Sabre
A Narrative Planner Supporting Intention and Deep Theory of Mind

Stephen G. Ware
Cory Siler
Narrative Planning

A single decision maker creates the appearance of a multi-agent system.
Intentions and Beliefs

C: Classical
Actions are actually possible.
Intentions and Beliefs

C: Classical
Actions are actually possible.

I: Intention
Actions can achieve agent’s goal.

• Riedl and Young, “Narrative planning: balancing plot and character,” in JAIR 2010
• Teutenberg and Porteous, “Efficient intent-based narrative generation...,” in AAMAS 2013
• Ware and Young, “Glaive: a state-space narrative planner...,” in AIIDE 2014
Intentions and Beliefs

C: Classical
Actions are actually possible.

I: Intention
Actions can increase agent's utility.
Intentions and Beliefs

C: Classical
Actions are actually possible.

B: Belief
Agent believes the actions are possible.

I: Intention
Actions can increase agent’s utility.

- Eger and Martens, “Character beliefs in story generation,” INT 2017
- Thorne and Young, “Generating stories ... by modeling false character beliefs,” in INT 2017
- Shirvani, Ware, and Farrell, “A possible worlds model of belief...,” in AIIDE 2017
Intentions and Beliefs

- Shirvani, Farrell, and Ware, “Combining intentionality and belief...,” in AIIDE 2018
Syntax and Features
Fluents

\[ \text{at}(\text{Tom}) = \]

Fluents

\[ at(Tom) = \textit{Cottage} \]

Fluents

\[ at(Tom) = Cottage \]

\[ path(Cottage, Market) = T \]
Fluents

\[\text{at}(Tom) = Cottage\]

\[\text{path}(Cottage, Market) = T\]

\[\text{wealth}(Merchant) = 3\]
Fluents

\[
at(Tom) = \text{Cottage} \\
\text{path(Cottage, Market)} = \top \\
\text{wealth(merchant)} = 3 \\
\text{believes(Tom, wealth(merchant))} = 2
\]
Fluents

\( at(Tom) = Cottage \)

\( path(Cottage, Market) = T \)

\( wealth(Merchant) = 3 \)

\( believes(Tom, wealth(Merchant)) = 2 \)

\( believes(Merchant, believes(Tom, wealth(Merchant))) = 3 \)
Theory of Mind

• Arbitrarily deep
  what $x$ believes $y$ believes $z$ believes...

• No uncertainty
  Everyone commits to beliefs, which can be wrong.
Other Syntactical Features

• Negation
• Disjunction
• Conditional Effects
• First Order Quantifiers
buy(Tom, Potion, Merchant, Market)
Actions

\[ a: \text{buy}(Tom, Potion, Merchant, Market) \]
Actions

\[ a: \text{buy}(\text{Tom, Potion, Merchant, Market}) \]

\[ \text{PRE}(a): \]
Actions

\[ a: \text{buy}(Tom, Potion, Merchant, Market) \]

\[ \text{PRE}(a): \text{at}(Tom) = Market \]
Actions

\[ a: \text{buy}(\text{Tom, Potion, Merchant, Market}) \]

\[ \text{PRE}(a): \text{at}(\text{Tom}) = \text{Market} \land \text{at}(\text{Merchant}) = \text{Market} \]
Actions

\[ a: \text{buy}(\text{Tom}, \text{Potion}, \text{Merchant}, \text{Market}) \]

\[ \text{PRE}(a): \text{at}(\text{Tom}) = \text{Market} \land \text{at}(\text{Merchant}) = \text{Market} \land \]
\[ \text{at}(\text{Potion}) = \text{Merchant} \]
Actions

\[ a: \text{buy}(Tom, Potion, Merchant, Market) \]

\[
\text{PRE}(a): \; at(Tom) = Market \land at(Merchant) = Market \land \\
\text{at(Potion)} = Merchant \land \text{wealth}(Tom) \geq 1
\]
Actions

\( a: \ buy(Tom, Potion, Merchant, Market) \)

\( \text{PRE}(a): \ at(Tom) = Market \land at(Merchant) = Market \land \)
\( at(Potion) = Merchant \land \text{wealth}(Tom) \geq 1 \)

\( \text{EFF}(a): \)
**Actions**

\[ a: \text{buy}(\text{Tom}, \text{Potion}, \text{Merchant}, \text{Market}) \]

\[ \text{PRE}(a): \quad \text{at}(\text{Tom}) = \text{Market} \land \text{at}(\text{Merchant}) = \text{Market} \land \text{at}(\text{Potion}) = \text{Merchant} \land \text{wealth}(\text{Tom}) \geq 1 \]

\[ \text{EFF}(a): \quad \text{at}(\text{Potion}) = \text{Tom} \]
**Actions**

\[ a: \text{buy}(Tom, Potion, Merchant, Market) \]

\[ \text{PRE}(a): \text{at}(Tom) = \text{Market} \land \text{at}(Merchant) = \text{Market} \land \]  
\[ \text{at}(Potion) = \text{Merchant} \land \text{wealth}(Tom) \geq 1 \]

\[ \text{EFF}(a): \text{at}(Potion) = Tom \land \text{wealth}(Merchant) += 1 \]
**Actions**

\[ a: \text{buy}(Tom, Potion, Merchant, Market) \]

\[ \text{PRE}(a): \text{at}(Tom) = Market \land \text{at}(Merchant) = Market \land \]
\[ \text{at}(Potion) = Merchant \land \text{wealth}(Tom) \geq 1 \]

\[ \text{EFF}(a): \text{at}(Potion) = Tom \land \text{wealth}(Merchant) += 1 \land \]
\[ \text{wealth}(Tom) -= 1 \]
Actions

\[ a: \text{buy}(Tom, Potion, Merchant, Market) \]

\[ \text{PRE}(a): \quad at(Tom) = Market \land at(Merchant) = Market \land \]
\[ at(Potion) = Merchant \land \text{wealth}(Tom) \geq 1 \]

\[ \text{EFF}(a): \quad at(Potion) = Tom \land \text{wealth}(Merchant) += 1 \land \]
\[ \text{wealth}(Tom) -= 1 \]

\[ \text{CON}(a): \]
**Actions**

\[ a: \text{buy}(\text{Tom}, \text{Potion}, \text{Merchant}, \text{Market}) \]

\[ \text{PRE}(a): \text{at}(\text{Tom}) = \text{Market} \land \text{at}(\text{Merchant}) = \text{Market} \land \]
\[ \text{at}(\text{Potion}) = \text{Merchant} \land \text{wealth}(\text{Tom}) \geq 1 \]

\[ \text{EFF}(a): \text{at}(\text{Potion}) = \text{Tom} \land \text{wealth}(\text{Merchant}) += 1 \land \]
\[ \text{wealth}(\text{Tom}) -= 1 \]

\[ \text{CON}(a): \{\text{Tom}, \text{Merchant}\} \]
Actions

\( a: \) \( buy(Tom, Potion, Merchant, Market) \)

\( \text{PRE}(a): \) \( at(Tom) = Market \land at(Merchant) = Market \land \)
\( \quad at(Potion) = Merchant \land \text{wealth}(Tom) \geq 1 \)

\( \text{EFF}(a): \) \( at(Potion) = Tom \land \text{wealth}(Merchant) += 1 \land \)
\( \quad \text{wealth}(Tom) -= 1 \)

\( \text{CON}(a): \) \( \{ Tom, Merchant \} \)

\( \text{OBS}(a, c): \)
Actions

\[ a: \text{buy}(Tom, Potion, Merchant, Market) \]

\[ \text{PRE}(a): \quad at(Tom) = Market \land at(Merchant) = Market \land \]
\[ \quad at(Potion) = Merchant \land \text{wealth}(Tom) \geq 1 \]

\[ \text{EFF}(a): \quad at(Potion) = Tom \land \text{wealth}(Merchant) += 1 \land \]
\[ \quad \text{wealth}(Tom) -= 1 \]

\[ \text{CON}(a): \quad \{Tom, Merchant\} \]

\[ \text{OBS}(a, c): \quad at(c) = Market \]
Triggers

\[ t: \text{see}(\text{Tom}, \text{Merchant}, \text{Market}) \]

\text{PRE}(t):

\text{EFF}(t):
Triggers

\[ t: \text{see}(Tom, Merchant, Market) \]

\[ \text{PRE}(t): \text{at}(Tom) = Market \]

\[ \text{EFF}(t): \]
Triggers

\[ t: \text{see}(Tom, Merchant, Market) \]
\[ \text{PRE}(t): \text{at}(Tom) = Market \land \text{at}(Merchant) = Market \]
\[ \text{EFF}(t): \]
Triggers

\[ t: \) see(Tom, Merchant, Market) \]

\[ \text{PRE}(t): \) at(Tom) = Market \land at(Merchant) = Market \land \]
\[ \text{believes}(Tom, at(Merchant)) \neq Market \]

\[ \text{EFF}(t): \]
Triggers

\( t: \text{see}(\text{Tom}, \text{Merchant}, \text{Market}) \)

\( \text{PRE}(t): \ at(\text{Tom}) = \text{Market} \land at(\text{Merchant}) = \text{Market} \land \)
\( \quad \text{believes}(\text{Tom}, at(\text{Merchant})) \neq \text{Market} \)

\( \text{EFF}(t): \ \text{believes}(\text{Tom}, at(\text{Merchant})) = \text{Market} \)
Pre-Processing

• Make action and trigger results explicit
• Detect and remove immutable fluents
• Detect and remove impossible actions and triggers
Results of an Event

After Tom buys the potion from the merchant...
• Tom has the potion.
• Tom knows he has the potion.
• The merchant knows Tom has the potion.
• Tom know that the merchant knows that he has the potion.
• ... and so on.
Example Trigger: Two-Way Paths

\[ t: \text{add\_path}(y, x) \]

\[ \text{PRE}(t) \text{ path}(x, y) = \top \land \text{path}(y, x) = \bot \]

\[ \text{EFF}(t): \text{path}(y, x) = \top \]
Example Trigger: Two-Way Paths

\[ t: \text{add\_path}(\text{Market, Cottage}) \]

\[ \text{PRE}(t): \, \text{path}(\text{Cottage, Market}) = T \land \]
\[ \text{path}(\text{Market, Cottage}) = \bot \]

\[ \text{EFF}(t): \, \text{path}(\text{Market, Cottage}) = T \]
Example Action: Walk

\[ a: \text{walk}(Tom, Market, Cottage) \]

\[ \text{PRE}(a): \ at(Tom) = Market \land \]
\[ \quad \text{path}(Market, Cottage) = T \]

\[ \text{EFF}(a): \ at(Tom) = Cottage \]

\[ \text{CON}(a): \ \{Tom\} \]

\[ \text{OBS}(a, c): \ at(c) = Market \lor at(c) = Cottage \]
Example Action: Walk

\[ a: \text{walk}(\text{Tom}, \text{Market}, \text{Cottage}) \]

\[ \text{PRE}(a): \, at(\text{Tom}) = \text{Market} \wedge \]

\[ \text{path}(\text{Market}, \text{Cottage}) = \top \]

\[ \text{EFF}(a): \, at(\text{Tom}) = \text{Cottage} \]

\[ \text{CON}(a): \, \{\text{Tom}\} \]

\[ \text{OBS}(a, c): \, at(c) = \text{Market} \lor at(c) = \text{Cottage} \]
Example Action: Walk

\( a: \text{walk}(\text{Tom}, \text{Market}, \text{Cottage}) \)

\( \text{PRE}(a): \text{at}(\text{Tom}) = \text{Market} \)

\( \text{EFF}(a): \text{at}(\text{Tom}) = \text{Cottage} \)

\( \text{CON}(a): \{\text{Tom}\} \)

\( \text{OBS}(a, c): \text{at}(c) = \text{Market} \lor \text{at}(c) = \text{Cottage} \)
Algorithm 1 The Sabre algorithm

1: Let $\mathcal{A}$ be the set of all actions defined in the domain.
2: $\text{SABRE}(c_{\text{author}}, s_0, \emptyset, s_0)$
3: function $\text{SABRE}(c, r, \pi, s)$
4: Input: character $c$, start state $r$, plan $\pi$, current state $s$
5: if $u(c, s) > u(c, r)$ and $\pi$ is non-redundant then
6: return $\pi$
7: choose an action $a \in \mathcal{A}$ such that $s \models \text{PRE}(a)$.
8: for all $c' \in \text{CON}(a)$ such that $c' \neq c$ do
9: let state $b = \alpha(a, \beta(c', s))$.
10: if $b$ is undefined then return failure.
11: else if $\text{SABRE}(c', b, \emptyset, b)$ fails then return failure.
12: return $\text{SABRE}(c, r, \pi \cup a, \alpha(a, s))$
\[ \alpha(s, s_0) = s_1 \]

Tom walks to the market.
\[ \beta(s_0, s_0) = s_2 \]

Tom walks to the market.
Tom walks to the market.

I walk to the market.
Tom walks to the market.

I walk to the market.

I buy the potion from the merchant.
Tom walks to the market.

I walk to the market.

I buy the potion from the merchant.

Tom buys the potion from me.
Tom walks to the market.

I walk to the market.

I buy the potion from the merchant.
Tom walks to the market.

I walk to the market.

I buy the potion from the merchant.

I walk home.
Tom walks to the market.

I walk to the market.

I buy the potion from the merchant.

I walk home.
Tom walks to the market.

I walk to the market.

I buy the potion from the merchant.

I walk home.
Tom walks to the market.

Tom buys the potion from the merchant.
$s_0$  Tom walks to the market.  $s_1$  Tom buys the potion from the merchant.  $s_3$  
$s_9$  Tom buys the potion from me.  $s_0$
Tom walks to the market.

Tom buys the potion from the merchant.

NIL
Tom walks to the market.

Tom buys the potion from the merchant.

I buy the potion from the merchant.

I walk home.
Tom walks to the market. 

Tom buys the potion from the merchant. 

I buy the potion from the merchant. 

I walk home.
Tom walks to the market.

Tom buys the potion from the merchant.
Tom walks to the market.

Tom buys the potion from the merchant.

Tom walks home.
Tom walks to the market. 
Tom buys the potion from the merchant. 
Tom walks home. 
I walk home.
Tom walks to the market.

Tom buys the potion from the merchant.

Tom walks home.
Evaluation
### Comparing Sabre to Other Planners

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Intentions</th>
<th>Beliefs</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sabre</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>✗</td>
</tr>
</tbody>
</table>
Comparing Sabre to Other Planners

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Intentions</th>
<th>Beliefs</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sabre</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Glaive</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

- Riedl and Young, “Narrative planning: balancing plot and character,” in JAIR 2010
- Ware and Young, “CPOCL: a narrative planner supporting conflict,” in AIIDE 2011
- Teutenberg and Porteous, “Efficient intent-based narrative generation...,” in AAMAS 2013
- Ware and Young, “Glaive: a state-space narrative planner...,” in AIIDE 2014
Comparing Sabre to Other Planners

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Intentions</th>
<th>Beliefs</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sabre</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Glaive</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HeadSpace</td>
<td>✓</td>
<td>✗</td>
<td>~✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

- Thorne and Young, “Generating stories ... by modeling false character beliefs,” in INT 2017
Comparing Sabre to Other Planners

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Intentions</th>
<th>Beliefs</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sabre</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>Glaive</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HeadSpace</td>
<td>✔</td>
<td>✗</td>
<td>~✔</td>
<td>✗</td>
</tr>
<tr>
<td>IMPRACTical</td>
<td>✔</td>
<td>✔</td>
<td>~✔</td>
<td>✗</td>
</tr>
</tbody>
</table>

- Teutenberg and Porteous, “Incorporating global and local knowledge...,” in AAMAS 2015
### Comparing Sabre to Other Planners

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Intentions</th>
<th>Beliefs</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sabre</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Glaive</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HeadSpace</td>
<td>✓</td>
<td>✗</td>
<td>~✓</td>
<td>✗</td>
</tr>
<tr>
<td>IMPRACTical</td>
<td>✓</td>
<td>✓</td>
<td>~✓</td>
<td>✗</td>
</tr>
<tr>
<td>Thespian</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- Ryan, Summerville, Mateas, and Wardrip-Fruin, “Toward characters who observe...,” in EXAG 2015
- Si and Marsella, “Encoding Theory of Mind in character design...,” in AHCI 2014
## Comparing Sabre to Other Planners

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Intentions</th>
<th>Beliefs</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sabre</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Glaive</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HeadSpace</td>
<td>✓</td>
<td>✗</td>
<td>~ ✓</td>
<td>✗</td>
</tr>
<tr>
<td>IMPRACTical</td>
<td>✓</td>
<td>✓</td>
<td>~ ✓</td>
<td>✗</td>
</tr>
<tr>
<td>Thespian</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ostari</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- Eger and Martens, “Practical specification of belief manipulation in games,” in AIIDE 2017
Test Problems

• *Raiders*
• *Space*

• Ware and Young, “Glaive: a state-space narrative planner...,” in AIIDE 2014
Test Problems

• Raiders
• Space
• Treasure
• Lovers
• Hubris

• Farrell and Ware, “Narrative planning for belief and intention recognition,” in AIIDE 2020
• Shirvani, Farrell, and Ware, “Combining intentionality and belief ....,” in AIIDE 2018
• Christensen, Nelson, and Cardona-Rivera, “Using domain compilation to add belief ....,” in AIIDE 2020
Test Problems

- Raiders
- Space
- Treasure
- Lovers
- Hubris
- BearBirdJr

- Meehan, “TALE-SPIN, an interactive program that writes stories,” in AAAI 1977
Test Problems

- Raiders
- Space
- Treasure
- Lovers
- Hubris
- BearBirdJr
- Grandma

- Ware, Garcia, Shirvani, and Farrell, “Multi-agent experience management …,” in AIIDE 2019
## Results

<table>
<thead>
<tr>
<th>Domain</th>
<th>Nodes Generated</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raiders</td>
<td>17,815</td>
<td>1.4 s</td>
</tr>
<tr>
<td>Space</td>
<td>192</td>
<td>6 ms</td>
</tr>
<tr>
<td>Treasure</td>
<td>288</td>
<td>1 ms</td>
</tr>
<tr>
<td>Lovers</td>
<td>5,198,414</td>
<td>40.3 m</td>
</tr>
<tr>
<td>Hubris</td>
<td>831</td>
<td>47 ms</td>
</tr>
<tr>
<td>BearBirdJr</td>
<td>34,084,068</td>
<td>14.0 m</td>
</tr>
<tr>
<td>Grandma</td>
<td>105,178,466</td>
<td>6.2 h</td>
</tr>
</tbody>
</table>
Conclusion
Limitations

- No true uncertainty
- $h^+$ heuristic often performs poorly\(^1\)

Future Work

• More search methods

```
Algorithm 1 The Sabre algorithm

1: Let $A$ be the set of all actions defined in the domain.
2: SABRE($c_{\text{auth}}, s_0, \emptyset, s_0$)
3: function SABRE($c, r, \pi, s$)
4: Input: character $c$, start state $r$, plan $\pi$, current state $s$
5: if $u(c, s) > u(c, r)$ and $\pi$ is non-redundant then
6: return $\pi$
7: Choose an action $a \in A$ such that $s \models \text{PREF}(a)$
8: for all $c' \in \text{CON}(a)$ such that $c' \neq c$ do
9: Let state $b = \alpha(a, \beta(c', s))$.
10: if $b$ is undefined then return failure.
11: else if SABRE($c'$, $b$, $\emptyset$, $b$) fails then return failure.
12: return SABRE($c$, $r$, $\pi \cup a$, $\alpha(a, s)$)
```
Future Work

- More search methods
Future Work

• More search methods
• Better heuristics
• Agent emotions and personalities\(^1\)

1. Shirvani and Ware, “A formalization of emotional planning for strong-story systems,” in AIIDE 2020