Probabilistic Behavioural State Machines

Peter Novák

Clausthal University of Technology, Germany

Tuesday, May 12th, 2009,
ProMAS 2009, Budapest, Hungary

Motivation

Underspecification in agent-oriented programming:

- KR granularity
- environment $\rightsquigarrow$ incomplete information

non-determinism

Practical experience with Behavioural State Machines:

some behaviours (plans) should be tried more often than others

finer grained control of non-deterministic action selection!

$\rightsquigarrow$ probabilistic extension of the BSM framework!

Motivation

Underspecification in agent-oriented programming:

- KR granularity
- environment $\rightsquigarrow$ incomplete information

⬇

non-determinism

Practical experience with Behavioural State Machines:

some behaviours (plans) should be tried more often than others

⬇

finer grained control of non-deterministic action selection!

$\rightsquigarrow$ probabilistic extension of the BSM framework!

Agenda

Behavioural State Machines/Jazzyk

Behavioural State Machines

A programming framework with clear separation between *knowledge representation* and agent's *behaviours*.

BSM framework provides:

- clear *semantics:* Gurevich's Abstract State Machines
- *modularity:*
  - **vertical**: heterogeneous knowledge bases
    - ⇝ easy *integration* with external/legacy systems
  - **horizontal**: structured source code, re-usability

BSM: the core concepts

KR module $\mathcal{M} = (\mathcal{S}, \mathcal{L}, \mathcal{Q}, \mathcal{U})$

- $\mathcal{S}$ - a set of states
- $\mathcal{L}$ - a KR language
- $\mathcal{Q}$ - a set of query operators $\models: \mathcal{S} \times \mathcal{L} \rightarrow \{\top, \bot\}$
- $\mathcal{U}$ - set of update operators $\oplus : \mathcal{S} \times \mathcal{L} \rightarrow \mathcal{S}$

mental state transformers τ:

- *conditional:* $\models_i \varphi \longrightarrow \oplus_j \psi$
- *compositionality:* choice $|$, sequence $\circ$, block $\{\ldots\}$
  - associativity: $\tau_1|\tau_2|\cdots|\tau_n$, $\tau_1 \circ \tau_2 \circ \cdots \circ \tau_n$

/* PICK an item behaviour */
when $\models_{goal}$ [{ task(pick(X)) }] **and** $\models_{bel}$ [{ see(X) }] **then** {
  when $\models_{bel}$ [{ dir(X, Angle) }] **then** $\oslash_{env}$ [{ turn Angle }] |
  when $\models_{bel}$ [{ dir(X,'ahead'), dist(X,Dist) }] **then** {
    $\oslash_{env}$ [{ move forward Dist }] $\circ$
    $\oplus_{bel}$ [{ holds(X) }]
  }
}

BSM: the core concepts

KR module $\mathcal{M} = (\mathcal{S}, \mathcal{L}, \mathcal{Q}, \mathcal{U})$

- $\mathcal{S}$ - a set of states
- $\mathcal{L}$ - a KR language
- $\mathcal{Q}$ - a set of query operators $\models: \mathcal{S} \times \mathcal{L} \rightarrow \{\top, \bot\}$
- $\mathcal{U}$ - set of update operators $\oplus : \mathcal{S} \times \mathcal{L} \rightarrow \mathcal{S}$

mental state transformers τ:

- *conditional:* $\models_i \varphi \longrightarrow \oplus_j \psi$
- *compositionality:* choice $|$, sequence $\circ$, block $\{\ldots\}$
  - associativity: $\tau_1 | \tau_2 | \cdots | \tau_n$, $\tau_1 \circ \tau_2 \circ \cdots \circ \tau_n$

```
/* PICK an item behaviour */
when |=_goal [{ task(pick(X)) }] and |=_bel [{ see(X) }] then {
        when |=_bel [{ dir(X, Angle) }] then ⊘_env [{ turn Angle }] |
        when |=_bel [{ dir(X,'ahead'), dist(X,Dist) }] then {
                ⊘_env [{ move forward Dist }] ∘
                ⊕_bel [{ holds(X) }]
        }
}
```

Semantics: $\mathcal{A} = (\mathcal{M}_1, \ldots, \mathcal{M}_n, \mathcal{P})$

transition system over states $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ induced by updates $\oplus\psi$

Jazzyk BSM semantics (operational view)

A sequence $\sigma_1, \ldots, \sigma_i, \ldots$, s.t. $\sigma_i \rightarrow \sigma_{i+1}$, is a trace of BSM.
An agent system (BSM), is characterized by a set of all traces.

Jazzyk BSM semantics (functional view)

$\tau \rightsquigarrow \quad \mathfrak{f}_\tau : \sigma \mapsto$ *enabled* updates in states

subprograms as state transforming functions

Semantics: $\mathcal{A} = (\mathcal{M}_1, \ldots, \mathcal{M}_n, \mathcal{P})$

transition system over states $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ induced by updates $\oplus\psi$

Jazzyk BSM semantics (operational view)

A sequence $\sigma_1, \ldots, \sigma_i, \ldots$, s.t. $\sigma_i \rightarrow \sigma_{i+1}$, is a trace of BSM.
An agent system (BSM), is characterized by a set of all traces.

Jazzyk BSM semantics (functional view)

$\tau \rightsquigarrow \quad \mathfrak{f}_\tau : \sigma \mapsto$ *enabled* updates in states

subprograms as state transforming functions

Probabilistic Behaviour State Machines/Jazzyk(P)

Probabilistic extension of BSM

P-BSM $\rightsquigarrow \mathcal{A} = (\mathcal{M}_1, \ldots, \mathcal{M}_n, \mathcal{P}, \Pi)$:

- $\mathcal{A} = (\mathcal{M}_1, \ldots, \mathcal{M}_n, \mathcal{P})$ is a BSM
- $\Pi : \tau \mapsto P_\tau$ assigns to each choice mst $\tau_1 | \cdots | \tau_k \in \mathcal{P}$ a probability function $P_\tau : \tau_i \mapsto [0, 1]$, s.t. $\sum_{i=1}^{k} P(\tau_i) = 1$

Jazzyk(P) example (Jazzbot)

when $\models_{bel}$ [{ threatened }] **then** {
/∗ ∗∗∗Emergency modus operandi∗∗∗ ∗/

/∗ Detect the enemy's position ∗/
0.7 : **when** $\models_{bel}$ [{ attacker(Id) }] **and** $\models_{env}$ [{ eye see Id player Pos }]
then $\oplus_{map}$ [{ positions[Id] = Pos }] ;

/∗ Check the camera sensor ∗/
0.2 : **when** $\models_{env}$ [{ eye see Id Type Pos }] **then** {
$\oplus_{bel}$ [{ see(Id, Type) }] ,
$\oplus_{map}$ [{ objects[Pos].addIfNotPresent(Id) }]
}

/∗ Check the body health sensor ∗/
when $\models_{env}$ [{ body health X }] **then** $\oplus_{bel}$ [{ health(X). }] ;
}

P-BSM semantics: $\mathcal{A} = (\mathcal{M}_1, \ldots, \mathcal{M}_n, \mathcal{P}, \Pi)$

transition system over states $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ induced by labelled updates p: $\oplus\,\psi$

$$\frac{\top}{\mathit{yields}_p(\mathbf{skip},\sigma,1{:}\mathbf{skip})} \qquad \frac{\top}{\mathit{yields}_p(\oslash\psi,\sigma,1{:}(\oslash,\psi))} \qquad \text{(primitive)}$$

$$\frac{\mathit{yields}_p(\tau,\sigma,p{:}\rho),\ \sigma\models\phi}{\mathit{yields}_p(\phi\longrightarrow\tau,\sigma,p{:}\rho)} \qquad \frac{\mathit{yields}_p(\tau,\sigma,\theta,p{:}\rho),\ \sigma\not\models\phi}{\mathit{yields}_p(\phi\longrightarrow\tau_p,\sigma,1{:}\mathbf{skip})} \qquad \text{(conditional)}$$

$$\frac{\tau=\tau_1|\cdots|\tau_k,\ \Pi(\tau)=P_\tau,\ \forall 1\leq i\leq k{:}\ \mathit{yields}_p(\tau_i,\sigma,p_i{:}\rho_i)}{\forall 1\leq i\leq k{:}\ \mathit{yields}_p(\tau,\sigma,P_\tau(\tau_i)\cdot p_i{:}\rho_i)} \qquad \text{(choice)}$$

$$\frac{\tau=\tau_1\circ\cdots\circ\tau_k,\ \forall 1\leq i\leq k{:}\ \mathit{yields}_p(\tau_i,\sigma_i,p_i{:}\rho_i)\wedge\sigma_{i+1}=\sigma_i\bigoplus\rho_i}{\mathit{yields}(\tau,\sigma_1,\prod_{i=1}^{k} p_i{:}\rho_1\bullet\cdots\bullet\rho_k)} \qquad \text{(sequence)}$$

P-BSM semantics (cont.)

P-BSM denotational semantics: τ as a function

$$\tau \rightsquigarrow \quad \mathfrak{fp}_\tau : \sigma \mapsto \{p : \rho | yields_p(\tau, \sigma, p : \rho)\}$$

$\mathfrak{fp}_\tau$ defines a probability distribution over the enabled updates (actions).

P-BSM operational semantics: set of all runs

computation run:

- $\omega = \sigma_1 \overset{p_1:\rho_1}{\longrightarrow} \sigma_2 \cdots \sigma_i \overset{p_i:\rho_i}{\longrightarrow} \sigma_{i+1} \cdots$
- probability of each finite prefix: $P(\omega') > 0$, s.t., $\omega' \in pref(\omega)$
- P-BSM weak fairness condition: $\liminf_{\substack{|\omega'| \to \infty \\ \omega' \in pref(\omega)}} \frac{freq_{p:\rho}(\omega')}{|\omega'|} \geq p$

P-BSM semantics (cont.)

P-BSM denotational semantics: τ as a function

$$\tau \rightsquigarrow \quad \mathfrak{fp}_\tau : \sigma \mapsto \{p : \rho | yields_p(\tau, \sigma, p : \rho)\}$$

$\mathfrak{fp}_\tau$ defines a probability distribution over the enabled updates (actions).

P-BSM operational semantics: set of all runs

computation run:

- $\omega = \sigma_1 \overset{p_1:\rho_1}{\longrightarrow} \sigma_2 \cdots \sigma_i \overset{p_i:\rho_i}{\longrightarrow} \sigma_{i+1} \cdots$
- probability of each finite prefix: $P(\omega') > 0$, s.t., $\omega' \in \mathit{pref}(\omega)$
- P-BSM weak fairness condition: $\liminf_{\substack{|\omega'| \to \infty \\ \omega' \in \mathit{pref}(\omega)}} \frac{\mathit{freq}_{p:\rho}(\omega')}{|\omega'|} \geq p$

Adjustable deliberation

probability specification for high level mst's

⇝ ***focus agent's deliberation to a specific task***

Example

e.g., focus agent's *attention* to a specific aspect of perception

```
when ⊨_bel [{ threatened }] then {
    /* ***Emergency modus operandi*** */
    0.7 : DETECT_ENEMY_POSITION ;
    0.2 : SENSE_CAMERA ;
          SENSE_HEALTH
} else {
    /* ***Normal mode of perception*** */
    SENSE_HEALTH ;
    SENSE_CAMERA
}
```

Summary

control over non-determinism in BDI style systems

Probabilistic Behavioural State Machines

- straightforward BSM extension
  - probabilities for choice mst's
- finer grained control behaviour selection
- semantics $\rightsquigarrow$ probability distribution over enabled actions
- adjustable deliberation

On-going & future work:

- preliminary tested in case studies
- extending the *Jazzyk* interpreter
- explore related work $\rightsquigarrow$ Markov chains and beyond(?)

Thank you for your attention...

http://jazzyk.sourceforge.net/