

An Inclusive View of Player Modeling

Adam M. Smith, Chris Lewis, Kenneth Hullett, Gillian Smith, Anne Sullivan
Center for Games and Playable Media, UC Santa Cruz
Santa Cruz, CA 95064, USA

{amsmith, cflewis, khullett, gsmith, anne}@soe.ucsc.edu

ABSTRACT

“Player modeling” is a loose concept. It can equally apply to everything from a predictive model of player actions resulting from machine learning to a designer's description of a player's expected reactions in response to some piece of game content. This lack of a precise terminology prevents practitioners from quickly finding introductions to applicable modeling methods or determining viable alternatives to their own techniques. We introduce a vocabulary that distinguishes between the major existing player modeling applications and techniques. Four facets together define the kind for a model: the *scope* of application, the *purpose* of use, the *domain* of modeled details, and the *source* of a model's derivation or motivation. This vocabulary allows the identification of relevant player modeling methods for particular problems and clarifies the roles that a player model can take.

Categories and Subject Descriptors

K.8.0 [Personal Computing]: General – *games*.

General Terms

Design, Human Factors.

Keywords

Player modeling, taxonomy, games, game design.

1. INTRODUCTION

What are the various kinds of player modeling? How do you tell if something even *is* a player model? Each of the authors originally had different perspectives on the kinds and desiderata of player modeling, emerging from exposure to different literature and the undocumented practices in game design. The term “player model” is nebulous, and a broad spectrum of possible definitions has been used in various publications. While many people have described player modeling and discussed its importance, there is no single clear definition or scope, and the definition and the usage of the idea has grown organically over time.

Our goal in this paper is to categorize the different kinds of player modeling that exist in practice in a way that makes clear which kinds of player models are applicable to which problems and what related kinds of models make for viable alternatives. To do this,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FDG'11, June 29-July 1, Bordeaux, France.

Copyright 2011 ACM 978-1-4503-0804-5/11/06 ... \$10.00.

we introduce a multi-faceted vocabulary which distinguishes player models on the basis of who they apply to (*scope*), what they are used for (*purpose*), the kind of details they model (*domain*), and how they are derived or motivated (*source*). It is important to note that we do not intend to provide an absolute definition for player modeling; we instead adopt an *inclusive* approach by casting a wide net to consider any work that has been published using the term “player modeling” and also many game design practices that are not traditionally labeled as such.

2. TAXONOMY

Our goal has been to build a broadly applicable taxonomy that can describe player modeling techniques across all games, both digital and non-digital, and in all game genres. This taxonomy provides a vocabulary for discussing the various different kinds of player modeling approaches that are already taken in industry and academia, and reveals areas where the field is currently lacking.

The taxonomy consists of four independent facets; it is important to note that these facets are non-hierarchical but are rather orthogonal to each other. A player model is described with a *kind*, defined by a selection from each of the four independent facets. Stringing the names of these selections together (in any convenient ordering) provides a very dense label for one corner of the space of player models, e.g. “Individual Induced Descriptive Reaction” models. Facets can also be omitted to abstract over a broader range of player models, such as “Action Generators” or “Hypothetical Analytic” models. Table 1 briefly describes these facets and the values they can take on for a particular player model. A long-form description for each facet and analysis of contextual examples is available in our technical report [14].

Table 1. Our taxonomy consists of four independent facets. This table lists these facets and their potential values.

| Domain | Purpose |
|--|--|
| Game Actions <i>details recorded inside of the game's rule system</i> | Generative <i>literally produces details in place of a human player</i> |
| Human Reactions <i>details observable in the player as a result of play</i> | Descriptive <i>conveys a high-level description, usually visually or linguistically</i> |
| Scope | Source |
| Individual <i>applicable only to one player</i> | Induced <i>learned/fit/recorded by algorithmic means</i> |
| Class <i>applicable to a sub-population</i> | Interpreted <i>concluded via fuzzy/subjective reasoning from records</i> |
| Universal <i>applicable to all players</i> | Analytic <i>derived purely from the game's rules and related models</i> |
| Hypothetical <i>unlikely to be applicable to any players, but interesting nonetheless</i> | Synthetic <i>justified by reference to an internal belief or external theory</i> |

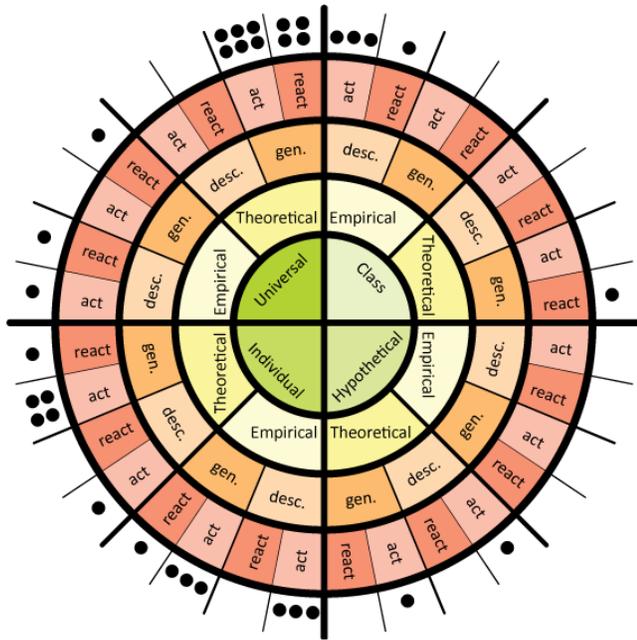


Figure 1. A visual summary of a representative sample of player modeling techniques. The Induced and Interpreted

Two simple examples of categorizing existing player modeling techniques using our taxonomy are the Playtracer system [1] and Yannakakis’s preference modeling. Playtracer is a tool for analyzing recorded play traces, which produces Individual Induced Descriptive Action player models in our terms. These take the form of a directed graph indicating (for several players at a time) trajectories through the abstract puzzle space of the game *Refraction*. These highlight common stumbling blocks on a player’s progression towards a goal state. Yannakakis’s work builds models of player reactions. This work generally produces Induced Reaction models. Meanwhile, game designers will naturally build up strong expectations for how their audience will react to their work, exercising Synthetic Reaction models.

A more sophisticated example is the multi-layered player modeling done in Thue’s *PaSSAGE*. An Individual Induced Descriptive Action model maps a player to a label (associating players who often engage in combat actions with the “fighter” label). Another model in the same system, a Class Synthetic Generative Reaction model, takes the form of a table encoding how players of a given type rate the suitability of a given event happening in their play experience (predicting that a fighter would rate “headlong assault” strongly positive). Layering these two player models together results in an Individual Synthetic Generative Reaction model which is able to, on a player-by-player basis, generate suitability scores which can be used to select the best event to trigger in the game world next (realizing an adaptive, personalized gameplay experience). This kind of layering (pairing a Descriptive model with a Generative model) can be used to incorporate Descriptive models into the live execution of a game. In layering, two different kinds of models are used to implement a larger player model, that when treated as a black box has a distinct kind of its own.

Descriptive models can be used to document assumptions made about the game or its intended authors. Walkthrough solutions (or input scripts) sometimes provided with interactive fictions [12]

Table 2. A listing of the representative systems depicted as dots in Figure 1. Rows are ordered as to label the dots in a clockwise fashion, starting from the top.

| Instance | Scope | Source | Purpose | Domain |
|---------------------------------------|--------|----------|---------|--------|
| "Speed-runner" and "completionist" | Class | Interp. | Descr. | Act. |
| Bartle’s player models [2] | Class | Interp. | Descr. | Both |
| WoW guild archetypes [21] | Class | Induced | Descr. | Act. |
| PaSSAGE [20] | Class | Synth. | Gen. | React. |
| Storyboards [5] | Hypo. | Synth. | Descr. | Act. |
| Ludocore [15] | Hypo. | Analytic | Gen. | Act. |
| Houlette [7] | Indiv. | Induced | Descr. | Act. |
| Playtracer [1] | Indiv. | Induced | Descr. | Act. |
| PaSSAGE [20] | Indiv. | Induced | Descr. | Act. |
| Race track generation [23] | Indiv. | Induced | Gen. | Act. |
| Drivatars [6] | Indiv. | Induced | Gen. | Act. |
| NonyBot ¹ | Indiv. | Interp. | Gen. | Act. |
| Polymorph [10] | Indiv. | Induced | Gen. | React. |
| Interactive fiction walkthroughs [12] | Indiv. | Synth. | Both | Act. |
| QuakeBot [11] | Indiv. | Synth. | Gen. | Act. |
| IBM’s Deep Blue and Watson [9][8] | Indiv. | Synth. | Gen. | Act. |
| Mario bots [22] | Indiv. | Analytic | Gen. | Act. |
| PaSSAGE [20] | Indiv. | Synth. | Gen. | React. |
| Heatmaps for Halo 3 [19] | Uni. | Induced | Descr. | Act. |
| Preference modeling | Uni. | Induced | Descr. | React. |
| Polymorph [10] | Uni. | Induced | Gen. | React. |
| Endgame tablebases [3] | Uni. | Analytic | Gen. | Act. |
| EMPath [17] | Uni. | Analytic | Gen. | Act. |
| IMPLANT [18] | Uni. | Analytic | Gen. | Act. |
| Ludocore [15] | Uni. | Analytic | Gen. | Act. |
| Market bots [13] | Uni. | Synth. | Gen. | Act. |
| Launchpad [16] | Uni. | Synth. | Gen. | Act. |
| EMPath [17] | Uni. | Synth. | Gen. | React. |
| Race track generation [23] | Uni. | Synth. | Gen. | React. |
| Flow inspired [4] | Uni. | Synth. | Gen. | React. |
| Mario bots [22] | Uni. | Analytic | Gen. | React. |

are Individual Synthetic models. However, it is interesting to note that to judge the Purpose of such models, one needs to reference a particular usage of the script. These scripts afford both Generative and Descriptive purposes: they are both executable as Action Generators and readable as natural-language Action Descriptors.

3. DISCUSSION

Figure 1 visually suggests that there are voids in our taxonomy: categories that apply to no realized player models. While some areas would likely remain unpopulated in even the most comprehensive survey of player modeling work, we assert that there are no unrealizable combinations of facet selections. Consider the seemingly contradictory kind of Hypothetical Induced models. At first blush, it seems unlikely that one could build a model of hypothetical player behavior when looking at

¹ <http://www.youtube.com/watch?v=AxBHwpItv84>

data from real players. However, suppose that an analysis of logs from a first person shooter shows that all but one weapon in the game is used relatively frequently. Immediately one tries to think of what it is like to play using that weapon to guess at why it would be ignored: this is a Hypothetical Induced Descriptive Action player model.

In our informal estimation, published work seems to cluster around Universal theoretical models and Individual empirical models. We think this is because theories usually attempt to be universally applicable and individuals are the primary source of empirical data, making these models the most direct. Class models are more difficult to motivate in an academic context, requiring either justification of a theory of stereotypes or aggregation of sufficient individual data to build up class descriptors. Game designers regularly invent Individual and Class Synthetic models as a product of their amassed design experience. The latter ephemeral models are difficult to convey without the sum of indirect experience and other pet theories that inspired them.

Our goal in this paper has been to propose a terminology for player modeling in a way that ties modeling methods (best distinguished by their Source) to the modeling problem they solve (described by their Scope, Purpose, and Domain). We hope this taxonomy (again, available in a long-form report [14]) inspires the reader to adopt a more inclusive view of player modeling and inspires them use the distinctions it draws to formulate new and interesting conclusions about player modeling that were previous difficult to express in a general way.

4. REFERENCES

- [1] ANDERSEN, E., LIU, Y. E., APTER, E., GENESSE, F. B., AND POPOVIC, Z. Gameplay analysis through state projection. In *Proc. of the 5th Int'l Conf. on the Foundations of Digital Games* (2010), FDG '10, pp. 1–8.
- [2] BARTLE, R. A. Players Who Suit MUDs. *Journal of Online Environments 1*, 1 (1996).
- [3] BELLMAN, R. E. On the Application of Dynamic Programming to the Determination of Optimal Play in Chess and Checkers. In *Proc. of the National Academy of Sciences of the United States of America* (February 1965), vol. 53, pp. 244–246.
- [4] CSIKSZENTMIHALYI, M. *Flow: The Psychology of Optimal Experience*, 1st ed. Harper Perennial Modern Classics, July 2008.
- [5] FULLERTON, T. *Game Design Workshop, Second Edition: A Playcentric Approach to Creating Innovative Games (Gama Network Series)*, 2 ed. Morgan Kaufmann, February 2008, p. 168.
- [6] HERBRICH, R. Drivatar. <http://research.microsoft.com/en-us/projects/drivatar/default.aspx>
- [7] HOULETTE, R. Player Modeling for Adaptive Games. In *AI Game Programming Wisdom 2*, S. Rabin, Ed. Charles River Media, December 2003.
- [8] IBM RESEARCH. Ibmwatson. <http://www-03.ibm.com/innovation/us/watson/>.
- [9] IBM RESEARCH. Kasparov vs deep blue: the rematch. <http://www.research.ibm.com/deepblue/>.
- [10] JENNINGS-TEATS, M., SMITH, G., AND WARDRIP-FRUIIN, N. Polymorph: A Model for Dynamic Level Generation. In *Proc. of the 6th Artificial Intelligence for Interactive Digital Entertainment Conf. (AIIDE 2010)* (October 2010).
- [11] LAIRD, J. E. It knows what you're going to do: adding anticipation to a Quakebot. In *Proc. of the fifth int'l conference on Autonomous agents* (2001), AGENTS '01, pp. 385–392.
- [12] REED, A. *Creating Interactive Fiction with Inform 7*, 1 ed. Course Technology PTR, August 2010, p. 387.
- [13] REEDER, J., SUKTHANKAR, G., GEORGIPOULOS, M., AND ANAGNOSTOPOULOS, G. Intelligent trading agents for massively multi-player game economies. In *Proc. of the 4th Artificial Intelligence for Interactive Digital Entertainment Conf. (AIIDE '08)* (Oct. 2008).
- [14] SMITH, A. M., LEWIS, C., HULLETT, K., SMITH, G., AND SULLIVAN, A. An Inclusive Taxonomy of Player Modeling. Tech. Rep. UCSC-SOE-11-13, University of California, Santa Cruz.
- [15] SMITH, A. M., NELSON, M. J., AND MATEAS, M. Ludocore: A Logical Game Engine for Modeling Videogames. In *Proc. of IEEE Conf. on Computational Intelligence and Games (CIG 2010)* (August 2010).
- [16] SMITH, G., WHITEHEAD, J., MATEAS, M., TREANOR, M., MARCH, J., AND CHA, M. Launchpad: A Rhythm-Based Level Generator for 2D Platformers. *IEEE Transactions on Computational Intelligence and AI in Games 3*, 1 (March 2011).
- [17] SULLIVAN, A., CHEN, S., AND MATEAS, M. From Abstraction to Reality: Integrating Drama Management into a Playable Game Experience. *Proc. of the AAAI 2009 Spring Symposium* (2009).
- [18] TAN, C. T., AND CHENG, H.-L. IMPLANT: An Integrated MDP and POMDP Learning Agent for Adaptive Games. In *Proc. of the 5th Artificial Intelligence for Interactive Digital Entertainment Conf. (AIIDE '09)* (October 2009).
- [19] THOMPSON, C. Halo 3: How Microsoft Labs Invented a New Science of Play. August 2007.
- [20] THUE, D., BULITKO, V., AND SPETCH, M. Player Modeling for Interactive Storytelling: A Practical Approach. In *AI Game Programming Wisdom 4*. Charles River Media, February 2008, pp. 633–646.
- [21] THURAU, C., AND BAUCKHAGE, C. Analyzing the Evolution of Social Groups in World of Warcraft. In *Proc. of IEEE Conf. on Computational Intelligence and Games (CIG 2010)* (August 2010), pp. 170–177.
- [22] TOGELIUS, J., KARAKOVSKIY, S., KOUTNIK, J., AND SCHMIDHUBER, J. Super Mario Evolution. In *Proc. of the IEEE Symposium on Computational Intelligence and Games (CIG)* (September 2009), pp. 156–161.
- [23] TOGELIUS, J., NARDI, R. D., AND LUCAS, S. M. Making racing fun through player modeling and track evolution. In *Proc. of the SAB'06 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games* (2006).