

From these observations, we believe that we can teach business processes we would like to model and analyze, using our BMDL/BMDS/AAPs toolkit.

Conclusion

This article has described our experience of developing an unconventional course on business gaming for business people in Japan. Conventional business gaming courses aim at studying the basic principles of simulated business processes, accounting, or data analysis techniques. But the objectives of our course are to develop students' skills (a) to understand business processes and (b) to implement their own specific models of their companies.

To attain the objectives, the most remarkable point is the development of do-it-yourself models using BMDL/BMDS/AAPs. This approach is promising for business simulator courses to develop more advanced systems and concepts. Now we are working on improving the functions of the toolkit to open it to the public in the near future (Shirai & Tanabu, 2003).

References

Burns, A. C., & Gentry, J. W. (1998). Motivating students to engage in experiential learning: A tension-to-learn theory. *Simulation & Gaming, 29*(2), 133-151.

Elgood, C. (1993). *Handbook of management games*. London: Gower Press.

Fujimori, H. (1993a). Application of business games in Japanese companies. *Pasokon Literacy, 18*(12), 17-24.

Fujimori, H. (1993b). Design and operation of business games on a network environment. *Journal of Japan Association of Simulation and Gaming, 3*(1), 15-24.

Fujimori, H., Kuno, Y., Shirai, H., Suzuki, H., & Terano, T. (1999). ALEXANDER ISLANDS: GSSM tiny business simulator on the WWW. *ABSEL*, 224-225.

Greenblat, C. S. (1989). *Designing games and simulations*. Newbury Park, CA: Sage.

Herz, B., & Merz, W. (1998). Experiential learning and the effectiveness of economic simulation games. *Simulation & Gaming, 29*(2), 238-250.

Ruben, B. D. (1999). Simulations, games, and experience-based learning: The quest for a new paradigm for teaching and leaning. *Simulation & Gaming, 30*(4), 498-505.

Rye, D. E. (1994). *The corporate game: A computer adventure for developing business decision-making skill*. New York: McGraw-Hill.

Shirai, H. (1994). Gaming simulation for management consulting. *Journal of Japan Association of Simulation and Gaming, 4*(1), 129-137.

Shirai, H. (1996). Gaming simulation for problem setting. *Journal of Japan Association of Simulation and Gaming, 4*(1), 85-92.

Shirai, H. (2001). Proposal of teaching material using gaming simulation for experiential learning. *Journal of Japanese Society for Information and Systems in Education, 18*(1), 34-41.

Shirai, H., & Tanabu, M. (2003). A do-it-yourself modeling toolkit using Web technology [CD ROM]. *International Simulation and Gaming Yearbook, 11*.

Terano, T., et al. (1999). Understanding your business through home-made simulator development. *ABSEL*, 65-71.

Hiroaki Shirai is a professor of business administration at Yokohama National University in Japan.

Motonari Tanabu is an associate professor of business administration at Yokohama National University in Japan.

Takao Terano is a professor of the Graduate School of Systems Management, University of Tsukuba in Japan.

Yasushi Kuno is a professor of the Graduate School of Systems Management, University of Tsukuba in Japan.

Hisatoshi Suzuki is a professor of the Graduate School of Systems Management, University of Tsukuba in Japan.

Kazuhiko Tsuda is an associate professor of the Graduate School of Systems Management, University of Tsukuba in Japan.

ADDRESSES: HS: 79-4 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, Japan; telephone: +81 (45) 339-3713; fax: +81 (45) 339-3713; e-mail: shirai@shirai.org. MT: 79-4 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, Japan; telephone: +81 (45) 339-3713; fax: +81 (45) 339-3713; e-mail: tanabu@business.ynu.ac.jp. TT: Bunkyo-ku 3-29-1, Tokyo, 112-0012, Japan; telephone: +81 (3) 3942-6877; fax: +81 (3) 3942-6829; e-mail: terano@gssm.otsuka.tsukuba.ac.jp. YK: Bunkyo-ku 3-29-1, Tokyo, 112-0012, Japan; telephone: +81 (3) 3942-6877; fax: +81 (3) 3942-6829, e-mail: kuno@gssm.otsuka.tsukuba.ac.jp. HS: Bunkyo-ku 3-29-1, Tokyo, 112-0012, Japan; telephone: +81 (3) 3942-6877; fax: +81 (3) 3942-6829; e-mail: suzuki@gssm.otsuka.tsukuba.ac.jp. KT: Bunkyo-ku 3-29-1, Tokyo, 112-0012, Japan; telephone: +81 (3) 3942-6877; fax: +81 (3) 3942-6829; e-mail: tsuda@gssm.otsuka.tsukuba.ac.jp.