AIIDE 2011

Artificial Intelligence for Interactive Media and Games

Professor Charles Rich
Computer Science Department
rich@wpi.edu

Mission Statement

AIIDE is the definitive point of interaction between entertainment software developers interested in AI and academic and industrial AI researchers. Sponsored by the Association for the Advancement of Artificial Intelligence (AAAI), the conference is targeted at both the research and commercial communities, promoting AI research and practice in the context of interactive digital entertainment systems with an emphasis on commercial computer and video games.

By the Numbers

- 3 days
- about 130 attendees (guess: 85% academic, 15% industry)
- 17 papers presented (all academic)
- 6 technical sessions
- 5 invited talks (all from industry)
- 1 panel
- 19 poster/demos
- 2 workshops (immediately before main conference)
- 1 StarCraft AI competition (before conference)
Workshops

- Intelligent Narrative Technologies and Non-Player Character AI (2 days)
  - automatically generating stories
  - making believable virtual characters

- AI in the Game Development Process (1/2 day)
  - automated playtesting
  - partially automating game design and development

Technical Sessions

1. Planning in Games
2. Character Agents
3. Story
4. Learning
5. Authoring
6. Recognizing Player Activity

1. Planning in Games

Build Order Optimization in StarCraft

- David Churchill, U. Alberta
- Michael Buro, U. Alberta

Build Order Optimization in StarCraft

In the opening phase of RTS games, players usually don’t interact with each other because:
- their starting locations are spread over large maps
- player visibility is limited to small regions

The main sub-goals in this game phase are to:
- establish a sufficient income flow by producing workers
- quickly build structures that are prerequisites for other structures or can produce combat units
- build a minimal force for defense or early attack
- send out scouts to explore the terrain and search for enemy

The order in which units and structures are produced is called a build order
The choice of initial build order often decides the game outcome.

Build order optimization:
- \textit{given}: a desired set of units and structures
- \textit{compute}: shortest possible sequence of game actions to achieve that goal
- Corresponds to a standard \textit{operations research} problem: "constraint resource allocation with concurrent actions and makespace minimization"

Algorithm 2 Compare Build Order

\begin{algorithm}
\caption{Build Order Time Limit (Increment Time I)}
\begin{algorithmic}
\Procedure{CompareBuildOrder}{$B_0, B_1, T$}
\State $S \leftarrow \text{Initial StarCraft State}$
\State $\text{SearchPlan} \leftarrow \text{DFBKBICGetGoal}(S, 0, \infty, 0)$
\If{$\text{SearchPlan.eval} \leq \text{Goal}$}
\State \Return \text{MakeSpace}(\text{SearchPlan}) / \text{MakeSpace}(B)$
\Else
\State $inc \leftarrow inc + 1$
\State $\text{SearchPlan} \leftarrow \text{GetPlan}(S, 0, \infty, 0)$
\While{$inc \leq \text{MaxSpace}$}
\State $\text{IncPlan} \leftarrow \text{DFBKBICGetGoal}(S, inc, \infty, 0)$
\If{$\text{IncPlan.eval} \geq \text{Goal}$}
\State \Return \text{failure}
\Else
\State $\text{SearchPlan}.\text{append}(\text{IncPlan})$
\State $S \leftarrow \text{execute}(\text{IncPlan})$
\State $inc \leftarrow inc + 1$
\EndIf
\EndWhile
\State \Return \text{MakeSpace}(\text{SearchPlan}) / \text{MakeSpace}(B)$
\EndProcedure
\end{algorithmic}
\end{algorithm}

Algorithm produces plans in real-time which are comparable to professional StarCraft players.

Integrated into StarCraft playing agent which competed in 2010 AIIDE StarCraft AI Competition.
2. Character Agents

All the World’s a Stage:
Learning Character Models from Film

• Grace Lin and Marilyn Walker,
U. California Santa Cruz

Learning Character Models from Film

• What does “mimic the style” mean?

Learning Character Models from Film

• Long term goal: automatically generate character dialogue that mimics the style of existing characters

• Internet movie script database (IMDb)
  • 862 film scripts
  • 7,400 characters
  • 664,000 lines of dialogue
  • 9,599,000 words

Learning Character Models from Film

automatically extract these features from scripts using natural language processing techniques:

Set Description

Basic: number of sentences, sentences per turn, number of words, number of words per sentence

PRF: Word categories, Actor (hain, kill, prison), Social processes (talk, en, friend), Friends (pal, buddy, coworker), Causation (because, know, sure), Disempower (should, would, could), Assents (yes, OK, onus), Interrogative (maybe, perhaps, gonna), etc.

Dialogue Acts: Accept, Bye, Clarify, Complain, Emotion, Emphasis, Great, No, Answer, Belief, Statement, Why, Question, Yes, Answer, Yes-No-Question, Other

First Dialogue Act: Ant. Same as IA but only look at first sentence of each turn.

Pragmatic Markers: Word counts and ratios, plus word category: concrete, positive, loving, position, pro-aggression, pro-offense, pro-phantic, pause, p-pauses, p-connection, p-continue, p-justify, p-contrast, p-counterpart, p-negation, p-tense, p-assert, p-relate

Politeness: overall politeness, politeness of sentences, politeness for concession

Merge Ratio: merging of subject and verb of two propositions

Tag Question Ratio: number of sentences with tag questions out of all sentences

Average Correct Word Length: content words are nouns, adjectives, adverbs, and verbs, average words’ length

Verb Bi-gram: average sentiment values of verbs

Passive Sentence Ratio: number of passive sentences out of all sentences
Learning Character Models from Film

1. Automatically extract features from existing scripts using natural language processing

2. Use machine learning to learn a “character model”

3. Use the character model to control parameters of an automatic dialogue generator

Character model for Annie:
- many disfluencies, such as um and uh
- says yes and yea a lot, especially at the beginning of her utterances
- produces short sentences, but talks a lot
- uses a lot of tag questions
- does not use long words
- uses really, sort of and I think a lot
- etc.

3. Story

The SAM Algorithm for Analogy-Based Story Generation
- Santiago Ontanon, Spanish Council for Scientific Research
- Jichen Zhu, U. of Central Florida
Analogy-Based Story Generation

**Goal:** Automatically generate a new story from given analogy source and target stories

**Challenges:**
- how to represent stories in a computer
- the analogy algorithm (uses previous work by Forbus & Gentner)

---

**Analogy Source Story**

Ales remembered the garage in which he had his first oil change, it was all red. His owners said he was rusty, and forced him to change his oil; he was a fool to accept. Ales felt very awkward afterwards, and decided that he would have to be really rusty before the next time he gets an oil change. He wondered why no one ever complained about oil changes.

---

**Target Story**

One day, Ales was walking in an alley, when he saw a cat in front of him. Ales hesitated about what to do with the cat since he was late for work. Ales played with the cat.
New Story

One day, Ales was walking outside, when he saw a cat in front of him. Ales hesitated about what to do with the cat since he was late for work. Ales played with the cat. Ales felt very awkward afterwards, and decided that Ales would have to be really rusty before the next time Ales played with the cat. Ales wondered why no one ever complained about that.

Conclusion

- not ready for prime time 😞
- authors working on improvements

4. Learning

Learning Policies for First Person Shooter Games Using Inverse Reinforcement Learning

- Bulent Tastan and Gita Sukthankar, U. of Central Florida

Learning Policies for FPS Games

- A policy is a mapping from situations to actions
  - situations described by features
- Goal: To learn policies from examples of human play
  - “learning from demonstration”
Learning Policies for FPS Games

- Given a set of (learned) reward/punishment rules, you can easily compute a policy by always choosing the action with the greatest reward.
- Used to build a bot for Unreal Tournament based on human players' behaviors.

---

Learning Policies for FPS Games

- Experimental Evaluation
  - Goal is "human-ness" of play, not optimal play (no fun in game to always be beat 🎉)
  - Trained system on about 45 minutes of human play
  - 13 experimental participants played two games
    - One against bot trained with inverse reinforcement learning
    - One against a hand-coded bot
  - Participants told they were playing against one human and one bot and supposed to figure out which is which!
  - Participants significantly chose the IRL bot as the human.

---

5. Authoring

- AIPaint: A Sketch-Based Behavior Tree Authoring Tool
  - David Becroft, Adrian Mejia, Jesse Basset, Charles Rich and Candace Sidner, Worcester Polytechnic Institute
6. Recognizing Player Activity

Goal Recognition with Markov Logic Networks for Player-Adaptive Games

- Eun Ha, Jonathan Rowe, Bradford Mott and James Lester, North Carolina State U.

Goal Recognition with Markov Logic Networks

- **Goal Recognition**: inferring users’ goals from sequences of observed actions
- **Application**: automatically adjust (adapt) play in Crystal Island serious game
- **Technique**: Markov logic networks

**Crystal Island**
- educational game for 8th-grade microbiology
- overall goal is to identify source of mysterious illness
- open, exploratory environment
- could be improved by a little guidance
Goal Recognition w. Markov Logic Networks

- **Markov logic network (MLN)**
  - combines first-order logic with probabilistic graphical models
  - instead of true/false values, logical constraints have *weights*
  - used to infer *hidden* predicates (goals) from *observed* predicates (actions)

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action(a, t)</td>
<td>Player takes an action a at time t</td>
</tr>
<tr>
<td>loc(t, l)</td>
<td>Player is at a location l at time t</td>
</tr>
<tr>
<td>state(t, s)</td>
<td>The narrative state at time t is s</td>
</tr>
<tr>
<td>goal(g, t)</td>
<td>Player pursues a goal g at time t</td>
</tr>
</tbody>
</table>

**Evaluation**
- collected data from 137 middle school students playing the game for approx 1 hour each
- data manually labeled and used to train MLN

<table>
<thead>
<tr>
<th>Total Number of Observed Player Actions</th>
<th>77192</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Goals Achieved</td>
<td>890</td>
</tr>
<tr>
<td>Average Number of Player Actions per Goal</td>
<td>86.8</td>
</tr>
</tbody>
</table>

- trained MLN compared to other goal recognition algorithms (including unigram and bigram)

**Conclusion**
- technique does a better job of goal recognition than any other existing system
- next step is to see how knowing players’ goals will help guide the game.

---

**Invited Talks**

- **The Evolution of RTS AI**
  - Bob Fitch, Blizzard Entertainment
- **Bringing Physical Characters to Life---Lessons and Challenges from Disney**
  - Akhil Madhani, Walt Disney Imagineering Research and Development
- **Darksore AI Post Mortem**
  - Dan Kline and Lauren McHugh, EA/Maxis
* Invited Talks [cont’d]

- Creating the Enemies of *Dead Space*
  - Louis Gascoigne, Visceral Games, an Electronic Arts Studio

- Social Games and the Role of Simulation in a Social World
  - Robert Zubek, Zynga

* Panel

- Player Modeling: Games, Data and Human Behavior
  - David Roberts, North Carolina State U.
  - Nick Yee, Palo Alto Research Center
  - Mike Carr, Novel Interactive
  - Carrie Heeter Michigan State U.

Questions? Comments?

- P.S. The other big yearly game AI confab is the *AI Summit* at GDC
  - March 5-6, 2012, San Francisco
  - organized by the AI Game Programmers Guild (http://gameai.com)
  - approx 85% industry, 15% academic