Level of Detail Event Generation

Luis Flores and David Thue

Center for Analysis and Design of Intelligent Agents
School of Computer Science, Reykjavik University
Menntavegur 1, Reykjavik, 101, Iceland
{luis12|davidthue}@ru.is

Abstract. Level of detail is a method that involves optimizing the amount of detail that is simulated for some entity. We introduce an event generation method to optimize the level of detail of upcoming events in a simulation. Our method implements a cognitive model, which uses an estimate of the player's knowledge to estimate their interest in different aspects of the world. Our method predicts the salience of upcoming events, and uses this salience value to define the level of detail of potential new events. An evaluation of our method's predictive capacity shows generally higher accuracy than a baseline predictor.

1 Introduction

Level of detail (LOD) is a method that involves reducing the complexity of a simulation when doing so would be transparent to one or more players. It can decrease the complexity of geometry [3], artificial intelligence [3, 10], or physics computations [7]. Since simulating everything at a high level of detail could be very expensive, LOD is very important in video games, as it allows a larger game world to be simulated while still maintaining believability. Simulations can take advantage of this method by reducing the detail in which different aspects of the simulation are run, ideally producing and maintaining just enough data to serve the player's current interests.

In the context of Interactive Storytelling, conveying a rich simulation (or at least a convincing illusion thereof) can improve the believability of the story's world, and particularly with respect to its scale and feeling of being "alive" [11]. While distance and visibility are commonly used metrics when deciding the LOD of simulated *objects* (e.g., to speed up graphics rendering), these metrics seem unlikely to work for every aspect of a simulation, and particularly for the *events* that occur therein. For example, if an upcoming event is critical to the player's pursuit of a given goal, the simulation should ensure that that event is generated with a high enough LOD to allow the player to achieve their goals, regardless of how distant or visible the event currently is.

In this work, we propose a method to optimize the level of detail with which events in a simulated world are generated, using a given estimate of the player's knowledge to then estimate their interest in different aspects of the world. By estimating a potential event's level of interest for the current player (i.e., its *salience*) we can avoid generating unnecessary details for that event.

1.1 Background

Our model of salience draws inspiration from Cardona-Rivera et al.'s *Indexter model* [1], which predicts the salience of previous events in the user's memory based on the current world state. Contrary to their work, the model that we propose uses salience as a metric when generating *future* events; a lack of salience for a potential new event means that we can reduce the amount of detail that gets generated for the properties of that event.

The Indexter model is based on the Event-Indexing Situation Model (EISM) by Zwaan et al. [12, 13], which is a cognitive model of online narrative comprehension. The EISM categorizes a narrative into events with important details, and each event is categorized by different key factors (called *indices*). The EISM represents each event by indexing the following elements: *Time*, *Space*, *Protagonist*, *Causation*, and *Intention*. Using the EISM, the Indexter model computes the salience of a past event with respect to a current event, and this computation is initially made with respect to each of the five indices. Two events have a value of 1 for a given index when they both "share" that index (i.e., they both occur in the same time frame or place, they both involve the story's protagonist, they are causally related, or they both serve a single intention). For example, two events that occur in the same location share the space index, and thus have a value of 1 for space salience. The Indexter model attaches a weight w_* to each index such that the weighted sum of all of the indices is bound to $[0, 1]$. Given this constraint, the equation to calculate the salience of event e_i w.r.t. e_n is:

$$salience(e_i, e_n) = w_t t_{e_n} + w_s s_{e_n} + w_p p_{e_n} + w_c c_{e_n} + w_i i_{e_n} \tag{1}$$

where t_{e_n}, s_{e_n}, p_{e_n}, c_{e_n}, and i_{e_n} represent the time, space, protagonist, causality, and intention indices, respectively. The authors of the Indexter model set the same weight to every index, that is $\forall *, w_* = 0.2$.

Some limitations arise when considering applying the Indexter model to LOD-based generation. First, it estimates salience as a relationship between two events (one past, one current), while we aim to estimate the salience of a *future* event with respect to the player's current knowledge. Second, it computes only binary salience values for each index, which is too coarse too allow for any more than two levels of detail, and also risks producing an unbelievable simulation when LOD-based generation is used. For example, if the player's current event was not at the location of a candidate event, the Indexter model would compute its space salience as 0 – the lowest possible value. In LOD-based generation, however, a minimum salience value must be reserved for "no detail" to enable large-scale simulations. By setting every non-local event to minimum salience, the Indexter model (as-is) would prevent the details of any such events from ever being generated, leading to an unbelievable simulation.

In this paper, we present an extension to the Indexter model that allows the salience of a potential future event to be estimated with respect to an estimate of the player's knowledge. To enable LOD-based event generation using this model, we expand the value range of three of the five indices and generate event details on a per-index basis.

2 Problem Formulation

We aim to generate events at varying levels of detail (LODs), based on a model of the player's expected interest in the events that could be generated. We use the term *salience* to describe this player interest, and *generator* to describe the system that produces the events. An LOD-based event generator that uses salience must meet several requirements.

First, to balance the simulation's computation with its believability, the level of detail that the generator uses must vary with the expected salience of each candidate event. Second, since salience must be used during the generation process, each event's salience must be given or estimated before generation occurs, and such an estimate can only be made *after* some properties of the candidate event are already known (such as its time, location, the agents involved, etc.). We assume that the salience of each event will not be given, and therefore must be estimated. Third, since judgments of salience are influenced by memory [1], salience should be estimated based on both the current state of the simulation and the player's knowledge of what has been simulated thus far. Fourth, each output of the generator must be a description of an event whose content was determined using the level of detail that the generator selected.

Intuitively, a good solution to this task can be identified by its production of events that players find to be salient, given their knowledge of the simulation and its current state. We will judge our solution by using it to predict the salience of different events (with respect to their properties) and comparing the results to ground truth values obtained from a survey.

3 Related Work

One popular approach to applying level of detail is to use distance-based metrics (e.g., between an object's position and the player's position [3]). The closer the object to the player, the higher the level of detail rendered. As we argued in Section 1, distance-based metrics can be insufficient for the generation of events.

Sunshine-Hill's *LOD Trader* [10] aims to optimize the perceptual quality of a simulation by estimating the realism of every agent within each scene. *LOD Trader* lacks or ignores the influence of the current player's knowledge of the given world, contrary to our approach.

In physical simulations, LOD techniques focus on adapting the type of physical model used for different objects by understanding the importance of the object and the efficiency of different models at reacting to the current simulation state [7]. Our method aims to simulate the events of a world; not its physics.

Estimating salience is an important part of information retrieval [6], social network analysis [6], and text summarization [8]. Typically, a graph-based ranking algorithm is used to estimate the importance of each node in a graph by considering the global information of the graph. The algorithm checks the number and the quality of the links connected through the graph to determine how important each node is. We employ similar graph-based techniques to estimate the salience of some event properties, including space and time (Section 4.1).

4 Proposed Approach

We propose a method to optimize the generation of upcoming events in a simulation using a level of detail approach. By using an estimate of the player's knowledge about both the simulation and the current world state, this method allows us to estimate the salience of different event properties during generation, which then determine the level of detail of the generated events. We define the properties of an event based on the indices in Zwaan et al.'s EISM model [12, 13], as described in Section 4.1. In general, events with many salient properties are created with a high level of detail, meaning that each event's properties are described in full detail. Events with few salient properties are created with a low level of detail, meaning that their properties are lightly detailed or omitted completely. Figure 1 gives an overview of how our generator produces descriptions of new events.

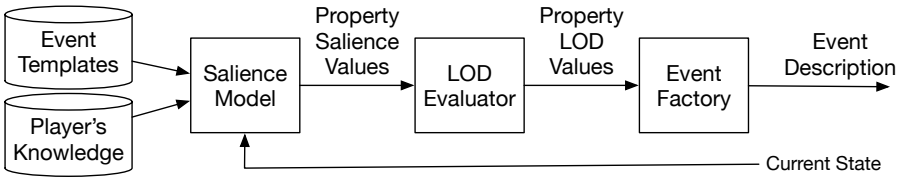

Fig. 1. Our process for level of detail event generation.

Our generator makes use of event templates, which constrain the values that each event's properties can take on during the generation process. Given an event template and an estimate of the player's knowledge, our generator first estimates a separate salience value for each property in the given event template (Salience Model in Figure 1). Given these values, it then decides a level of detail independently for each property (LOD Evaluator). Finally, our generator creates a description of the event based on the chosen LOD values (Event Factory).

4.1 Salience Model

Our generator's salience model estimates the salience of potential upcoming events with respect to the current world state and the player's knowledge. Specifically, we extend the Event Indexing Situation Model (EISM) [12] to define the salience of an potential future event. To give our generator a variety of levels of detail to choose from, the salience model estimates the salience of each event property independently. Properties are evaluated differently depending on how they affect the player or the current state of the world. The evaluated properties are protagonist (renamed "social"), time, space, causation, and intention. While the EISM protagonist index represents whether the player is involved in the event, we chose to expand the range of this value by additionally representing how socially close the player is to the agents involved in such event.

The **social salience** of an event depends on how closely the agents in the event are related to the player in a social sense. When the player is involved in the event, a salience of 1 is set. Otherwise, we create a social graph (Figure 2: right) to represent relations between agents and determine the social salience. To compute social salience for event e, we use the following equation:

$$social\_salience(e) = \begin{cases} 1 & \text{if player is involved} \\ 1 - \frac{social\_distance_e + 1}{n} & \text{otherwise} \end{cases} \quad (2)$$

where n is the total number of agents and $social\_distance_e$ represents the minimum number of edges between any agent that the player knows and any agent that is involved in the event e.

The **space salience** of an event is calculated by checking how close the event location is to the player. A salience of 1 occurs when the event is held in the same location as the player. Otherwise, a location graph is constructed where nodes are locations and edges represent traversable paths between locations (Figure 2: left). To compute space salience for event e, we use the following equation:

$$space\_salience(e) = \begin{cases} 1 & \text{if player is involved} \\ 1 - \frac{space\_distance_e + 1}{n} & \text{otherwise} \end{cases} \quad (3)$$

where $space\_distance_e$ is the minimum distance from event e's location to any location that the player knows and n is the total number of locations.

The **time salience** of an event depends on both the event time (when it occurs) and its propagation rate (how quickly its effects spread across the world). A low time salience means the event time is not salient to the player, or the event's effects will take too much time to reach the player. This model prefers events that reach the player sooner. To compute time salience for event e, we use the following equations:

$$time\_salience(e) = \begin{cases} 0 & \text{if } reach\_time(e) > limit \\ \frac{limit - reach\_time(e)}{limit} & \text{otherwise} \end{cases} \quad (4)$$

$$reach\_time(e) = \frac{player\_distance_e}{propagation_e} \quad (5)$$

where $reach\_time(e)$ represents the time that the effects of the event e will take to reach the player, $player\_distance_e$ represents the distance between the player and event e's location, $propagation_e$ represents the propagation rate of event e, and $limit$ is the longest time that the player will stay interested in an event.

The **causation salience** of an event was used in the EISM to represent whether or not two events were causally related. Since our model evaluates a single event in isolation, we adapted the causation index to indicate whether the player can potentially acquire a new goal (thus triggering a new causal chain of actions) by attending the event. We define causation salience $c_e = 1$ when the event e offers a new goal to the player and $c_e = 0$ otherwise.

The **intention salience** of an event was used in the Indexter model to represent whether or not two events were part of the same intentional plan.

Since our model evaluates only one event, we adapted this index to indicate whether the player can achieve a current goal by attending the event. We define intention salience $i_e = 1$ when the event e shares the same goal as the player, and $i_e = 0$ otherwise.

Once it has computed the salience of every property, the generator calculates the **total salience** for a given event. Similar to the Indexter model [1], we compute this total as a weighted sum of individual salience values with equal weights ($\forall *, w_* = 0.2$). We compute the total salience of event e as follows:

$$event\_salience(e, p) = 0.2s_e + 0.2p_e + 0.2t_e + 0.2c_e + 0.2i_e \tag{6}$$

where p is the player and s_e, p_e, t_e, c_e, and i_e are the property salience values for social, space, time, causation, and intention respectively.

The total salience is used to select the most salient event after all the event templates have been considered. Before generation can begin, our method must first determine a viable level of detail for each property in the selected event.

4.2 The Level of Detail Evaluator

The level of detail evaluator converts each event property salience value to a high, medium, or low LOD. The event factory avoids generating any details when a low LOD is used.

The **social property** has three levels of detail. *High* LOD: the player is involved; the event factory creates a full detail description of the agents involved. *Medium* LOD: a known agent is involved, or the agent involved is very close to a known agent; the event factory creates details of the known/close agents involved. *Low* LOD: the agents involved are unknown.

The **space property** has three levels of detail. *High* LOD: the event is happening in the same location as the player; the event factory creates a full description of the event location. *Medium* LOD: the event location is known to the player; the event factory only creates the location name. *Low* LOD: the location is not known to the player.

The **time property** has three levels of detail. *High* LOD: the event's effects will reach the player quickly; the event factory creates full details of the event's time. *Medium* LOD: the event's effects reach the player late, but it is still salient to the player based on the value of *limit* (Equation 4); the event factory creates the event date without the exact time frame. *Low* LOD: the event's effects reach the player late, and it is not salient to the player.

The **causation property** has two levels of detail. *High* LOD: the event allows the player to acquire a new goal; the event factory creates a description of the new goal that could be acquired. *Low* LOD: the event does not allow the player to acquire a new goal.

The **intention property** has two levels of detail. *High* LOD: the event satisfies one of the player's goals. The event factory creates the description of the event goal. *Low* LOD: the event does not satisfy one of the player's goals.

Table 1. Sample event templates.

	Event 1	**Event 2**	**Event 3**
Title	The orcs are gathering	A wild dragon appeared	A troll has kidnapped the wizard
Player	No	Yes	No
Agents	Traveler	Wizard	King, Sage
Locations	Valley	Fire Cave	Castle
Propagation	1	2	4
Goal	Spy on the orcs	Kill the dragon	Talk to the king
Cause	Talk to the king	Recover the treasure	Rescue the wizard

4.3 The Event Factory

Once we have the level of detail for each property in a given event template, the event factory creates an event description according to the given levels of detail. The event factory uses *Tracery* [2] to create event descriptions. *Tracery* is a grammar-based generation library and uses grammar rules to generate content. The event factory specifies one grammar rule per index, and each grammar rule is more or less detailed depending on the given level of detail. Finally, the event factory combines the grammar for each event property to create a full event description. Sample outputs are given in Section 4.4.

4.4 Event Generator in Action

Our generator begins by instantiating an event template, each of which is specified as a set of high level event properties as shown in Table 1. Overall, the generator estimates the total salience of each template and generates an event description from the template that has the highest total salience. For the sake of simplicity, consider the world shown in Figure 2. Assume that all the events would occur at the same time, and that the player's location is the *castle*.

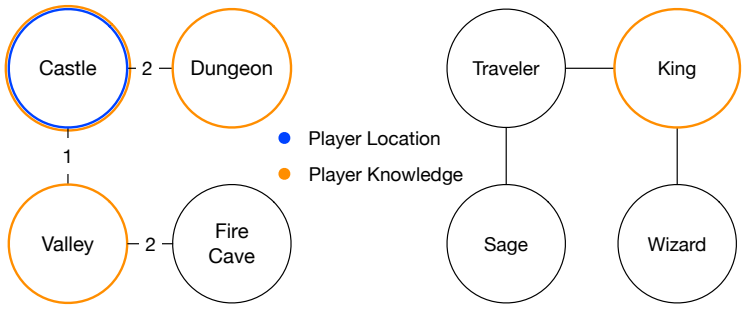

Fig. 2. Locations and social graphs. The player's knowledge is shown in orange. Numbered edges show distances between locations.

Table 2. Events salience and level of detail (L=Low, M=Medium, H=High).

Property	Event 1		Event 2		Event 3	
	Salience	**LOD**	**Salience**	**LOD**	**Salience**	**LOD**
Social	0.50	L	1.0	H	0.75	M
Space	0.75	M	0.5	L	1.0	H
Time	0.66	M	0.5	L	1.0	H
Intention	0.0	L	0.0	L	1.0	H
Causation	0.0	L	1.0	H	1.0	H
Total	0.38	N/A	0.6	N/A	0.95	N/A

The player's goals are to *talk to the king* and *recover the treasure*. The player has knowledge of the following locations: *castle*, *dungeon*, and *valley*. The only agent the player knows is the *king*. To compute the time salience, we set the *limit* time to 3 time units.

Table 2 shows the property salience values that would be computed for the event templates in Table 1, given the player's knowledge and the world state. The event factory would then create the following event descriptions for the levels of detail given in Table 2. Annotations appear as emphasized text.

Event 1: The orcs are gathering *(event title)*. It will take place at the valley *(space, medium LOD)*. It is happening today *(time, medium LOD)*.

Event 2: A wild dragon appeared *(event title)*. You are involved in this, and the hunter is attending as well *(social, high LOD)*. You will be offered the chance recover the treasure if you attend *(causation, high LOD)*.

Event 3: A troll has kidnapped the wizard *(event title)*. It is taking place at castle, it is happening outside the gates *(space, high LOD)*. The king is attending *(social, medium LOD)*. It will take place today at noon *(time, high LOD)*. Get ready to talk to the king *(intention, high LOD)*. You will be offered the chance to rescue the wizard if you attend *(causation, high LOD)*.

5 Evaluation

Given an event template and the player's knowledge, can we identify the "right" level of detail to use for each salience index, when generating the event for a given player? Since our generator aims to use higher levels of detail only for the more salient properties of an event, finding the "right" levels of detail amounts to correctly predicting the salience of each of the event's properties.

To evaluate our approach, we conducted a pilot study to assess which properties of potential future events would be more or less salient to players, given certain knowledge about a simulated world. We created two simple game worlds as testbeds, defining the world state and the player's knowledge of the agents, locations, and goals as shown in Table 3. To reduce bias in our experiment, the worlds have the same structure but different contexts: the location and social graphs in both worlds will result the same graph structure, but the characters and locations are different. Figure 2 shows the location and social graphs for

Table 3. Game worlds used to test our approach.

	World 1	**World 2**
Player Location	Castle	Husavik
Player's Goal	Talk to the king	Capture an elf
Known Locations	Castle, Dungeon, and Valley	Husavik, Keflavik, and Borganes
Known Agents	King	President

World 1; World 2 had four locations based on the real world and four agents named "tourist", "president", "bellman", and "dentist". The player's knowledge and world state were shown via text during the study.

For each event property and for each level of detail that our method would consider for that property, we proposed an event to each participant such that our generator would use the given level of detail for the given property. Then, we asked the players to rate how interesting the event seemed with respect to that event property. We asked one such question per property's level of detail, for each of our testbed world states. We asked questions related to the current world state and player's knowledge such as: *"will you be interested in a new event if it is located at the castle?"*, *"will you be interested in the new event if the king is involved?"*, and *"will you be interested in the new event if it is happening today?"* Participants could respond to each question by selecting "Very much interested", "Somewhat interested" (for social, space, and time), or "Not at all interested". We collected data for each event property at each of the levels of detail that our method considers for that property (recall Section 4.2).

For **social salience**, we asked the players how interested they would be in three new events, where each event involved one of the three different agents that would cause our method to use a high, medium, or low LOD, respectively. The high LOD is used when the social salience is between 1.0 and 0.75. The medium LOD is used when the social salience is between 0.75 and 0.5, meaning an agent involved is known to the player. A low LOD is used when the social salience is less than 0.5, meaning that agent unknown agents are involved.

For **space salience**, we asked the players how interested they would be in a new event for each of the three levels of detail for the space property. A high LOD is used when the space salience is between 1.0 and 0.75. A medium LOD is used when the space salience is between 0.75 and 0.5, meaning the event location is known to the player. A low LOD is used when the space salience is less than 0.5, meaning the event location is unknown.

For **time salience**, we asked the participants how interested they would be in new events involving three different levels of detail: when the event is about to happen, when the event is happening in a few days, and when the event will only happen after a long time.

For **causation salience**, we asked the participants how interested they would be in the new event if it would motivate them to achieve new goals.

For **intention salience**, we asked the participants how interested they would be in a new event if they could achieve their current goals by attending the event.

Table 4. Summary of results for the user study, using the worlds shown in Table 3.

Salience	World	Accuracy	Precision		
			High	Med	Low
Social	1	46.5%	77.0%	18.7%	43.7%
	2	54.8%	77.0%	25.0%	62.5%
Space	1	43.1%	66.7%	35.4%	27.1%
	2	50.0%	56.3%	54.2%	40.0%
Time	1	52.1%	60.4%	64.5%	31.2%
	2	49.3%	62.5%	54.1%	31.2%
Causation	1	65.6%	79.1%	N/A	52.1%
	2	58.3%	72.9%	N/A	43.7%
Intention	1	60.4%	79.1%	N/A	41.6%
	2	68.7%	89.5%	N/A	47.9%

5.1 Data and Results

Our model is trying to solve a prediction problem: given a particular event property, we predicted how much interest a property can produce in players. Our system effectively classifies each property as *very interesting*, *somewhat interesting*, or *not interesting* and then maps each of those values to the appropriate level of detail (*High*, *Medium*, or *Low*, respectively). In our pilot study, we gathered a total of 48 responses from participants recruited via social media. We computed accuracy and precision for each of the five event properties (social, time, space, intention, causation), for each world set. Accuracy measures how often our generator's predictions are correct overall. Precision measures how often our generator predicts a level of detail correctly. As we were unable to find a similar solution to compare against our method, we decided to compare our generator against a uniform random predictor. Table 4 shows our results.

6 Discussion and Future Work

The results for the five saliences values showed accuracies much better than what a uniform random predictor would produce (33.3% and 50.0% when using as two or three levels of detail, respectively). However, if we compare against a random predictor that considers the bias observed in the data, the results for social and space are disappointing. For example, a predictor that always predicted high detail would obtain accuracies near 50% for social salience, using the data from our study. That said, our results for social salience may have been biased by the agent names we used. Specifically, having an important agent or someone with a high authority (e.g., "king" or "president") may have raised levels of interest among players, even though they did not know those agents. We suspect that places with inherently interesting names can also influence the player's interests. We used a compelling name ("fire cave") in World 1 and a random real place name in World 2, and players were more interested in an event whose location had the more interesting name.

Players seemed to be somewhat interested even when our method predicted a low level of detail, suggesting that a generator should always display something at this LOD, rather than omitting all details. Given these results, we are interested in changing our approach to better distinguish between low detail and no detail.

The event factory used in our approach generates a simple text description of an event, which is not computationally expensive. We thus do not gain much computational savings from our approach. The gains for a more complicated factory could be larger, and testing this hypothesis remains as future work.

A few limitations arise when using our event generator's model. This model focused on a single player, and so we used the EISM's notion of a "protagonist" as the current player of the simulation. We would like to determine if extending this model is needed to handle more players, since important agents in the simulation seem to influence the interest of players.

Space and social saliences can be computed in a better way. For instance, we use the total number of nodes in the graph when creating the space or social graphs. This might be overkill if the simulation has a large number of locations and agents. A possible way to improve it is to use the width of the graph (the "longest shortest distance") instead of the total number of nodes. Finding good values for the parameters of our model, such as the time index's *limit* value, the thresholds for mapping property salience values to levels of detail, and the weights for computing total salience, remains an open problem.

Although we aim to support more than two levels of detail for each type of salience, we have currently done so for only three of the five types. We are interested in finding a way to compute causation and intention salience across a wider range of values; the Indexter model's use of a plan-based representation seems promising in this regard [1]. Finally, our method relies on having a reasonable estimate of the player's knowledge of the simulated world. While tracking what information a player has been exposed to is conceptually straightforward, knowing which information they have retained is a difficult problem [5]. It would be interesting to integrate our work with an active model of player knowledge [9].

7 Conclusion

Level of detail is a method that involves reducing the amount of detail that is generated for some entity. Using level of detail in a simulation has the potential to help improve performance, reducing computational load, as creating new events in a simulation can be resource intensive.

In this paper, we proposed an event generator that works to optimize the level of detail with which events in the world are generated, based on estimates of the salience of individual properties in each event. Our salience model is an extension of the Event-Indexing Situation Model (EISM) [12] and inspired by the Indexter model [1], offering the new capability of estimating the salience of a future event with respect to the current world state and an estimate of the player's knowledge. We presented this model in the context of a novel event generator, which uses the model to produce event descriptions at various levels of detail. By estimating

salience values and generating event descriptions on a per-property basis, our generator is capable of producing a wide range of descriptions for each generated event. We validated the accuracy and precision of our salience model in a pilot user study, finding that our model outperforms a uniform random predictor.

Acknowledgements

Some parts of this text appear in the first author's M.Sc. dissertation [4].

References

1. Cardona-Rivera, R.E., Cassell, B.A., Ware, S.G., Young, R.M.: Indexter: a computational model of the event-indexing situation model for characterizing narratives. In: Proceedings of the 3rd Workshop on Computational Models of Narrative. pp. 34–43 (2012)
2. Compton, K., Mateas, M.: Casual creators. In: Proceedings of the International Conference on Computational Creativity, ICCC (2015)
3. Cournoyer, F., Fortier, A.: Massive Crowd on Assassin's Creed Unity: AI Recycling. Presentation at the Game Developer's Conference (GDC 2015). GDC Vault, UBM Tech. (2015)
4. Flores, L.: Level of Detail Event Generation. M.Sc. dissertation. School of Computer Science, Reykjavik University (2017)
5. Magerko, B.: Evaluating preemptive story direction in the interactive drama architecture. Journal of Game Development 2(3), 25–52 (2007)
6. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Conference on Empirical Methods in Natural Language Processing. pp. 404–411. Association for Computational Linguistics (2004)
7. Paris, S., Gerdelan, A., O'Sullivan, C.: Ca-lod: Collision avoidance level of detail for scalable, controllable crowds. In: International Workshop on Motion in Games. pp. 13–28. Springer (2009)
8. Roth, M., Ben-David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, A., Leiser, N., Matias, Y., Merom, R.: Suggesting friends using the implicit social graph. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 233–242. ACM (2010)
9. Rowe, J.P., Lester, J.C.: Modeling user knowledge with dynamic bayesian networks in interactive narrative environments. In: 6th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. pp. 57–62 AAAI Press (2010)
10. Sunshine-Hill, B.: Perceptually driven simulation. Ph.D. thesis, University of Pennsylvania (2011)
11. Sunshine-Hill, B.: Managing Simulation Level-of-Detail with the LOD Trader. In: 6th International Conference on Motion in Games. pp. 13–18 ACM (2013)
12. Zwaan, R.A., Langston, M.C., Graesser, A.C.: The construction of situation models in narrative comprehension: An event-indexing model. Psychological science 6(5), 292–297 JSTOR (1995)
13. Zwaan, R.A., Radvansky, G.A.: Situation models in language comprehension and memory. Psychological bulletin 123(2) 162–185 APA (1998)