# Generating Explanations for Internet-based Business Games

M. Fischer and M.Lusti
University of Basel, Faculty of Economics, Department of Information Systems, Basel, Switzerland

***Abstract*—It is widely established debriefing in business games is important and influences the students' learning performance. Most games only support game statistics instead of explaining solution paths. We suggest the automatic generation of explanations for internet-mediated business games to improve the debriefing quality. As a proof of concept we developed a prototype of an internet-based auction game embedding an open simulation model and an automatic explanation component helping students and teachers to analyse the decision making process. This paper describes the usefulness of automated explanations and the underlying generic software architecture.**

***Index Terms*—Computer based learning (CBL), New learning models and applications, Web based learning (WBL)**

## I. Introduction

Business games teach various economic skills in an exciting way and are an established training method in business management, logistics and military. Twenty-five years of research show that students often prefer business games to other teaching approaches [1].

Each business game consists of three elements: role, simulation and game. The role places the player in a decision-making situation, the simulation defines the learning subject and the game motivates the player within a risk-free environment.

Since the first widely-used game developed by the American Management Association in 1956, technology (such as computers or internet) is the biggest factor driving the popularity of business games [2, p. 17]. Internet-mediated business games get more and more important because of advantages like time and location independence.

On the other side internet-based business games can accentuate the limitations of current business games. In their account of the literature Dobson et al. [3] list six limitations: lengthy durations, high training costs, lack of timely feedback, lacking technology to support debriefing, unintelligible game output, and poor emotional engagement. The first two can be reduced by porting the game to a network. The major drawback, however, is that the learner gets insufficient feedback, i.e. the important debriefing process is often ignored.

There are different solutions to this problem. The most popular is blended learning which uses presence lessons for the debriefing phase. It diminishes some advantages of a pure online game but combines the strength of online and face-to-face approaches. Blended learning is very practicable for institutions which have access to existing infrastructure at near distance from the learner. Alternatively, the interaction may be enhanced by 24 hour support and communication technologies like internet video conferencing, see e.g. MINT [4, p. 144]. Such an approach, however, requires a good communication infrastructure and IT experienced students.

We use automatically generated explanations to support debriefing and to give a timely feedback to students. To let the application generate explanations has many benefits, including: fast response to questions, reduced fear of asking questions, lower teacher workload and increased learning motivation with a transparent model. The downside is more complex software architecture, and therefore a higher implementation cost.

Automatically generated explanations for e-learning have been introduced by Lusti in the eighties. His *FATUTOR*, for example, teaches financial analysis by supporting protocol driven hypertext explanations [5]. So far, the idea has not been adopted by business game authors primarily because the integration into an internet based business game architecture is not trivial.

*Full Spectrum Command*, a technologically advanced war game developed for the US army, uses generated explanations. But it cannot unleash its full power because important declarative information is missing. Moreover, the explanation component could not be fully integrated in the simulation because it was not considered in the original design [6].

As a proof of concept we developed *ProfiBieter*, a business game for auction theory with an integrated explanation component. Its core is designed for reuse across a defined range of business games.

We have chosen elementary auction theory to illustrate our ideas because it meets the demands of a rule based and multi-player simulation model. Moreover, auction theory is part of the curriculum in our Economics department.

*ProfiBieter* teaches how …

- to find the appropriate auction format for a given problem text
- to execute the optimal bidding strategy
- to recognize the winner's curse.

The winner's curse describes the phenomenon that the winner of an auction pays more than the objective value. This overestimation can occur in auctions where the value of a good is uncertain. Assuming that the average of a bidder's valuation corresponds to the objective value then the winner will overestimate the value of the good if she does not bid down from her estimated value.

This paper uses a simple auction format, the so called 'English Private Value Auction', as an example. In an

English auction the auctioneer increases the price until only one bidder remains. The good is then sold to the value of the second highest bidder. The private value indicates that the bidder knows the value of the object to herself and that this value is not influenced by other bidders. Obviously, the optimal strategy for this auction format is to bid up to one's private value [7].

First, we look at the possibilities to include feedback in business games. Subsequently, we describe the implementation. The final section draws some conclusions.

## II. Introduction

To incorporate feedback into internet-based business games you have to decide on its content, availability and display. Feedback research shows feedback has a positive effect on learning efficiency and is influenced by these considerations [8]. This section analyses more closely the feedback mechanism of ProfiBieter.

### *A. Content*

The literature distinguishes between verification and elaboration feedback. Verification feedback judges whether an answer is right or wrong and is further categorised into 'Knowledge of the Result', 'Knowledge of the Correct Result' and 'Answer until Correct'. 'Knowledge of the Result' tells the learner only the correctness of her response. This kind of feedback does not elucidate failure or success. 'Answer until Correct' lets the learner try until she gets the correct answer, while 'Knowledge of the Correct Result' immediately displays the answer.

In *ProfiBieter*, 'Knowledge of the Correct Result' is the entry point for optional explanations. Because a bidding strategy is seldom completely right or wrong the feedback shows first the degree of conformity with the computed solution. This grading is more motivational than educational and should awake the learner's interest to delve deeper.

Displaying the correct result first and accessing the elaborated feedback on demand was dissatisfactory in Teslow's empirical study [9]. Only 13% (10%) of his students opted to review the feedback after incorrect (correct) response, further the average time per review was approximately 13 seconds. Thus, valuable alternatives are forcing the student to undergo feedback or displaying the correct result after the feedback.

*ProfiBieter* presents the solution first because the player has an incentive to look at it. Teslow tests factual knowledge on instructional objectives with multiple choice questions. For such questions the correct result often conveys enough information. For example, answering "which is the capital of Switzerland" does not necessitate any feedback except the correct result. In contrast the correct auction strategy alone does not contribute a lot to the learners understanding. Because of repeating auctions the player has to learn from his faults and needs the elaborated feedback to succeed in the next auctions.

For the elaboration process *ProfiBieter* uses how-explanations. The question how the system finds the correct auction strategy is in the learner's foremost interest. The explanations are ordered hierarchically from the solution to the problem. The learner can navigate through this explanation tree accessing intermediate results. The end nodes of the explanation tree refer to the highlighted facts of the problem text. Another explanation type is hypothetical reasoning (what-if explanation). Under hypothetical reasoning the user can modify premises, i. e. the system answers questions like "what is the correct bid if the auction format is a Dutch[1] instead of an English auction". *ProfiBieter* also provides how-explanations under the modified premises. We have designed hypothetical reasoning and will include it in the next release.

Although the student would benefit from why-not explanations, we will not implement them. Why-not explanations answer such questions as "why is this not an English auction". On one hand how-explanations illustrate the auction formats already, on the other hand more complicated why-not explanations for questions like "why is my bid not an optimal strategy" are very difficult to implement. The rules are designed for finding the optimal strategy and allocating one rational number to one problem. The reverse allocation to find a problem for a user specified number is only possible if the input fits one hypothetical result.

### *B. Timing of Feedback*

Research shows that the timing of feedback influences learning efficiency. For multiple choice problems, Dihoff et al. demonstrated that immediate feedback outperforms delayed or no feedback [10]. Generally, faster feedback is more efficient for learning [8].

Nevertheless, a business game like *ProfiBieter* has to consider the characteristics of game environments. Early feedback, which guides the student through the task, would turn the game into a tutorial. Such guidance can introduce the first solution steps, but should be abandoned afterwards.

*ProfiBieter*'s first response possibility is after the bidding. This feedback profits from an activated learner, who has dealt with the problem just before, but such an early response meets practical problems. There are multiple bid auctions where the player can submit multiple times. Giving feedback after each bid would be premature (as discussed above). To detect the last bid is impossible as the learner could always bid once more. These shortcomings of early feedbacks warrant our decision for the auction end as the appropriate feedback moment.

At the end of the auction the player is still interested because she wants to know the winner. With the whole auction history available *ProfiBieter* transparently shows all decisions and can deliver more useful feedback than during the auction. These are the reasons why *ProfiBieter* displays feedback only after the end of each auction.

Normally, business games are divided into a preparation, a game and a debriefing phase, which gives feedback at the end of the game. In contrast to *ProfiBieter*, which gives feedback after each auction, one hour games do not need to disturb the playing session because the end of the game comes fast enough to remember the student's decisions. Short games can be replayed many times.

[1] Dutch auctions are the counterpart of English auctions. Beginning from a very high price the auctioneer lowers the price until the first bidder accepts it [7].

### *C. Display*

Business games need a careful and often elaborate user interface design. The importance of the user interface separates business games from simulations [11]. Since the current version of *ProfiBieter* is a demonstration prototype, its user interface is frugal.

Internet-based business games need client software to communicate with. *ProfiBieter* uses a popular browser. A browser interface may be less flexible than a proprietary client interface, but it offers a ready-made standard solution. The following screenshots illustrate our approach:

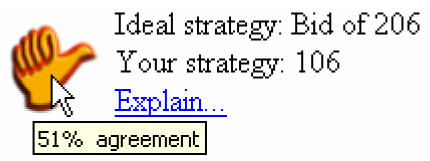


Figure 1. First feedback.

Fig. 1 (First feedback) shows the output of the result. Two simple sentences compare the player's bid with the ideal bid strategy. The hand symbol visualises the grading. The idea of the symbol is to motivate the player to dig deeper. With the 'explain…' link she can call the detailed explanations.

Instead of presenting the entire explanation tree, *ProfiBieter* only displays one node at a time, thereby offering the player to navigate through the entire tree. This protects the learner from information overload.

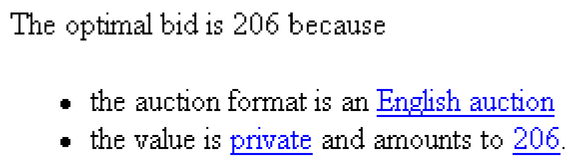


Figure 2. Elaborated feedback

As you will see in the next section, the text of fig. 2 (Elaborated feedback) is generated from the rule "if the auction is an English auction and the value is private then bid the private value". The sentence structure is the subordinating conjunction "because" connects the conclusion with the conditions. The learner can navigate down the explanation tree with the hypertext links and up with a navigation symbol not displayed here. If she calls an explanation of a fact, for example 'English auction', then the system highlights the appropriate passage of the problem text.

## III. Explanations

The last section argued the importance of feedback and introduced the way *ProfiBieter* expresses hierarchical feedback. The following section describes how facts are recorded and rules are explained.

### *A. Knowledge base*

A web authoring tool allows the teacher to modify or to add problem texts, which carry the factual knowledge. At runtime, the problem solving component uses the facts as input into the rule interpreter. When displaying explanations, underlying facts are highlighted within the appropriate problem text.

A knowledge base is represented in *RuleML*. **Rule** **M**arkup **L**anguage has been defined in XML [12]. Since a teacher is not able to understand *RuleML*, *ProfiBieter* offers an XML-based graphical user interface under *MS Visio*.

Fig. 3 (Simplified and rule representation in MS Visio) illustrates the graphical representation of a rule. The rule shown consists of the conditions A and B, and the implication C. An implication fires (executes) as soon as its conditions are true. C, for example, fires as soon as the fact list contains A and B. An identifier preceded by a question mark (e.g. value) denotes a variable.

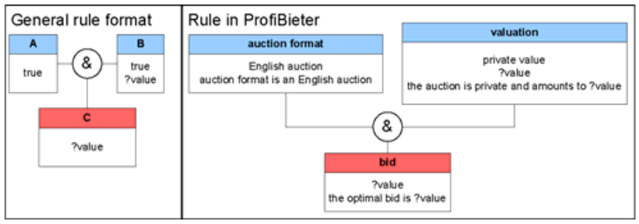


Figure 3. Simplified and rule representation in MS Visio

Section 2 (Feedback in Business Games) introduced a strategy for an 'English Private Value Auction'. The right part of fig. 3 displays the corresponding strategy rule. Compared to the general rule format the *ProfiBieter* rule adds output placeholders.

Placeholders allow the teacher to control the output format of explanations. For example, the second rule condition contains the placeholder *?value*. Instantiating the placeholder results in the following output: "the auction is private and amounts to 206" (cf. fig. 2).

### *B. Generating explanations*

*ProfiBieter* uses a forward chaining rule engine, the .NET Business Rule Engine [13], to draw conclusions from the rules and facts of the current problem. At runtime, the engine applies the facts of the problem to the knowledge base to see if any rule will fire. Thus, the solution path results from a traversal through the knowledge base which contains the facts and rules needed to solve the problems submitted.

The explanation tree is generated by tracing this solution path. The algorithm generates the explanation tree bottom-up by adding implication and conditions for each rule that fires. The last firing rule is the solution.

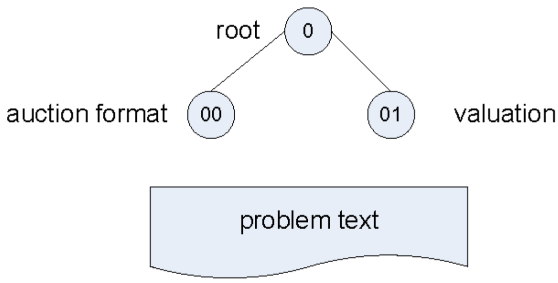


Figure 4. A short explanation tree

Fig. 4 (A short explanation tree) displays the trivial explanation tree for the simplest auction model, an 'English Private Value Auction'. Usually, there are several rules and the corresponding tree is deeper. If a user wants to know how a solution is calculated, the system jumps to the root and outputs its conditions. The hyperlinks in the condition texts allow the learner to navigate further. Since the Hypertext Transfer Protocol (http) is stateless, a number identifies each node. Otherwise, a user's goal node would get lost when calling another web page. A node without children is a fact leading to a marked text position in the problem text.

## IV. Conclusions

Generating automatic explanations improves the intelligibility of business game decisions. We show how elucidating the black boxes common in business games alleviates the player's frustration after a failure.

Automatic explanations not only help students to understand the outcome of their decisions but also support the teacher to understand the game model and to modify its parameters and problem texts.

But the approach has its limitations: Integrating an explanation component is costlier than developing a drill-and-practice game. In particular, an intelligible, for example a rule-based, knowledge representation is not suitable for some implementation approaches. Moreover, our implementation model excludes games without preset rules which are adjudicated by human referees. However, some of the popular games operate on defined rules and integrate nicely with the proposed design.

Finally, explanation components cannot address every question. Alternatives are for example blended learning and intelligent hypertext. Any of these alternatives is more efficient than simple internet-mediated business games like *Sim-Log* [14].

## References


[1] A.J. Faria, “The changing nature of simulation research: A brief Absel history”, *Business Simulation & Experiential Learning Developments*, pp. 84–90, 2000.

[2] B. Högsdal, *Planspiele: Einsatz von Planspielen in der Aus- und Weiterbildung*, ManagerSeminare Gerhard May Verlags GmbH, 1996.

[3] M. Dobson, V. Kyrylov and T. Kyrylova, “Training using Agent-based Business Strategy Games Decision”, *Computers and Advanced Technology in Education*, pp. 66–71, 2004.

[4] H. Mandl, C. Keller, M. Reiserer and B. Geier *Planspiele im Internet: Konzepte und Praxisbeispiele für den Einsatz in Aus- und Weiterbildung*, W. Bertelsmann Verlag, 2001.

[5] M. Lusti, “Protocol Driven Hypertext Explanations in a Tutor for Financial Analysis”, *Proceedings of the 7th World Conference on AI in Education*, August 1995.

[6] D. Gomboc, S. Solomon, M.G. Core, H.C. Lane and M. van Lent, “Design Recommendations to Support Automated Explanation and Tutoring”, *Conference on Behavior Representation in Modeling and Simulation*, 2005.

[7] V. Krishna, *Auction Theory*, Academic Press, 2002.

[8] B. Jacobs, *Aufgaben stellen und Feedback geben*, Medienzentrum der Philosophischen Fakultät der Universität Saarbrücken, retrieved from: http://feedback-jakobs.notlong.com, 21.11.2002.

[9] J.L. Teslow, “An evaluation of humor as a motivational, cognitive, and affective enhancement to lean feedback and remediation strategies in computer-based instruction”, *Annual Convention of the Association for Educational Communications and Technology*, 1996.

[10] R.E. Dihoff, G.M. Brosvic and M.L. Epstein, *The Role of Feedback during Academic Testing: The Delay Retention Effect Revisited*, The Psychological Record, 2003.

[11] N. Viknashvaran, K.W. Wong, C.C. Fung and A. Depickere, “Distinguishing simulation games from simulators by considering design characteristics”, *Proceedings of the second Australasian conference on Interactive entertainment*, pp. 141–144, 2005.

[12] RuleML, *The Rule Markup Initiative*, retrieved from http://www.ruleml.org/, 15.08.2006.

[13] D. Dossot, *.NET Business Rule Engine*, retrieved from http://www.agilepartner.net/oss/nxbre/, 15.08.2006.

[14] Universität Göttingen, *Unternehmenssimulation SIM-LOG: Simulation zur Logistik, Institut für betriebswirtschaftliche Produktions- und Investitionsforschung*, retrieved from http://sim-log.notlong.com, 15.08.2006.


## Authors


**M. Fischer** is with the Department of Information Systems, Faculty of Economics, University of Basel, Switzerland (e-mail: martin.fischer@unibas.ch).

**M. Lusti** is with the Department of Information Systems, Faculty of Economics, University of Basel, Switzerland (e-mail: markus.lusti@unibas.ch).


Manuscript submitted 14 March 2007.