The King of Reasonshire – a computer game to practice critical thinking

Walter Carnielli
State University of Campinas-
CLE and IFCH
Campinas (SP) BRAZIL
13083-859
walter.carnielli@cle.unicamp.br

Juliana Bueno-Soler
Federal University of ABC-
CCNH
Santo André (SP) BRAZIL
09210-170
juliana.bueno@ufabc.edu.br

Flavio S. Correa da Silva
University of Sao Paulo- IME
Sao Paulo (SP) BRAZIL
05508-090
fcs@ime.usp.br

ABSTRACT

Critical thinking refers to the improvement of reasoning abilities, based on techniques grounded on formal logics. The development of critical thinking skills is an important tool to build participative societies in which citizens make fair and well grounded decisions guided by tenets of rationality. Critical thinking does not refer only to the construction of techniques for well grounded reasoning. It also requires the development of individual skills, through practice, so that individuals can employ the appropriate techniques for the analysis and construction of arguments with naturality and efficiency. In the present work we introduce a computer game for the development of such skills.

Keywords

Serious games, critical thinking, edutainment.

1. INTRODUCTION

Critical thinking refers to the improvement of reasoning abilities, based on techniques grounded on formal logics. Critical thinking is an important tool to build participative societies, in which citizens make fair and well grounded decisions guided by the tenets of rationality.

The techniques for critical thinking have been studied and developed broadly (see e.g. [4, 1]). In order for critical thinking to be truly useful, however, the *practice* of the corresponding techniques must be developed, in such way that they can be incorporated and become natural and intuitive for individuals.

We can find in the Web some pedagogical resources to support the teaching and the practice of critical thinking[1]. However, these resources can be challenging for the passer-by user, as they require significant effort and skills development to go beyond the simplest patterns of reasoning for the analysis and construction of well grounded arguments.

Serious games have been successfully used for learning in a variety of domains [6]. Serious games refer to games (of any genre – board games, action games, computer games etc.) which are designed and built for a specific purpose – e.g. for learning, simulation or training. In the present work we introduce *The King of Reasonshire*, which is a computer game for the development of critical thinking skills.

In *The King of Reasonshire*, a series of scenarios is presented to the user, in which the evaluation of featured situations grows progressively in complexity. The user is invited to formalize each scenario symbolically, as a collection of logical sentences, and then to identify which sentences are missing, irrelevant or faulty. User interventions are scored with positive or negative points, depending on whether the user has added correct or incorrect sentences and/or evaluations about the relevance and correctness of sentences. The goal is to score the maximum number of points.

No clues are provided to the user. Instead of clues, the user is directed to appropriate literature (e.g. [4, 1]) for self-study, in order to improve his/her skills. This way, we intend to simulate real-life situations, in which individuals need to evaluate the quality of arguments and reasoning "on the fly" and with no expert advice.

The King of Reasonshire is actually work in progress. In the present article, we present a mock-up for the actual game, which is at the moment under development. It shall be implemented using the *JamSession* platform for knowledge-based interaction protocols [2].

In section 2 we detail our approach to critical thinking. In section 3 we introduce in detail our mock-up for *The King of Reasonshire*. In section 4 we briefly review the *JamSession* platform and advance how *The King of Reasonshire* shall be implemented. Finally, in section 5 we present some discussion and planned future work.

2. CRITICAL THINKING

This game is thought as an informal but solid introduction to argumentation and critical thinking, highlighting its tight relationship with logic. We follow [1], [3] and [4] as a guide for a solid and useful theoretical background to the art of argumentation. The game emphasizes the notion of argument

[1] See e.g. http://philosophy.hku.hk/think/

as composed by claims, viewed as declarative sentences that are either true or false, but not both.

An argument is *valid* if it is impossible for the premises to be true and the conclusion to be false (at the same time). An important notion to be learned is the notions of *good argument*: an argument is good if its premises give good reason to believe the conclusion is true.

Arguments that are not valid, or that we do no know whether are valid or not, are classified on a scale from *strong* to *weak*. An argument is strong if it is very unlikely that the premises could be true and conclusion false (at the same time). An argument is weak if it is not unlikely that the premises could be true and conclusion false (at the same time).

For an argument to give us good reason to believe its conclusion, its premises should be more plausible than its conclusion. Any argument that uses a premise that is not more plausible than the conclusion is said to *beg the question*.

The necessary conditions for an argument to be good include:

- The premises are plausible.
- The argument is valid or strong.
- The argument does not beg the question.

Implicit in the games rules are also The Principle of Rational Discussion, the notion that arguments in the real life should be repaired (a guide to repairing arguments), as well as the notion that certain arguments are unrepairable (fallacies). Elements of the the theory of reasoning about experience by means of analogies, generalizations, numbers and statistics, cause and effect are also provided in the *The King of Reasonshire*.

3. THE KING OF REASONSHIRE

The King of Reasonshire is a computer game designed to support the development of critical thinking skills. It contains a collection of scenarios, which are presented to the user for analysis and correction in case faulty reasoning patterns are identified.

The game plot is as follows:

In the kingdom of Reasonshire there lived a wise king, who opened his castle once a year to know about his people. In that day, people from all corners of Reasonshire went to the king, to present their quarrels and issues related to land dispute, commercial quandaries, love affairs, and the like. The king of Reasonshire needed, then, to advise his people based on fair and wise judgement. In order to produce better judgements, the king used to ask his wise counsellor for advice.

In this game, the user takes the role of the king's counsellor.

The user shall be asked to evaluate each situation that is presented to him/her. Based on the user's analysis, certain conclusions and measures can be taken:

- The user may conclude that the arguments that have been presented to him/her are faulty: relevant information is missing, irrelevant information is provided – which can cause confusion and misguidance for decisions – or inadequate reasoning patterns are being used. In this case, the user should explain to the king why no decision should be proclaimed.
- The user may conclude that the arguments that have been presented to him/her are indeed coherent and complete, and that therefore he/she is prepared to suggest a decision to the king. In this case, the user should explain to the king his/her line of reasoning and what decision is advised.

Once the user starts the game, short stories are presented to him/her, making use of carefully crafted animations. After the presentation of a story, a short text featuring the main elements of the plot is presented to the user. Specific phrases within the text can be associated to logical symbols, such as constants, variables and predicates. These phrases are underlined, to simplify the analysis for the user as well as to standardize the interaction between the user and the game.

After connecting phrases with logical symbols, the user can assemble logical sentences, which should comprise a logical model for the text that has been presented to him/her. Based on this logical model, the user can then employ standard patterns of inference (of the form of `if-then` rules) and build conclusions and judgements. The user can also identify missing information that would be required in order to build proper deductions which could lead to well founded conclusions and judgements.

Correct logical analyses and deductions give positive points to the user, and mistakes give negative points. The goal, of course, is to accumulate the maximum number of points from the analysis of a collection of situations. The difficulty of each situation presented for analysis grows progressively as the user advances on the game.

No clues or hints are provided to the user, beyond the identification of phrases that can be turned into logical symbols. This way, the user has to face situations in similar fashion to what may occur in everyday life. Instead of getting hints, the user is advised to check the appropriate literature to improve his/her scores in *The King of Reasonshire*. As the game progresses, feedback is provided to the user through the facial expressions of the king and the citizens of Reasonshire, who become happy if fair, wise and well grounded decisions are taken, suspicious if decisions are not carefully grounded and explained, and sad or angry if bad decisions are advised.

This game is work in progress. We are, at the moment, starting to work on the implementation of the game. Once implemented, we shall develop empirical evaluation of the effectiveness of the game, based on experimentation with groups of users. Our target audience are students and professionals whose routine activities require decision making. We expect to provide the means for members of this audience to improve their performance in judgement and decision making in everyday activities, such as buying goods and trusting information provided by the general media.

The game consists, essentially, of dialogical interactions between the user and a software agent who manages the situations, the user's score and the feedback provided to the user through facial expressions of characters in the game. The implementation of *The King of Reasonshire* shall be founded on the *JamSession* platform. *JamSession* is based on a simple, executable process algebra, through which interaction protocols can be specified and executed. Interaction protocols in *JamSession* are used to specify scripts, which are rules of engagement which regulate the interactions between the user and the game.

In the next section we describe briefly the main components of *JamSession* and how they shall be used to implement *The King of Reasonshire.*

4. THE JAMSESSION PLATFORM

JamSession is a platform to specify and execute knowledge-based interaction protocols, which are scripts that characterize how groups of agents interact in order to reach their (individual as well as collective) goals. We shall use *JamSession* to implement *The King of Reasonshire.*

A fundamental notion in *JamSession* is the concept of *location.* Intuitively, we have agents in an environment, whose capabilities are blocked or released for use, depending on the the locations where they are situated. When a user needs a specific service, he/she must make sure that the agent that has the capability of furnishing that service in a specific location has actually moved to the appropriate location. Locations are, therefore, abstractions of groups of services, whose accessibility is controlled by how users transport named agents to/from specified locations.

In our case, services (i.e. the capabilities of agents situated in specified locations) characterize states in the game, and determine what actions shall be available at each step during the gameplay, e.g. retrieval and presentation of a new situation, interaction with the user to capture the evaluation and decisions related to a situation, score assessment of the user based on his/her decisions and evaluations, as well as control of the game environment (start/pause/end game, etc.). *JamSession* behaves as a mediator between user requests and the required computational resources that implement the game presentation, including interactive graphics and user input-output.

Formally, we have a directed graph to specify locations and their connections. The nodes of the graph are the locations, and the arrows characterise the admissible transitions that agents can perform to move about locations. *JamSession* is a coordinator of resources, which are represented as capabilities of situated agents. An agent stays in a location until it receives an order to move to a different location.

An order for an agent to move is a triple of the form[2] $move(Agent, Location_1, Location_2)$.

In the order above, the agent $Agent$ is assumed to be in $Location_1$ and is being requested to move to $Location_2$. An order to move can be evaluated, in which case an attempt to execute it shall be performed and a corresponding truth value shall be assigned to it, depending on the success of the execution. If the agent $Agent$ is indeed in $Location_1$ and there is a path from $Location_1$ to $Location_2$, then $Agent$ is moved from $Location_1$ to $Location_2$, and the order is evaluated to $\top$. Otherwise, $Agent$ stays wherever it is and the order is evaluated to $\bot$.

The capabilities of situated agents are represented as first-order *predicates* in *JamSession.* Each predicate is associated to a pair $[Agent, Location]$. Predicates also have *Input* and *Output* parameters, which are formed respectively as first-order terms and free variables. Hence, a predicate has the form $[Ag, Loc]predicate((ITerm_1, ..., ITerm_n), (OVar_1, ..., OVar_n))$ where Ag is an agent, Loc is a location, $ITerm_i$ are input terms and $OVar_j$ are output variables.

Predicates are defined during the design of a system, specifying what resources can be triggered by what agents in which locations. During the execution of a system, predicates are used to actually trigger resources.

A predicate can be triggered, i.e. there can be an attempt to evaluate it, at any time. Most typically, predicates are used as attempts to activate system resources. The predicate input terms are syntactically and semantically verified, and it is also checked whether the agent Ag is located in Loc. If all verifications are successful, the corresponding resource is activated, possibly instantiating the output variables, and the predicate is evaluated either to $\top$ or to $\bot$, depending on its programmed behaviour. If Ag is not located in Loc, then the predicate is blocked and the corresponding resource cannot be activated. In this case, the predicate is evaluated to $\bot$ and the output variables are returned uninstantiated.

Agents can communicate through asynchronous messages. A message has the form $[Ag, Loc, (rwc)]message(Template)$ where Ag is an agent, Loc is a location, (rwc) are three bits stating whether the agent Ag in location Loc is granted the power to read, write and consume messages of a given type, *message* is the message name and $Template$ is a message template (e.g. given as an XML structure). During the design of a system, the three bits can be specified to be on or off, to determine how an agent is allowed to use a message. During the execution of a system, one and only one bit must be on at each time: when the *write* bit is on, the agent shall queue a message on a globally accessible message gateway; when the *read* bit is on, the agent shall read a message from the queue and the message shall remain available for other agents to read it; when the *consume* bit is on, the agent shall read the message and remove it from the queue. If the agent is located at the appropriate location and is asked to perform the messaging action that conforms to its power, as determined during the design of the intelligent interactive environment, then the corresponding action is performed and the message request is evaluated to $\top$. Otherwise, it is evaluated to $\bot$.

Asynchronous messages are a powerful tool for the coordination of resources and for the interoperability between

[2] We employ, in the presentation of *JamSession*, the PROLOG convention for terms and variables. Hence, terms starting with capital letters are free variables.

heterogeneous services [5]. Movement of agents are a powerful tool for the control of resources, through which security and reliability can be designed and implemented. Predicates can be as powerful as needed, and their power can be enhanced by the coordination and parameter passing provided by messages and movements.

The resource we have in *JamSession* to combine all these entities is the construction of *knowledge-based interaction protocols*. A knowledge-based interaction protocol is a structure of entities that specifies their order of evaluation. Entities can be of four types: (1) orders for agents to move; (2) orders for agents to send/read/consume messages; (3) predicates; and (4) auxiliary knowledge-based interaction protocols. Entities can also be combined by means of connectives, as explained below.

Knowledge-based interaction protocols are linked to locations. A request to trigger a knowledge-based interaction protocol can result in the following alternative situations:

- The requested knowledge-based interaction protocol is not actually defined for the specified location. In this case, the obtained truth value is $\bot$.
- The requested knowledge-based interaction protocol is defined for the specified location. In this case, the specification of the knowledge-based interaction protocol is retrieved and evaluated, based on the algebraic rules that govern the behaviour of the connectives that are used in the specification of the knowledge-based interaction protocol. The result of the evaluation determines the truth value that shall be assigned to the knowledge-based interaction protocol, which can be $\top$ or $\bot$.

A knowledge-based interaction protocol is denoted as $[Loc]$ $kbip((ITerm_1, ..., Term_m), (OVar_1, ..., OVar_n))$. Loc stands for the location to which the knowledge-based interaction protocol is connected, $ITerm_i$ are first order input terms and $OVar_j$ are output variables as before. The expected utilization of terms and variables in the specification of knowledge-based interaction protocols is for parameter passing accross predicates and auxiliary knowledge-based interaction protocols.

A knowledge-based interaction protocol takes the form of a formula in disjunctive normal form, in which atoms are the four entities in the list above (namely, orders for agents to move, orders for agents to process messages, predicates and auxiliary knowledge-based interaction protocols). More formally, the specification of a knowledge-based interaction protocol takes the form $[Loc]kbip((ITerm_1, ..., ITerm_m), (OVar_1, ..., OVar_n)) ::= \bigvee_1^k F_i, F_i = \bigwedge_1^{r_i} e_j, e_j = Move|Message|Predicate|Kbip, Move ::= move(Agent, Location_1, Location_2), Message ::= [Ag, Loc, (rwc)]message(Template), Predicate ::= [Ag, Loc]predicate((ITerm_1, ..., ITerm_n), (OVar_1, ..., OVar_n)), Kbip ::= [Loc]kbip((ITerm'_1, ..., ITerm'_{m'}), (OVar'_1, ..., OVar'_{n'}))$.

Both disjunction and the conjunctions are assumed to be non-commutative, as a means to simplify the computational task of evaluating the truth-value of a knowledge-based interaction protocol[3].

Knowledge-based interaction protocols are used to specify, implement and execute interactions among users of an intelligent interactive environment, and between users and the environment itself. Agents, locations and all entities that comprise the knowledge-based interaction protocols are the conceptual resources used to characterise the desired interactions.

Knowledge-based interaction protocols can also be triggered concurrently. This feature improves significantly the expressive power of *JamSession*. An important feature of *JamSession* is that knowledge-based interaction protocols can be formally verified with respect to desired properties. In future publications we shall detail how this can be done.

5. DISCUSSION AND FUTURE WORK

Most daily-life arguments are non-deductive, as well as a good part of scientific, philosophical and juridical argumentation. Further improvements in the *The King of Reasonshire* would include finer distinctions between deductive arguments and non-deductive arguments, and more weight towards evaluating the strength of inductive and abductive arguments. Specific modulus such as critical thinking in economics and legal (juridical) reasoning are planned as future work.

6. REFERENCES

[1] W. Carnielli and R. L. Epstein. *Pensamento Critico: o Poder da Lógica e da Argumentação (2nd. ed.)*. Rideel, São Paulo, 2010.

[2] F. S. Correa da Silva. Knowledge-based interaction protocols for intelligent interactive environments. *IME technical report (electronic version available at http://lidet.wordpress.com/)*.

[3] R. L. Epstein. Arguments and explanations. *Bulletin of Advanced Reasoning and Knowledge*, 1(1):3–17, 2001.

[4] R. L. Epstein and C. Kernberger. *Critical Thinking (3rd. ed.)*. Wadsworth, 2005.

[5] G. Hohpe and B. Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley, 2003.

[6] D. Michael and S. Chen. *Serious Games: Games That Educate, Train, and Inform*. Course Technology, 2005.

[3] Again, we employ this evaluation strategy based on the PROLOG convention.