Learning Player Preferences for Fun Interactive Stories

David Thue and Vadim Bulitko

Department of Computing Science
University of Alberta, Canada
Once upon a time,
there was a muskrat.
Every day,

The muskrat sang
Tell a story

Until one day,

he died
Because of this, his brother took up singing.
Tell a story

and because of that,

everyone fled the forest

All blanks filled by participants of the AAAI 2010 Workshop on AI and Fun
Finally, the muskrat’s daughter started singing as well.
The End?
Introduction

Decisions in Storytelling

- Idea
- Actors
- Time
- Place
- Actions
- Reasons

What should happen next?

- What?
- Who?
- When?
- Where?
- How?
- Why?
## Introduction

### Decisions in Storytelling

<table>
<thead>
<tr>
<th>What was</th>
<th>Who</th>
<th>When was it</th>
<th>How was it</th>
<th>Why was that</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decided?</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Idea
- Actors
- Time
- Place
- Actions
- Reasons

<table>
<thead>
<tr>
<th>Author</th>
<th>During Telling</th>
<th>Funny?</th>
<th>Random Spontaneity</th>
<th>Clever?</th>
</tr>
</thead>
</table>

Funny? or Clever? during telling?
Tell me a story

Once upon a time,
there was a

Workshop on AI and Fun

All blanks filled by participants of the AAAI 2010 Workshop on AI and Fun
Tell me a story

Every day,

David took pictures
Tell me a story

Until one day,

his camera broke
Tell me a story

Because of this, he started drawing instead.
Tell me a story

and because of that, no one could understand his presentation.
Tell me a story

Finally,

David bought a new camera
The End
### Introduction

#### Decisions in Storytelling

<table>
<thead>
<tr>
<th>What was</th>
<th>Who</th>
<th>When was it</th>
<th>How was it</th>
<th>Why was that</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decided?</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idea</td>
<td>Author</td>
<td>During Telling</td>
<td>Audience Modelling &amp; Feedback</td>
<td>Random Spontaneity</td>
</tr>
<tr>
<td>Actors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Place</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reasons</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hypothesis

If you know about your audience, you can tell a better story.
When a story’s events are chosen based on feedback from its audience, the telling of that story is *Interactive*. 

*Definition*
How can we make this work?

We need a way to (computationally):

Learn about an audience

Make decisions based on audience feedback

Learning Player Preferences for Fun Interactive Stories
PaSSAGE
Player-Specific Stories via
Automatically Generated Events
Introduces Player Modelling to Interactive Storytelling to mimic human storytellers

Learns player preferences on-line by observing reactions to story events

Uses player preferences to dynamically choose subsequent story events

Suggests courses of action within events which fit the player’s preferred style of play
PaSSAGE
Framework Overview

- Observe player actions as they relate to in-game events
- Learn player preferences by measuring inclinations toward different styles of play
- From a library of encounters, choose an encounter to occur which allows the player to play in the modelled style
  - if possible, surreptitiously suggest the event-responding course of action that fits the modelled style (“hinting”)
- Repeat
**PaSSAGE**

Framework Operation

- **Encounter Manager**
  - Model Values
  - Encounters
  - (Decide)
  - Model Updates

- **Player Model**
  - (Relate)
  - (Interpret)
  - (Convey)
  - Audio/Video
  - Actions
  - (Gather)

- **Game Engine**

- **Player**
An Encounter is...

- A sequence of events that directly involve the player
- Each encounter has at least one course of action available to the player
- Each course of action is tailored to appeal to one or more types of player
## Robin Laws’ RPG Player Types

<table>
<thead>
<tr>
<th>Player Type</th>
<th>Enjoys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Gamer</td>
<td>Acquiring items and abilities</td>
</tr>
<tr>
<td>Fighter</td>
<td>Fighting</td>
</tr>
<tr>
<td>Tactician</td>
<td>Solving logical puzzles</td>
</tr>
<tr>
<td>Specialist</td>
<td>Exploiting their character’s special skills</td>
</tr>
<tr>
<td>Method Actor</td>
<td>Having their personality tested</td>
</tr>
<tr>
<td>Storyteller</td>
<td>Complex plots</td>
</tr>
<tr>
<td>Casual Gamer</td>
<td>Being with their friends</td>
</tr>
</tbody>
</table>

Larger values indicate stronger inclinations to play in the given style.
System wants an encounter to occur

Examine Encounter Library

Retrieve each encounter’s suitability data:

Example encounter suitability data:

<table>
<thead>
<tr>
<th>Style</th>
<th>Fighter</th>
<th>Tactician</th>
<th>Method Actor</th>
<th>Storyteller</th>
<th>Power Gamer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔️ ✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
</tbody>
</table>

- ✔️ - this encounter is good for players who prefer this style
- ✗ - this encounter is bad for players who prefer this style

Encounter suitability is defined by the suitability of its courses of action ("branches").
PaSSAGE
Encounter Selection

Choose Encounter
By Quality

Encounter Quality = \( \max_{\text{branches}} \left( \text{suitability}(\text{branch}) \cdot \text{PlayerModelData} \right) \)

Player Model

<table>
<thead>
<tr>
<th></th>
<th>Fighter</th>
<th>Tactician</th>
<th>Method Actor</th>
<th>Storyteller</th>
<th>Power Gamer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fighter</strong></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tactician</strong></td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Method Actor</strong></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Storyteller</strong></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Power Gamer</strong></td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Branch 1

<table>
<thead>
<tr>
<th></th>
<th>Fighter</th>
<th>Tactician</th>
<th>Method Actor</th>
<th>Storyteller</th>
<th>Power Gamer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fighter</strong></td>
<td>✔️ ✔️</td>
<td>✘</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tactician</strong></td>
<td>✘</td>
<td>✘ ✔️ ✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Method Actor</strong></td>
<td>✘</td>
<td>✘ ✔️ ✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Storyteller</strong></td>
<td>✘</td>
<td></td>
<td>✔️ ✔️</td>
<td></td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td><strong>Power Gamer</strong></td>
<td>✔️ ✔️</td>
<td>✔️ ✔️</td>
<td>✘ ✔️ ✔️</td>
<td>✔️ ✔️ ✔️ ✔️</td>
<td>✔️ ✔️ ✔️ ✔️</td>
</tr>
</tbody>
</table>

Branch 2

<table>
<thead>
<tr>
<th></th>
<th>Fighter</th>
<th>Tactician</th>
<th>Method Actor</th>
<th>Storyteller</th>
<th>Power Gamer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fighter</strong></td>
<td>✘</td>
<td>✘ ✔️ ✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tactician</strong></td>
<td>✘ ✔️ ✔️</td>
<td>✘ ✔️ ✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Method Actor</strong></td>
<td>✘ ✔️ ✔️</td>
<td></td>
<td></td>
<td>✔️ ✔️ ✔️</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td><strong>Storyteller</strong></td>
<td>✘ ✔️ ✔️</td>
<td>✔️ ✔️</td>
<td>✘ ✔️ ✔️</td>
<td></td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td><strong>Power Gamer</strong></td>
<td>✔️ ✔️</td>
<td></td>
<td></td>
<td></td>
<td>✔️ ✔️ ✔️</td>
</tr>
</tbody>
</table>
Choose Encounter By Quality

Encounter Quality = \[
\max_{\text{branches}} \left( \text{suitability}(\text{branch}) \cdot \text{PlayerModelData} \right)
\]

**Player Model**

<table>
<thead>
<tr>
<th>Player Model</th>
<th>Fighter</th>
<th>Tactician</th>
<th>Method Actor</th>
<th>Storyteller</th>
<th>Power Gamer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fighter</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tactician</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method Actor</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storyteller</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Gamer</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Branches**

<table>
<thead>
<tr>
<th></th>
<th>Branch 1</th>
<th>Branch 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fighter</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>Tactician</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>Method Actor</td>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>Storyteller</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Power Gamer</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Branch Quality**

- Branch 1: 43
- Branch 2: -21

**Encounter Quality**

- 43

Search can end early if a quality threshold is met.
The encounter’s “trigger function” is started, which watches for the world conditions necessary for the encounter to run.

Example Trigger Conditions:

i) There is an actor with the ‘child’ tag within 10m of the player.

ii) There is an actor with the ‘parent’ tag further than 50m from the actor found in i).
PaSSAGE

Encounter Specification: Triggers

The encounter’s “trigger function” is started, which watches for the world conditions necessary for the encounter to run.

All conditions satisfied?

Time-out exceeded?

Encounter terminates

No Yes

No

Cue Chosen Encounter

Run Chosen Encounter

45m 9m
The encounter’s “trigger function” is started, which watches for the world conditions necessary for the encounter to run.
PaSSAGE

Encounter Specification: Role Passing

From the set of actors currently not playing a role in another encounter, those that satisfy this encounter’s conditions are “checked out”

Example Check-out Process

i) The actor with ‘child’ tag within 10m of the player assumes the role of a lost child in search of his parent.

ii) The actor with ‘parent’ tag further than 50m from the lost child assumes the role of a parent in search of her child.

Assuming a role involves acquiring the dialog and behaviours that were authored for that role.
To maximize potential learning about the player, all branches should remain possible. To maximize player fun, surreptitiously direct them along the chosen branch.

**Troll**

Oh ho! All in good time, all in good time, young woman.

1. What do you want, then?
2. Let me by, troll, or I'll have your head!
To maximize potential learning about the player, all branches should remain possible. To maximize player fun, surreptitiously direct them along the chosen branch.

Hint Chosen Branch

Update Player Model

Hinting Fighter Branch

Troll
And what happensss if I refuse, tiny girl?

1. Let me by, troll, or I'll have your head!
2. What do you want, then?
By observing player responses in dialog along with their reactions to encounter events, knowledge of their player type can be refined.

---

**PaSSAGE**

Modelling the Player

- Update Player Model
- Repeat

**Method Actor**
- Fighter
- Method Actor
- Storyteller
- Tactician
- Power Gamer
PaSSAGE
User Study

LITTLE RED RIDING HOOD
PaSSAGE

User Study

- Test Group
  - player-specific stories
- Control Group
  - static stories, balanced with test group
- Deception
  - participants were unaware of potential adaptation
  - “evaluate student-created stories”

Fun_{Adaptive} > Fun_{Static}
PaSSAGE
User Study

- 76 participants
- 9 encounters
- 5 possible endings
- 3 decision points
- 8 static stories
- 1 adaptive system

Home ➔ Call to Adventure ➔ Crossing the Threshold ➔ Trials ➔ Ordeal

Start
Mercy
Recruit
Distract
D1
D2
Bounty
Recruit
Distract
Monsters
Traveller
Rescue from Defeat
Good
Evil
Neutral
Change of Heart

MONSTER
AER
Y
N
Y
N
WE
Troll?
Save Wizard?
Warn Wizard?
Y
N
WE
Y
N
WKT
WL
Player-Specific Stories are more Fun: 93% Confidence

In comparison to an average video game of similar length that you’ve played in the past (or your expectation of one), how enjoyable was your game experience?
What should we do now?

- What other aspects of our players’ experiences can we improve?
  - Agency is unique to interactive settings
  - Work with psychologists to learn more
- What set of principles should drive the creation of fun interactive stories?
  - Work with and co-train authors
AI systems can be treated as decision-making proxies for experience creators.

Decisions concerning story content should be delayed for as long as possible.

If you know about your audience, you can tell a better story.
PaSSAGE

Player-Specific Stories via Automatically Generated Events

www.playpassage.com