Test your Strategy – Intuitive Strategy Definition and Evaluation for Novices and Experts

Sven Jörges
TU Dortmund
sven.joerges@tu-dortmund.de

Marco Bakera, Tiziana Margaria
University of Potsdam
{bakera,margaria}@cs.uni-potsdam.de

Christoph Lattemann
University of Potsdam
christoph.lattemann@uni-potsdam.de

Abstract: *ConnectIT* uses a graphical representation to express strategies for playing the connect-four game. With this tool we can transfer complex knowledge about the connect-four game itself, simple algorithmic concepts and modern high-level component-based modeling techniques from theoretical grounds into an inspiring and practical learning experience. We proved the adequacy of this approach in several projects with school pupils who had no experience in computer science. Solely based on intuition and knowledge about the game, they were able to build really competitive strategies. *ConnectIT*'s intuitive approach to strategy formulation and enactment could be an asset when applied to serious games, e.g. in business and management.

1 Introduction

Developing, managing, evaluating and comparing strategies for problem solving are central skills when handling demanding and complex problems, both in computer science and economy. Such strategies often exist in the minds of single individuals, and are often based on intuition, experience and personal knowledge. Especially when working in teams, it is a big challenge to find common means for clearly formulating, communicating and evaluating strategies. Games form a very common domain for examining strategies. Due to their entertaining nature, they are easily and intuitively accessible for everyone, and thus are subject to a multitude of research.

We present *ConnectIT* [BJM09], a tool that uses an intuitive yet powerful representation of strategies for the well-known connect-four game in terms of service orchestrations. Technically, *ConnectIT* is based on jABC [SMN$^+$06], a framework for model driven development of service-oriented applications that embodies the eXtreme Model Driven Development paradigm (XMDD [MS08]). Applied to strategies, this means that simple, yet immediately functional and non-trivial strategies can easily be created from a taxonomy of predefined domain-specific building blocks. In this context, the building blocks concern the connect-four scenario, but other domains in use include scientific workflows in bio-informatics, business processes, text analyses in computer linguistics, and several applications in industrial projects.

Once a strategy is created, having a measure at hand that allows for both concise and easy evaluation and comparison of several strategies offers two benefits. First, a fast validation technique of the overall outcome reveals errors at an early development stage. Second, the learning curve to become comfortable with the strategy under development is greatly improved - since the strategies' behavior and impact usually depend on highly context-sensitive fine-tuning of parameters. Accordingly, *ConnectIT* provides an integrated evaluation functionality of the strategies and of their parametrizations in terms of tournaments. The effects can be studied in depth by playing several different kinds of tournaments (everyone against everyone, league mode or knock-out tournament) with (virtual) users that adopt the one or the other strategy. Each tournament returns a graphical presentation of the results, and details about each match in the tournament. Combining several evolution steps in strategy development into a tournament enables students to directly witness and quickly pinpoint the effects of the decisions taken, and to improve their strategy. This capability directly reveals whether a supposed enhancement of the strategy also yields a significant improvement of the strategy's competitiveness.

We used and refined *ConnectIT* for several years in the do-camp-ing project [Tec08] and in one-day workshops ("Schnuppertage")[1], with heterogeneous groups of pupils who had no previous computer science training. Solely leveraging their intuition and knowledge about the connect-four game, the pupils were able to build really competitive strategies.

Connect-four In this game, first studied by Allis [V. 88] with a knowledge-based approach, two players throw in turn a token in a board of seven columns and six rows. The token falls down to the lowest unoccupied position in the column. Each player plays with its own token color. The player that gets a group of four connected tokens (either horizontally, vertically or diagonally) wins the game. When the board is full and no player has won yet, the game is drawn.

2 The jABC Framework

To model strategies in an intuitive graphical way, we used the *jABC* , an environment that enables service-oriented development along the lines of the Extreme Model-Driven Development (XMDD) paradigm [MS08]. XMDD is designed to continuously involve the customer/application expert throughout the whole system's life cycle. It is extreme in that, like extreme programming, it has short cycles based on user feedback, but instead of working at the code level, it applies this customer-near process at a graphical modelling level.

In jABC, users develop service-oriented systems in a behavioral, process-driven way, by composing reusable building blocks into flow graph-like structures called *Service Logic Graphs* (SLG). The building blocks are called *Service Independent Building Blocks* (SIBs).

In *ConnectIT*, we use jABC to build strategies for the connect-four game. The cus-

[1] `http://connectit.cs.tu-dortmund.de/blog/`

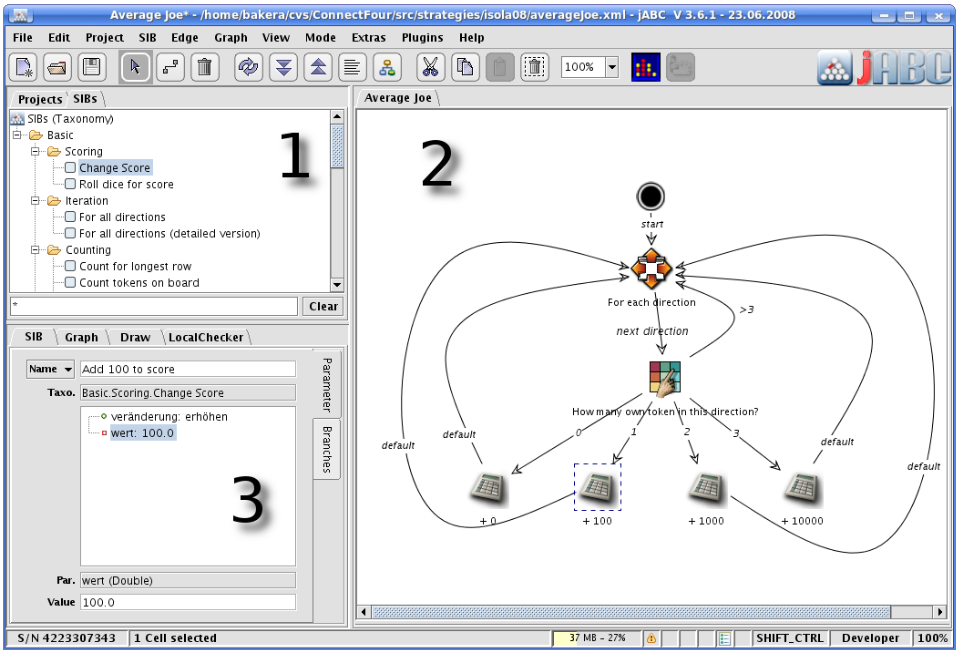

Figure 1: Screenshot of the jABC GUI, and the strategy "Average Joe" on the right

tomer/application expert mentioned above is typically someone who at least has an idea about how to play the game, and who transfers this knowledge to a graphical representation (an SLG).

Fig. 1 shows a screenshot of jABC's user interface, which consists of three main elements (indicated by the numbers):

1. The *project and SIB explorers* enable users to browse available jABC projects and the domin-specific library of building blocks available for modelling. Here, the SIB `Change Score` is selected and instantiated by dragging it to the canvas.
2. The *graph canvas* is used for graphically composing SLGs/strategies.
3. The *inspectors* provide detailed information about selected SIBs and may be used by jABC plugins to add further functionality.

SIBs, the building blocks used in jABC, represent atomic, executable service/behavioral units. A SIB's *formal parameters* allow customizing its behavior depending on the current context of use. A SIB's outgoing *branches* direct the further flow of execution depending on the results of the current building block's execution, i.e. they denote its possible "exits".

The SIB `+100`, highlighted in the canvas (2) of Fig. 1, is an instance of the building block `Change Score`, which is found in the SIB taxonomy (1). `Change Score` handles a score attached to each column of the connect-four game board (cf. Sect. 3). As shown in the SIB inspector (Fig. 1 (3)) `Change Score` takes two parameters:

- *modification* ("veränderung"), determining how the score should be modified. Possible values are: increase, decrease, set, and multiply, and
- *value* ("wert"), the numeric value used to modify the score.

The SIB's *default* branch is taken if the score was modified successfully, and the *error* branch if the modification did not succeed, e.g. if no score could be found at the current column.

Technically, *ConnectIT* is realized as a jABC plugin. Plugins augment jABC's functionality and provide features like animation, rapid prototyping, formal verification, debugging, monitoring, and evolution.

We now describe the tool itself, and how non-experts can use jABC as a modelling environment for building connect-four strategies.

3 The ConnectIT Tool

Once strategies are modelled in the jABC, the *ConnectIT* tool[2] supports playing, measuring and comparing those strategies. As usual in domain specific languages, the tool comes with a set of SIBs (the primitives of this language), that are useful to model the strategies (expressed as SLGs). For instance, we provide building blocks for checking whether the board is empty, whether the current player is the start player, or who will win at the next turn. Furthermore, *ConnectIT* provides a graphical interface for playing the game against another human or a strategy, and for arranging tournaments among multiple strategies.

When playing, a strategy is applied to each of the board's columns. By analyzing and evaluating the current situation on the board, a strategy rates the current column with a score. Compound strategies add to a global score the scores of their sub-strategies. The token is then thrown in the column with the highest score. If several columns have the top score then *ConnectIT* chooses non-deterministically.

For instance, the simple strategy *Average Joe*, depicted in Fig. 2 (left), is basically purely offensive strategy that counts the number of tokens of the same color. The more of such tokens the strategy finds in one direction, the more points will be added to the final score of the column.

Fig. 2 (right) is a snapshot of a game of a human player (yellow/light gray tokens) against *Average Joe* (red/dark gray tokens). In the last move, *Average Joe* moved into the fifth column (black border around this token). The score for each column is indicated on the top: The 1100 points of column 5 result from the south-west row with two tokens (1000 points) plus the single token (100 points) on the left.

[2]`http://connectit.cs.tu-dortmund.de`

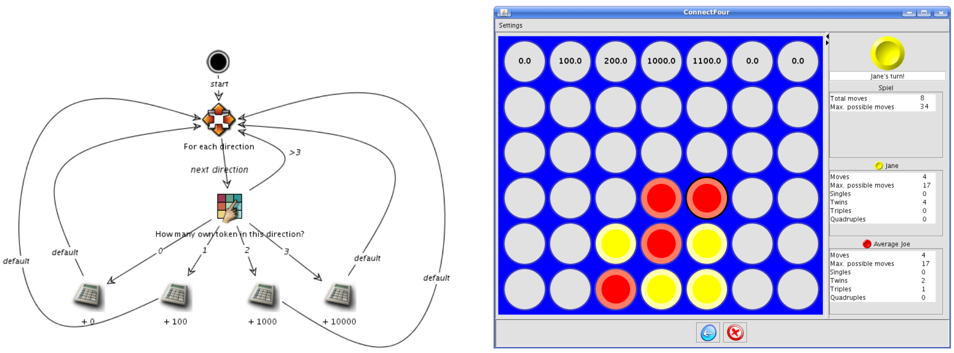

Figure 2: Left: Strategy *Average Joe* checks for connected tokens of his own color and prefers the column with most such connected areas of interest. Right: Sample play of a human player against *Average Joe*. At the column top: strategy's score for that column.

For more example strategies please refer to [BJM09].

4 Case Study: An Expert Strategy from Novices

Even novices can easily outperform *Average Joe*. "do-camp-ing" [Tec08] is an initiative of the TU Dortmund that offer pupils aged 15 to 18 to participate in one-week projects at the university, in teams of 8 to 12 participants with hardly any previous contact with the corresponding science. Representing computer science, since 2006 we successfully offered *ConnectIT* as a "do-camp-ing" project.

We usually start the project with"manual" connect-four games: the pupils just play the game by hand and write down patterns and rules of thumb they observe. After a short introduction in the use of the *ConnectIT* tool, they form teams and build their own strategy using the SIB library (see Sect. 3). At regular intervals, we stage tournaments (see Sect. 5) to compare and evaluate the strategies. If they need hints or new ideas, we show them other strategies like *Average Joe*, built by own peers in the previous years.

Note that the pupils build their strategies solely on the basis of their intuition. We do not provide them with any details about game-trees, α-β-pruning or related common algorithmic techniques for solving similar games. They are told to formalize their internalized experience with the game.

Fig. 3 shows a result from "do-camp-ing 2008". The *Chaos AI* strategy (name chosen by the two pupils who developed it) is very competitive. Again it contains the *Average Joe* component, but they also added a defensive part which considers the opponent's tokens. Furthermore, *sudden death* initially checks whether a throw in the current column leads to an immediate win. If so, it scores the column appropriately.

Whenever there are no tokens on the board, the strategy *prefers the middle column*, a direct implementation of a rule of thumb (like the favored center position in the tic-tac-toe game)

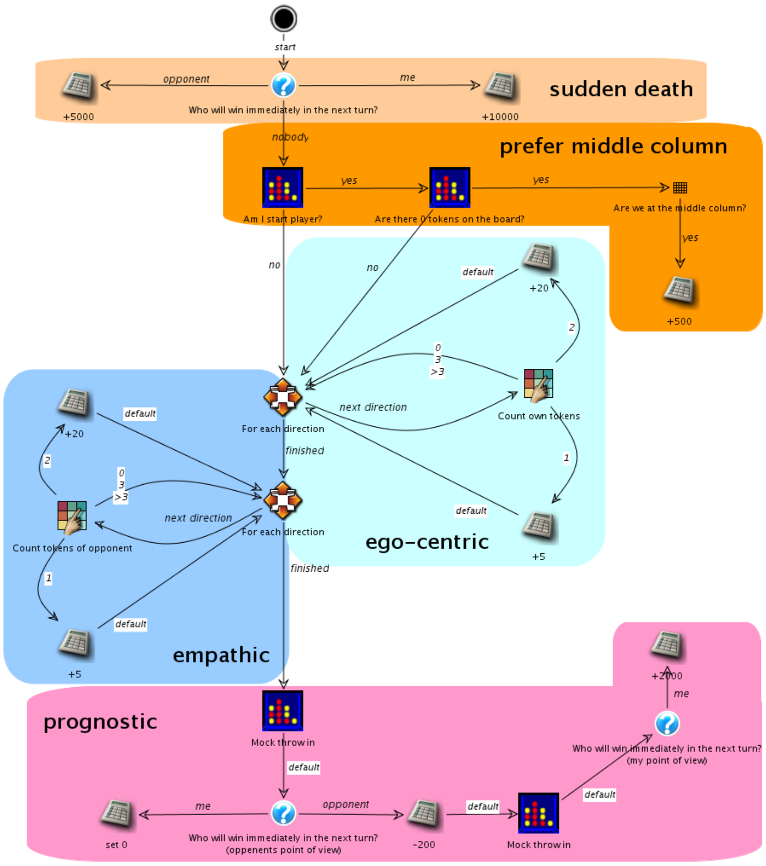

Figure 3: Strategy *Chaos AI* extends *Empathic Joe* by adding several tactics to decision-making.

that emerges as a best practice when playing.

After counting the own and opponent's tokens on the board, the *prognostic* part of the strategy explores the near future via a mock throw-in at the current column. The mock throw-in building block exchanges the roles of player and opponent, thus all subsequent building blocks must be treated from the opponent's point of view (until another mock throw-in appears, as in our case). After the first mock throw-in, the strategy checks whether the opponent would win. In this case, one would help the opponent, so the score for the current column is 0. The other case is first punished with a moderate malus (-200 points), as one would build a too obvious triple that is likely to be blocked by the opponent. Second, another mock throw-in follows in this case (and the roles are switched again). If one would still win in this situation, the column gets a high score, as this indicates a catch-22.

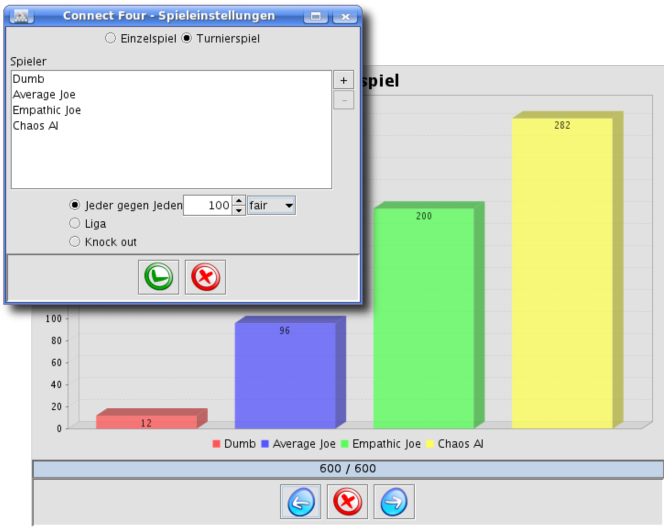

Figure 4: *Top left:* Configuring a tournament with four strategies that play each against all, every match to be repeated 100 times. *Bottom right:* The results of the tournament.

5 Evaluation and Comparison of Strategies

Having a measure at hand that allows for both concise and easy evaluation and comparison of several strategies offers two benefits. First, a fast validation technique of the overall outcome reveals errors at an early development stage. Second, the learning curve to become comfortable with the strategy under development is greatly improved - since the strategies' behavior and impact usually depend on highly context-sensitive fine-tuning of parameters. For instance, once opting for a market-driven pricing strategy, and once observing the prices of similar goods on the market, we then also need to decide to which extent they affect our own pricing decision. This certainly includes other endogenous aspects like costs and profits. In connect-four, considering the adversary's possible moves additionally to the own ones also requires deciding which (relative) weight this information carries in the decision of the next move. Accordingly, ConnectIT provides an integrated evaluation functionality of the strategies and of their parameterizations, in terms of tournaments. The effects can be studied in depth by playing several different kinds of tournaments (everyone against everyone, league mode or knock-out tournament) with (virtual) users that adopt the one or the other strategy. The dialog in Fig. 4 (top left) shows how tournaments can be set up.

Everyone against everyone denotes an each against all game, and one can configure how many times each pairing should be repeated. In *league mode*, everyone plays against everyone twice, and the winners are determined via scores: a win is 3 points, a draw is 1 point and a defeat is zero points. The *knock-out tournament* uses random pairings. The winner of each pairing enters the next round, and this is repeated until the final winner is determined.

The results are shown graphically (Fig. 4, bottom right) with details about each match in the tournament. Combining several evolution steps in strategy development into a tournament enables students to directly witness and quickly pinpoint the effects of the decisions taken, and to improve their strategy. This capability directly reveals whether a supposed enhancement of the strategy also yields a significant improvement of the strategy's competitiveness.

The following section describes the potential of *ConnectIT* when applied to the more general setting of "serious games".

6 Serious Games: The Potential for Business

The development of strategies and strategic behavior is most relevant to entrepreneurs or management teams that have a clear vision and mission for their business and are in the process of developing primary strategies. The development of a suite of strategies is an iterative process and involves circular thinking on the basis that optimal strategies will evolve gradually and be very interdependent. Accordingly, the best way to test strategies is to define, model and test them. By applying *ConnectIT* to simulate strategies, entrepreneurs and managers can easily evaluate their strengths and weaknesses in strategic planning and behavior. Additionally under virtual conditions, the learning process for the managers and entrepreneurs is risk-free and costless. Furthermore, the results of a chosen strategy can be analyzed systematically and improved over time.

A venture is most prone to failure during its first three or so years of operation - the so-called "valley of death" [ME03]. A key to getting through these early years is to avoid the obvious mistakes. Generally speaking, businesses fail for significant and substantial reasons which are often very evident to outsiders. Insiders often fail to see them because of their closeness, determination and so on.

These issues are reflected by the project *Science in 3D* [LS08]. This project from Potsdam University employed a virtual world environment, in particular Second Life, as an innovative tool for knowledge transfer for train entrepreneurs in the sector of biosciences. Virtual Worlds enable, thanks to their 3D environment, new ways of communication, collaboration and cooperation. The level of interactivity of Virtual Worlds is higher compared to other web applications which are based on plain text, voice or 2D presentations. Users in Virtual Worlds are represented by 3D avatars [FL08].

In this 3D training camp entrepreneurs have to go through the whole cycle of a venture including selling products to customers. As Second Life has a quasi-real economy with an own currency and about 100.000 visitors daily who may act as customers, entrepreneurs could easily test their venture skills. One part of this educational setting is the phase of "managing, growing, and ending the new venture". This phase includes the development of entrepreneurial strategy in the field of generating and exploiting new entries, growth strategies, and strategies for accessing resources and so forth [HPS05]. The project will provide simulations the development of these kinds of strategies by implementing *ConnectIT* on a 3D basis in the virtual world.

7 Conclusion

ConnectIT proved to be a versatile learning platform that targets a multitude of different users. Very young pupils can use it to learn and play the game itself, whereas for older pupils (10 years and older), the construction of simple strategies shapes their algorithmic thinking, as verified in the "do-camp-ing project" (cf. Sect. 4).

Computer scientists experience an intuitive and low-threshold way to gain first hands-on experience with model-driven and component-based development technologies. These technologies are essential cornerstones of the XMDD paradigm underlying modern service-oriented development, and the jABC platform on which *ConnectIT* is built.

For the general public, *ConnectIT* conveys an intuitive way of approaching strategic thinking and the construction of concrete strategies, which are key to numerous application fields. In business economics, strategies are omnipresent, e.g. in any kind of negotiation, from purchases to mergers and acquisitions, in organizational matters, and for market dynamics, which all require strategic thinking, and tactical and at the same time flexible acting, based on experience.

Acknowledgments We thank all participants of do-camp-ing in 2007 and 2008, especially Jonathan Dölle and Alexandros Zinelis who developed *Chaos AI*.

References

[BJM09] M. Bakera, S. Jörges, and T. Margaria. Test your Strategy: Graphical Construction of Strategies for Connect-Four. In *Proc. ICECCS*, 2009. To appear.

[FL08] M. Fetscherin and C. Lattemann. Editorial: Virtual Worlds Research: A Conceptual Primer. volume 9, pages 192–194, Sept. 2008.

[HPS05] R.D. Hisrich, M.P. Peters, and D.A. Shepard. *Entrepreneurship*. McGraw-Hill, 6th edition, 2005.

[LS08] C. Lattemann and S. Stieglitz. Einsatz Virtueller Welten in der Aus- und Weiterbildung - Das Projekt Bio-VWe. In *GENEME Gesellschaft für neue Medien*, 2008.

[ME03] L.M. Murphy and P.L. Edwards. Bridging the Valley of Death: Transitioning from Public to Private Sector Financing. *National Renewable Energy Laboratory*, NREL/MP-720-34036, 2003.

[MS08] T. Margaria and B. Steffen. Agile IT: Thinking in User-Centric Models. In *Proc. ISoLA 2008*, CCIS N.17, pages 493–505. Springer, 2008.

[SMN$^+$06] B. Steffen, T. Margaria, R. Nagel, S. Jörges, and C. Kubczak. Model-Driven Development with the jABC. In *HVC - IBM Haifa Verification Conference*, LNCS N.4383, pages 92–108. Springer Verlag, October 2006.

[Tec08] Technische Universität Dortmund. do-camp-ing Website, 2008. `http://www.do-camp-ing.de/` (in german).

[V. 88] V. Allis. A Knowledge-based Approach of Connect-Four. Master's thesis, 1988.