

Serious Gaming for Predictive Analytics

Roderick M. Riensche, Patrick R. Paulson, Gary R. Danielson,
Stephen D. Unwin, R. Scott Butner, Sarah M. Miller, Lyndsey R. Franklin, Nino Zuljevic

Pacific Northwest National Laboratory
902 Battelle Boulevard, Richland, Washington 99352
{ rmr | patrick.paulson | gary.danielson | stephen.unwin | scott.butner | lyndsey.franklin | nino.zuljevic }@pnl.gov

Abstract

We describe a methodology and architecture to support the development of games in a predictive analytics context. These games serve as part of an overall family of systems designed to gather input knowledge, calculate results of complex predictive technical and social models, and explore those results in an engaging fashion. The games provide an environment shaped and driven in part by the outputs of the models, allowing users to exert influence over a limited set of parameters, and displaying the results when those actions cause changes in the underlying model. We have crafted a prototype system in which we are implementing test versions of games driven by models in such a fashion, using a flexible architecture to allow for future continuation and expansion of this work.

Introduction

The Technosocial Predictive Analytics Initiative (TPAI) at PNNL is an effort devoted to “addressing complex, interwoven issues with highly integrated, innovative models to help analysts and policy makers identify and counter strategic surprise.” (Pacific Northwest National Laboratory 2008) The TPAI is divided into three research focus areas: *Technosocial Modeling (Area 1)*, *Knowledge Inputs (Area 2)*, and *Cognitive Enhancement (Area 3)*. In the Cognitive Enhancement research focus of the TPAI, we use a gaming approach to facilitate exploration of the predictive model behavior as well as to forecast prospective human dynamics through game play.

A strong motivating force for employing serious games is the belief that a game-based interface can increase the bandwidth of the human-machine interface, either by utilizing additional “channels” (e.g., audio or tactile feedback) or by accessing cognitive pathways/mechanisms that are not utilized when an analyst reads a report, runs a computer simulation, or views static data visualizations. Though the claim that games can enhance human cognition and learning are oft-repeated, the research literature is still largely inconclusive on this point (Susi et al. 2007). Nonetheless, there is little question that serious gaming can provide an

environment that is highly engaging for users, encourages social and collaborative learning and discovery, and accesses different mental faculties than, for instance, reading a report. Whether these differences between game-based environments and traditional techniques for interacting with complex simulations represent cognitive enhancements remains to be seen. Addressing this question in the specific case of complex techno/social models is an important element of the TPAI. Hence, one of the objectives of this task is to identify and, where possible, evaluate, the specific opportunities for cognitive enhancements resulting from the use of game-based interfaces to facilitate the analysis of complex social/technical interactions that characterize the modern world.

Background

We use the term *cognitive enhancement* to refer to enhanced understanding of information or an analytical task that results from taking advantage of non-traditional cognitive pathways – for instance, using higher-level image processing or music processing regions of the brain to access complex memory associations. We are not concerned with enhancing human cognitive powers via the use of implanted devices or drugs, which is also referred to as cognitive enhancement (Sandberg and Bostrom 2006).

Serious Games

The term *serious games* has been used in several ways in recent years. Serious games often involve the use of gaming technology to support traditional training in real world problems (Zyda 2005). But serious games also offer new types of training: the availability of high-quality game engines provides the capability to allow multiple users to operate in a simulated environment, allowing them to experience scenarios and explore options (Chatham 2007). There is a steadily growing games industry, and increasingly a set of opportunities to define a science of games (Zyda 2007). Successful serious games encapsulate established expertise into an environment that enables the player to explore and to develop new understanding (Kelly et al. 2007). Barr et al. (2007) explores the idea that games are different than

other Human-computer interaction (HCI) modes in that the game directs players to particular activities; this leads the players to act according to implied values systems that are built into the game.

Serious Games and Decision Support

Schreiner (2008) describes several efforts that attempt to utilize gaming technology to improve decision making by exploring social change. Peacemaker (Burak et al. 2005) allows players to explore the actions and perspective of an Israeli prime minister or the Palestinian President.

Cognitive Enhancements

Several types of enhancement can be provided by serious gaming technology.

Augmented and Social Cognition -- Extending Cognition beyond the User One means of enhancing cognition is to provide ways of extending the user's sensory capacities, enhancing their memory, or involving additional users in the cognitive task. Three areas of particular interest to our team include *temporal bookmarking*, *organizing flux*, and *social cognition*.

Game environments often introduce a strong temporal element to the user's experience. If the game is built with an explicit functional linkage to a simulation or model, as is expected in the TPAI effort, then the game provides a means for exploring changes in the state of the simulation as time progresses. Ideally, the game environment will also provide a means for navigating in time – including the ability to replay earlier segments of the simulation through *temporal bookmarks* which tag specific events in time. Visually, such bookmarks may be represented by markers, icons or labels which persist in the information space, but which change in appearance (e.g., become more transparent, change color, brightness, or size) as the user moves forward or back in time from the bookmarked event. Such conventions allow users to annotate an event or sequence of events, either as a mnemonic device, or to bring the event to the attention of other users, similar to the “asynchronous collaboration” described in (Moloney and Harvey 2004).

Organizing Flux refers to the use of cues that appear as a game scenario progresses, drawing the players' attention to specific, important changes as they occur. The importance of changes might be determined by parameterized rules implemented in the game engine or by other game participants. Cues may show up for only a short time, or could persist until acknowledged, depending on the use case.

A growing body of research seems to support the somewhat counter-intuitive observation that “the wisdom of the crowds” may actually outperform domain experts in making predictions about future events. Sensing this wisdom—or *social cognition*—capitalizes on the increasingly social nature of game interfaces provides an opportunity to engage large numbers of people in problem analysis and developing solutions.

Channel Capacity An additional cognitive enhancement is to use the gaming interface to the bandwidth between the human and the problems being addressed. Compared to traditional modeling and simulation environments, games provide opportunities for opening up additional sensory and cognitive channels for getting information to the user. Rich audio and animation, for example, provide additional bandwidth for interacting with the human user, as well as a broader palette of cues which can be used to encode information intended for the user. For instance, animated data can take advantage of relative differences in motion speed, direction, acceleration, and steadiness to help convey information, in addition to the usual visual cues of color, intensity, brightness, contrast, transparency, etc. In addition, games may tap into visual metaphors or symbols capable of conveying large amounts of tacit information instantaneously, possibly in ways that are well beyond the ability of static text or graphical displays. In addition, the structure of some games allows the analytic or problem solving task to be distributed to several players simultaneously, allowing each of them to specialize in some aspect of the overall solution. This can increase the total amount of analytic attention devoted to the overall problem.

As an example of increased bandwidth within an immersive 3D environment, imagine the players control an avatar that is rendered as part of the game environment. In such a game, it is possible to gain insights into problem domain coverage by receiving constant visual feedback about the problem “areas” that other team members are currently looking at (Kot et al. 2005).

Stimulating Creative Thinking A well-designed serious game can be thought of as a problem-solving “sandbox.” The game environment engages a different approach to problem solving; one which encourages creative solutions and encourages risk taking and speculation. On the other hand, most game environments (serious games or otherwise) are not entirely “open ended” due to the difficulty of anticipating and responding to unbounded user inputs. Allowing creative and innovative inputs/responses by users within the constraints of practically implemented gaming environments presents a major challenge to the serious game developer. It is in this area that we believe we can most effectively employ gaming techniques and technologies to develop a decision support system, relying on robust back-end models to provide a greater detail of open endedness. It is also possible that players—through their exploration of model outputs—may implicitly define areas in which a model could be extended or refined. For example, if we note that many players are trying to take an action that isn't supported by the underlying model, this is important feedback that should be considered by the game master and modeling team.

A Framework for Using Models and Game Technology for Decision Support

Defining the Game

The game architecture supports the interface between results obtained from existing domain models and players interacting in a controlled, scripted environment. The architecture is intended to provide an abstract way of describing the interactions between the players and models; while the architecture does not prescribe any one gaming paradigm, it does provide some elements to support the definition of player roles and to ensure that the interactions make sense in the environment being modeled. A game configuration specifies

- a set of game parameters
- a set of domain models
- a set of roles
- a set of game elements
- a set of handles

The *game parameters* specify the game space; the values of the game parameters determine the current state of the game world. Each parameter is associated with a description of its meaning and a data type. The *domain models* are simulations that are external to the game; they are specified by describing their input and output parameters, each of which has a description and a data type. The *game elements* interact with the gaming environment, the controls and displays used by the players, and the game parameters. A game element may display the value of one or more game parameters within the gaming environment; game elements also provide a way for the values of game parameters to be manipulated by players. Finally, the *handles* are the logical abstraction of things that the players are able to influence. Handles drive changes to the game elements when manipulated by the players.

Roles The *roles* specify the requirements for players in the game. In order to guide play, each player has an objective function whose value they attempt to minimize. The objective function can be thought of as a *payoff function* in a decision theory model (Kot et al. 2005) that is implicitly conditioned by the current state of the game parameters. A role also specifies which information the player should be able to access in order to make decisions and the actions they can take while attempting to meet their objective.

The Game Master The process for developing a game demands tight coordination of Area 1, 2, and 3 activities. The responsibility of the application of the game framework to a new problem domain is the province of the *game master*. The game master identifies suitable domain models, identifies compelling roles, and determines what game elements and parameters will be required for an engaging story-line and to support the rules of game-play. The game master is supported by

subject matter experts in the domain and specifies data needs to the technosocial modeling teams.

Game Execution

At an extremely high level, a game execution environment accepts a game configuration and produces a record of the game play; however, a game execution is also required to allow players to interact with each other and with the domain models through manipulation of game elements. In order to allow for post-game analysis, all changes to game parameters, invocations of game models, and changes in the values of objective functions are logged.

Game play is controlled by a game clock. At a pre-set interval, the following steps occur:

- If changed, the values of objective functions for roles are logged.
- If the values of any game parameters have changed, log the new value.
- Update game elements from model interactions: for each domain model, if the value of a game parameter mapped to its input parameter has changed, invoke the model and update game parameters that correspond to the models output parameters. The model invocation is logged.

Integration with Existing Models

Emerging from the technosocial modeling element of the TPAI is a series of physical, meteorological, agricultural, engineering infrastructure, and social models, integrated with the objective of identifying plausible futures given a set of initial conditions. The physical and social domains of these models provide the backdrop for gaming scenarios, and the model predictions define the consequences of player actions.

Let D represent the set of semantics domains that are required to represent a particular problem domain. Each model suite m (that is, an integrated model set associated with each of the three Area 1 projects), or *model* for short, can be represented as a map:

$$m: I^m \rightarrow O^m$$

where I represents input and O represents output, with $I^m = I_1^m \times I_2^m \times \dots \times I_{|I^m|}^m$ with each $I_j^m \in D$, and O^m has a similar definition. This model forms the basis for game play by interpreting a subset of the model input parameters as decision variables; decision variables are the subset of the model's input parameters can be mapped, through game parameters, to the handles defined in the game. The decision variables represent factors that some social entity (an individual, organization, or larger societal unit) would be expected to control or strongly influence. For instance, a parameter that characterizes an aspect of agricultural land management, such as crop irrigation volume, may be a decision variable, whereas, an input parameter defining mean wind speeds would not. Game play then involves the adjustment of handles

(driving changes in decision variables) by the players - each representing the corresponding social group - while the technosocial model is, in effect, the physics engine that dictates the response of the environment to player actions.

Supporting real-time game-play As a practical matter, the run times associated with certain component models is too long to support a high-pace, engaging game, and, therefore, a fast-running model that approximates the properties of the original model is required. To support these surrogate models, we define a mechanism for creating a new domain model from an existing model. This type of model accepts a domain model and a set of valid inputs to that model and creates a new domain model that returns the same values as the given model would on the given inputs. On non-specified inputs, the output of the new model is undefined, although a particular implementation may provide a mechanism for interpolating an output value from known output values. In addition, the implementation may provide for the ability to record requests for unknown inputs. General means of developing such surrogate models are under evaluation, including the use of response surface methodology and/or regression models; however, the approach currently adopted is one in which a *mesh* of model input/output vector values is generated by the modeling team in accordance with the specifications of the Game master.

The mesh can be specified as the combination of suitable subspaces of the input parameter space, with each subspace specifying, for each input parameter, a subset of the parameter's domain that should be included in the mesh. The modeling team produces, for each input vector that matches the mesh specification, a corresponding output vector. The resulting mesh can be represented as

$$(\mathbf{x}, \mathbf{y})_i$$

where the index i runs over the calculated points in the mesh, and each point consists of \mathbf{x} , a single set of input parameter values, and \mathbf{y} , the resultant vector of output parameter values. This pre-established set of input/output vectors provides the basis for game play in lieu of the original model.

Since this mesh constitutes a series of discrete input/output vectors, several issues merit consideration. These include the question of how the I/O behavior of the original model dictates the appropriate density of the I/O vector mesh and where interpolative rules (for handles that are adjusted by players to lie between mesh points) need be established to meet various gaming contingencies. The modeling team should be consulted during the creation of the mesh specification to ensure that the specified input domains make sense in the context of the modeled domain. For instance, if there is a region of the I/O vector space where an output parameter becomes acutely sensitive to the values of certain input parameters, then a finer mesh may be warranted in that

region of the space and means of interpolation may become more critical.

Framework Architecture

With our formal definition and integration scheme in hand, we crafted a framework architecture to facilitate building of games for the TPAI. We note the following important features of the task and environment:

Flexibility of the “Engine”

As noted above, integration with the modeling tasks of the TPAI requires our games to be capable of dynamically adjusting the behavior of the game environment and game elements within the environment to respond to changes in the model outputs.

Relaxed Restrictions on Gameplay

In a typical online game, paying customers undertake tasks to drive outcomes, ultimately arbitrated by a server program that instantly determines the results of the combination of players' actions. Players are paying to be entertained, and have expectations regarding how the game should function, and how quickly it must respond to their actions. If the game does not deliver instant response in a (fairly) predictable fashion, players simply disconnect. In our case, we relax these restrictions somewhat, in light of the fact that our expected players are using the game as a tool in the execution of their tradecraft. It is expected that our games may at some points need to pause to allow for decision making by the Game Master and associated subject matter experts. While pausing is anathema to multiplayer online *entertainment* games, in our case the players are part of a team with an overall professional goal, and we therefore expect them to be somewhat more tolerant of occasional interruptions in the flow of the game.

Componentized Architecture

In the development of our prototype system—which forms the basis for our game engine as we continue our work—we have been careful to build in the levels of abstraction as defined in previous sections. For example, in cases where we have only 1:1 mappings of game elements to game parameters, 1:1 mappings of game parameters to model parameters, and so on, we avoid the shortcut of allowing game handles to directly manipulate model parameters.

Preserving the “middleware” abstractions allows us to break out individual classes in code, providing a flexible architecture in which we can rapidly plug in a variety of model and game components.

In our example system, we define “widgets,” which are the objects responsible for direct input/output relationships with the user. A *widget* provides the mapping between a *handle* and a *game element*, and

provides the physical manifestation of visualizations for a *game element*. We define two types of widgets: one that relates a game element to a simple form control (e.g. a button, text box, slider, or similar), and one that relates a game element to a 3D object rendered in a game space. This pairing of styles allows us to craft blended game interfaces, using form-based inputs when that is the simplest way for a user to provide input, and using a 3D display when we can make use of animation and spatial features to convey information to the user more effectively than with text.

Creating the Game

The game master manages the iterative and interactive process of creating the game. Participants in that process include the customer (for whom the predictive analysis is being undertaken), the Area 1 modeling team, appropriate subject matter experts, and the game master, who brings expert knowledge of serious gaming techniques. The scope, content, scenario, player identities, and rules of the game are driven by the customer's objectives. The issues of interest determine the selection of appropriate models, the player identities dictate which model input parameters are selected as *handles*, and the background scenario dictates the values at which model input parameters are set and the ranges over which the handles will be allowed to vary. The game making process is of an iterative nature since the roles, scenario, and rules identified by the game master may require a modeling capability not already incorporated into the Area 1 models. This situation will generally demand the identification of appropriate compromises between model augmentation and role play capability.

A critical consideration in game creation is establishing the motivations, objectives, and constraints for each of the player roles. Overall objectives may be partially competitive and partially collaborative between players, and the specific natures of these objectives need to be established. In general, the objectives will be representable in terms of model output parameters (such as a given output level - energy production rate, for instance - must exceed a specified threshold). There will also exist constraints on the actions of each player, again representable in terms of some combination of model input and output parameters. Often, quantification of the parameters relative to which the objectives and constraints are defined may require some supplementary analysis, even if this involves only simple combinations of the Area 1 model input/output parameters.

These specifications focus principally on the I/O space requirements, such as the boundaries of the game space (handle values), the mesh density, and interpolative rules. Based on their understanding of the game objectives and scenario, the Area 1 modeling experts may provide the game master with supplementary information to help ensure meaningful game play, such as identification of nonphysical regions of the vector space involving

parameter combinations that are not credible, or other parameter dependences that need to be respected during game play. This information provided by the Area 1 modelers is then cached in the Knowledge Encapsulation Framework (Area 2) along with any supplementary data resources that will be used to enhance visualization of the game scenarios and outcomes (such as maps or other graphic representations).

It is worth noting that the definition of the aspects of a game can be defined entirely outside of code. The practice of "paper prototyping" is popular among disciplines in which user centered design is employed to ensure that users are comfortable with and understand an interface. In the case of gaming there is also a need to make sure that the rules make sense and that the game is playable (in pure entertainment gaming, making sure that the game is fun and repeatable are tantamount; in our case, we can relax those restrictions slightly, but not entirely, as we still desire for our games to be engaging). To that end, we advocate an approach that involves documenting the game primitives (roles, elements, etc) and crafting a "paper" version of a game, noting that we don't literally mean paper in every case. For example, in the following section we describe an example game that was first implemented using a spreadsheet as the "game engine."

Building an Example

Once the initial architecture for building a TPAI game was defined, the next step was to develop a proof-of-concept game based on a simple model. The primary purpose of developing this initial model-based game was to evaluate and refine our methodology and mechanics before moving to a more complex modeling problem. Additionally, if a playable game can be developed from a simple model, it was assumed that scaling our architecture design to a more complex model is an achievable goal.

For the initial game development, we identified a simple supply and demand-based profit model for use in the gaming scenario. Defining the model parameters and handles prior to game development was an essential step in this process because these factors will be predefined in future gaming scenarios. The model parameters were:

$$\pi(x) = xp(x) - C(x) - t(x)$$

Where:

x is annual amount sold

$\pi(x)$ is the annual profit given annual production rate x

C(x) is the cost for the resources necessary to produce x

t is the tax rate per unit sold

p(x) is price per unit

Based on this initial prototype, the next step was to transform the model parameters into a playable game. To gain insight into how we might form a game to leverage this model, we created a "paper" game as described in the previous section. In this case we used a spreadsheet as the

“game engine” and found that due to the static nature of the model selected, a turn-based game made sense, with players attempting to achieve some goal relative to “their” individual pieces of the model equation.

The first step in this transformation was to invent a cover story that contained the gaming elements that could be mapped to the pre-defined model parameters. The scenario for this game involved the rise and fall of the housing market, where each of the individual players were responsible for an element of the market (e.g. a builder, speculator, banker, and buyer). Next, a graphical user interface (GUI) was designed that provided a playable gaming environment. The GUI included individually tailored displays that allowed for the manipulation of each player’s handles as well as a situational overview that was seen by all players. The GUI was designed with an eye toward future deployments. For example, the current design provided a map-based overview of the game environment since future models will likely include a geographical component. Once the interface was designed, the game elements were mapped to the model parameters. Although all the model parameters were included within elements of the game, not all game elements were mapped to model parameters. For example, we include elements that provide feedback about historical data (the results of prior turns), noting that – while this simplified model doesn’t require historical context – the player requires this information to make better, and more realistic, decisions.

Conclusion and Future Work

We have described an architecture and methodology that enables the combination of complex technical models with game-driven interfaces. Currently the project team is designing games around use-case scenarios from the TPAI Area 1 modeling projects. The initial game will be built around the ‘Biofuels in the Asian Subcontinent’ scenario.

The purpose of this game is to explore the outputs of the models being developed by Malone et al. in the context of their “Vulnerability of Food Security and Energy Infrastructures to Climate Change and Terrorism” project. This game will explore decision spaces involving the competing demands for agricultural resources to produce biofuels and food crops. The game is designed to help decision makers understand:

- How US policy initiatives will be viewed by India’s populace and policy and decision makers
- Benefits and drawbacks to American and Indian economic, political, and environmental interests
- How other American allies and strategic partners (Pakistan, Bangladesh, and possibly China) will view American efforts and India’s responses to them.

Work is underway on defining the components (Parameters, Elements, Handles, etc.), scenarios, and rules.

In the coming year, we will also be working with the other modeling tasks within the TPAI to further exercise and refine our methodology.

References

- Barr, P., et al. (2007). "Video game values: Human-computer interaction and games." *Interacting with Computers* 19(2): 180-195.
- Burak, A., et al. (2005). PeaceMaker: A Video Game to Teach Peace. *Intelligent Technologies for Interactive Entertainment, First International Conference, INTENTAIN 2005*. M. Maybury, O. Stock and W. Wahlster. Campaglio, Italy, Springer. 3814: 307-310.
- Chatham, R. E. (2007). "Games for Training." *Communications of the ACM* 50(7): 37-43.
- Kelly, H., et al. (2007). "How to Build Serious Games." *Communications of the ACM*. 50(7): 45-49.
- Kot, B. W., et al. (2005). Information visualisation utilising 3D computer game engines case study: a source code comprehension tool. *6th ACM SIGCHI New Zealand Chapter's international Conference on Computer-Human interaction: Making CHI Natural*. Auckland, New Zealand, ACM. 94.
- Moloney, J. and L. Harvey (2004). Visualization and 'auralization' of architectural design in a game engine based collaborative virtual environment. *Eighth International Conference on Information Visualisation*: 827-832.
- Pacific Northwest National Laboratory. (2008). "Techosocial Predictive Analytics Initiative." Retrieved October 1, 2008, 2008, from <http://tpai.pnl.gov/>.
- Sandberg, A. and N. Bostrom (2006). "Converging Cognitive Enhancements." *Annals of the New York Academy of Sciences* 1093: 201-207.
- Schreiner, K. (2008). "Digital Games Target Social Change." *Computer Graphics and Applications, IEEE* 28(1): 12-17.
- Susi, T., et al. (2007). Serious Games – an Overview. Skovde, Sweden, University of Skovde.
- Zyda, M. (2005). "From visual simulation to virtual reality to games." *Computer* 38(9): 25-32.
- Zyda, M. (2007). "Creating a Science of Games." *Communications of the ACM*. 50(7): 26-29.