



Force Protection Call 4 A-0938-RT-GC

EUSAS

European Urban Simulation for Asymmetric Scenarios



Behaviour analysis and cloning – part1

Matjaz Gams, Ales Tavcar

Jozef Stefan Institute



Institut "Jožef Stefan"

© EADS 2010 – All rights reserved



INSTITUTE OF INFORMATICS
SLOVAK ACADEMY OF SCIENCES



Presentation outline



- **Cognitive Multi-Agent Strategy Discovering Algorithm (CMASDA)**
 - Features
 - Algorithm structure
 - Analysis results
- **Pattern Discovery Replay (PDR)**



Institut "Jožef Stefan"

© EADS 2010 – All rights reserved



INSTITUTE OF INFORMATICS
SLOVAK ACADEMY OF SCIENCES



FOI



CASSIDIAN
AN EADS COMPANY



Cognitive MASDA

Features

- Algorithm for advanced data analysis and **extraction of behaviour patterns** from low-level observations of two agent groups.
- Behaviour is analyzed:
 - a) To provide descriptions of relevant behaviour patterns of humans
 - b) To transfer behaviour patterns of observed live persons into simple behaviour models



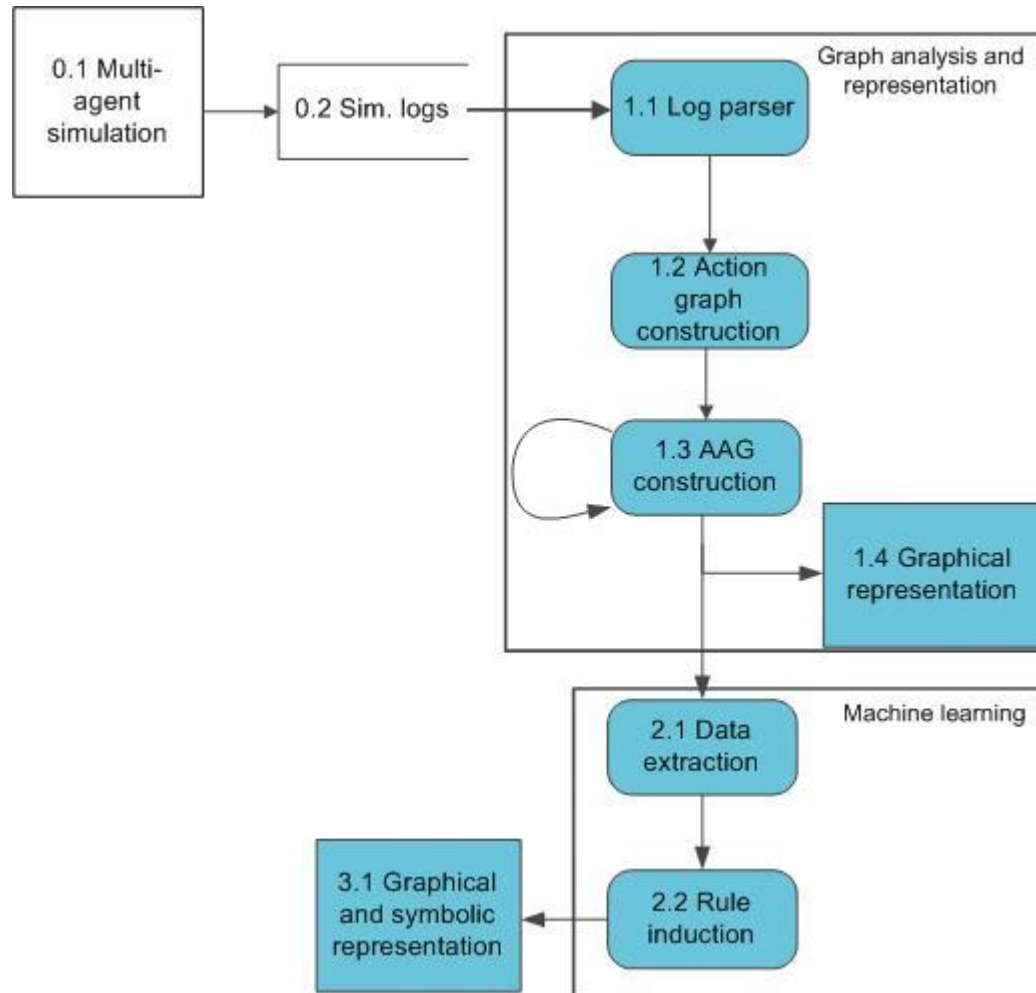
Institut "Jožef Stefan"





Cognitive MASDA

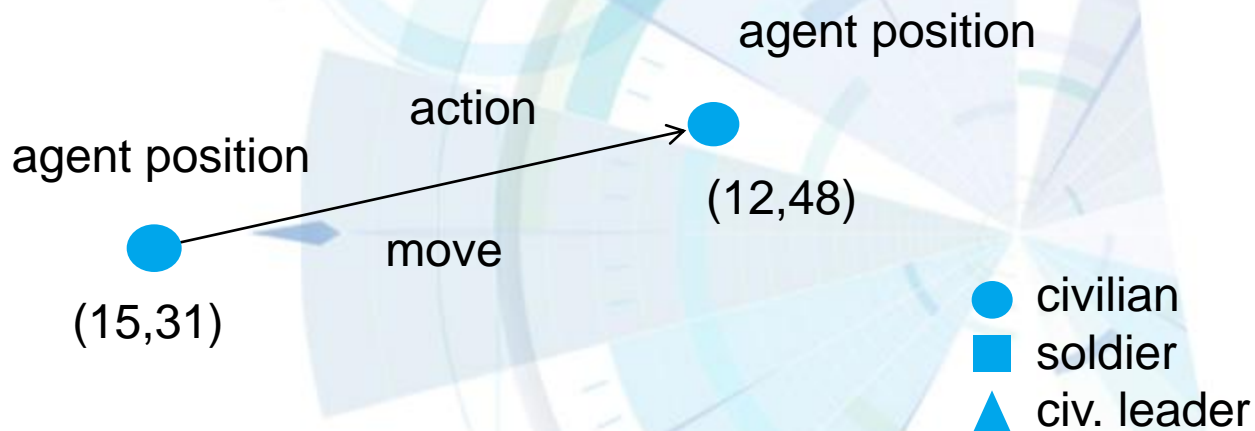
Algorithm structure



Cognitive MASDA

Action graph construction

- An example:



Cognitive MASDA / Vignette1.1

Action graph construction

cognitive features

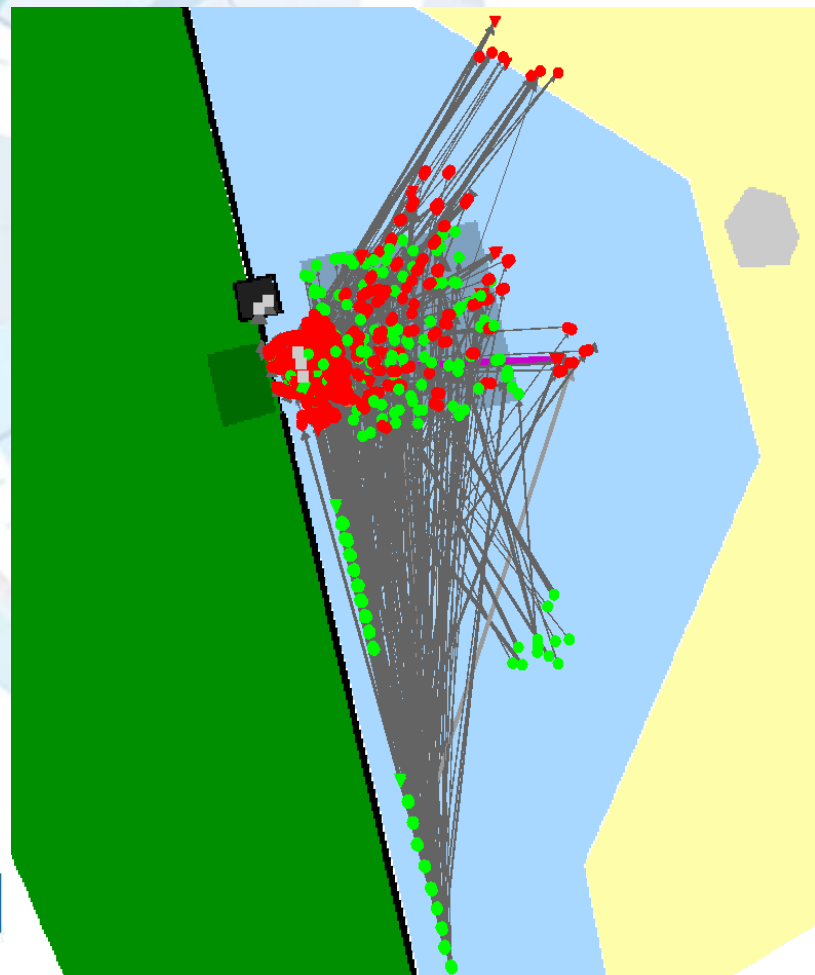
agents have cognitive states

cognitive taxonomy

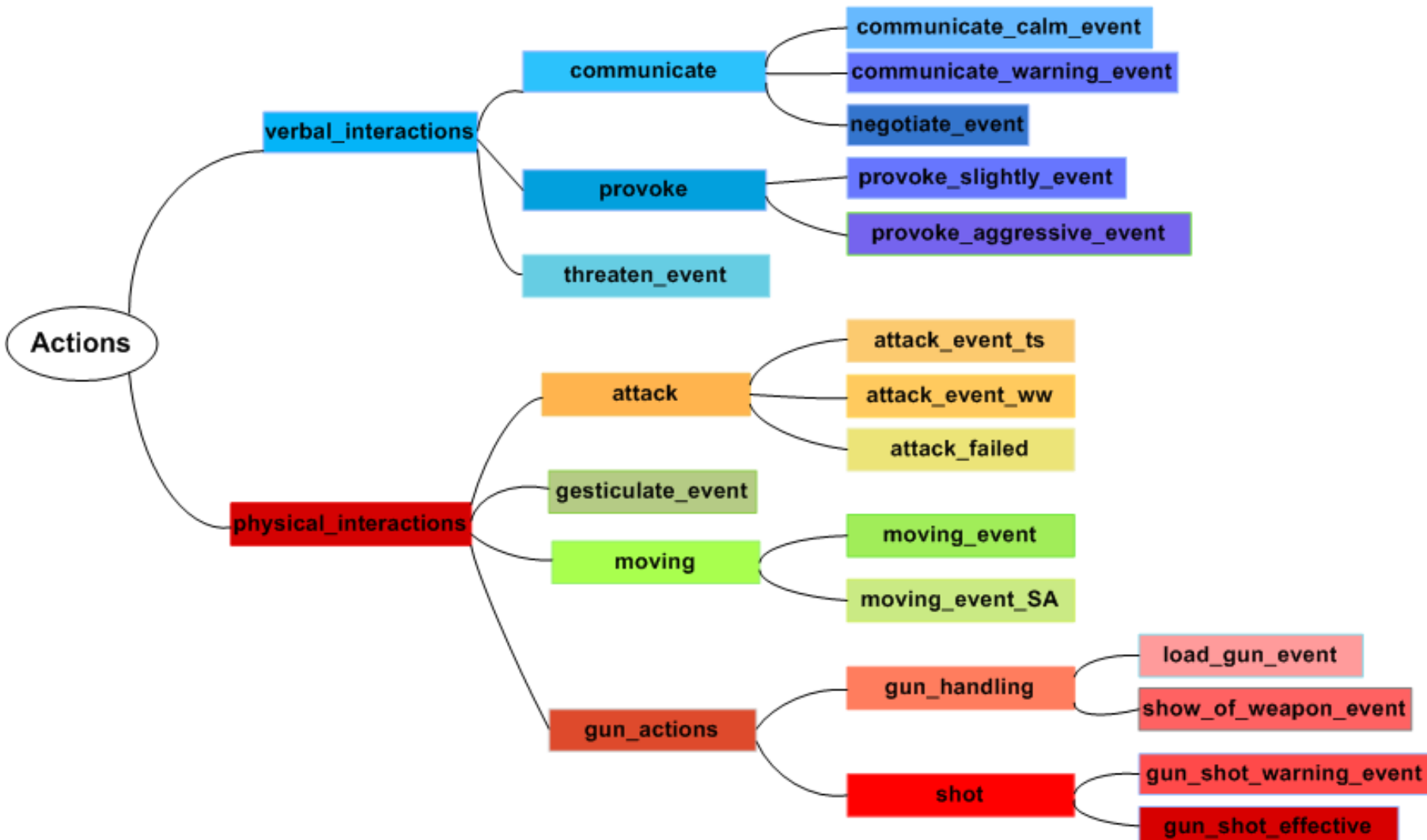
colour modifications:

nodes: cognitive states (red - anger,
green - need, blue - fear)

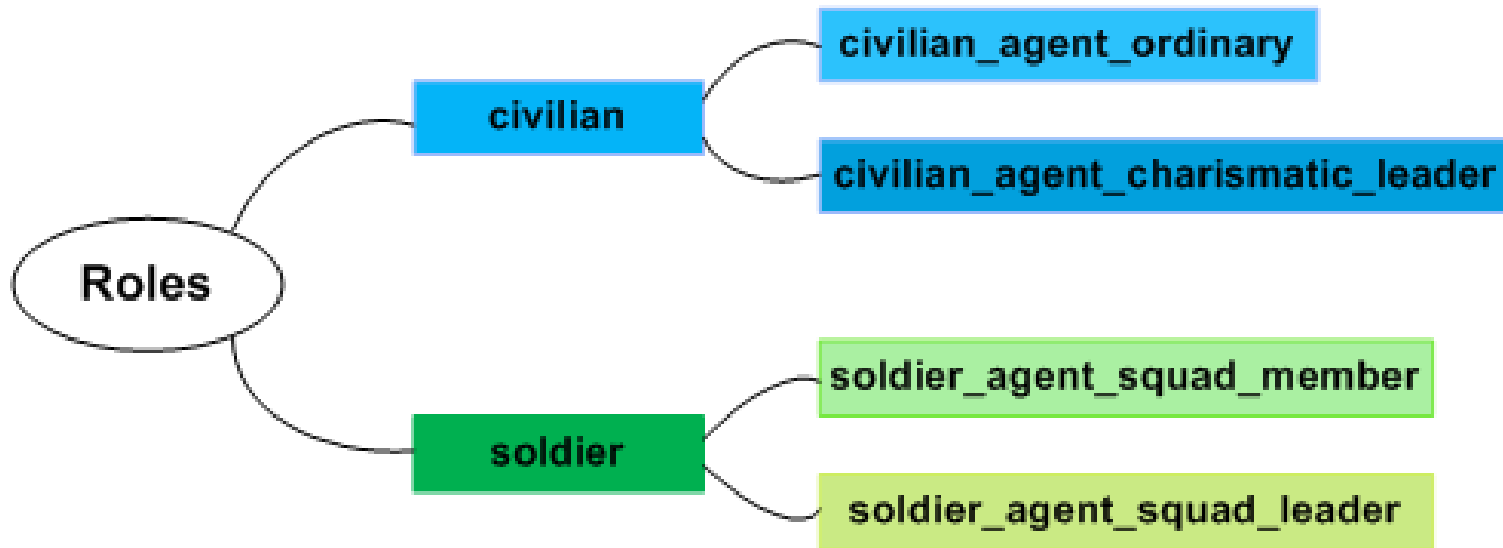
edges: actions (move_events – gray,..)



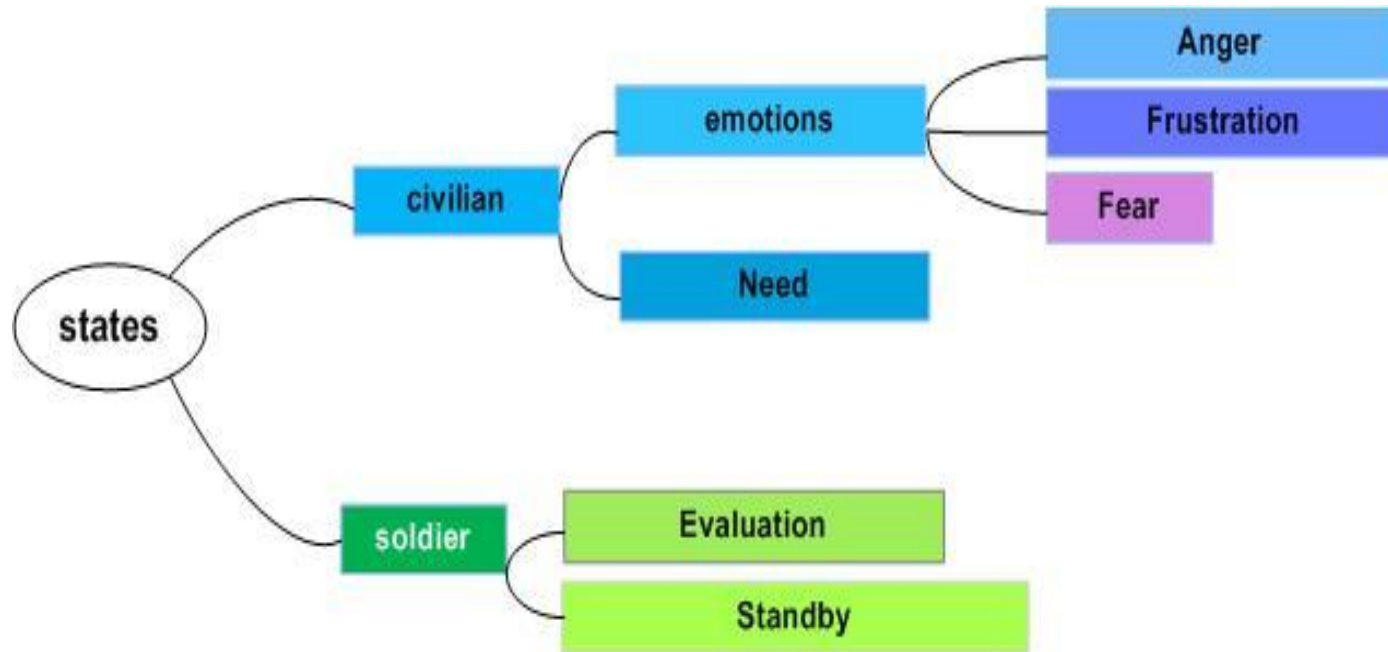
Taxonomies – action



Taxonomies – agent roles



Taxonomies – cognition





Multi-Agent Strategy Discovering Algorithm

Abstract action graph

- Abstraction process merges nodes together.
- Iteratively, the nearest two nodes are merged.
- Modified distance definition:
 - Weighted sum of distances between node positions and graph distances between role, action and cognitive concepts.

$$dist(a, b) = w_{pos} \cdot \overline{dist_{pos}(a, b)} + w_{role} \cdot \overline{dist_{Hrole}(a, b)} + w_{action} \cdot \overline{dist_{Haction}(a, b)} + w_{state} \cdot \overline{dist_{Hstate}(a, b)},$$



Institut "Jožef Stefan"



INSTITUTE OF INFORMATICS
SLOVAK ACADEMY OF SCIENCES

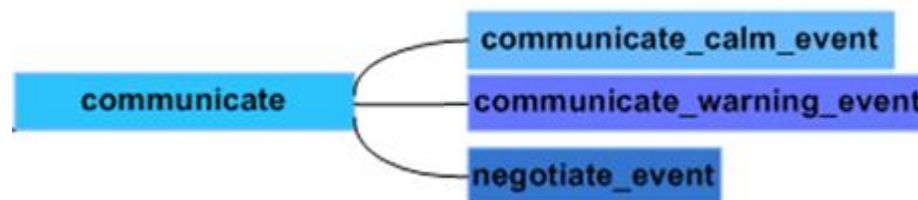
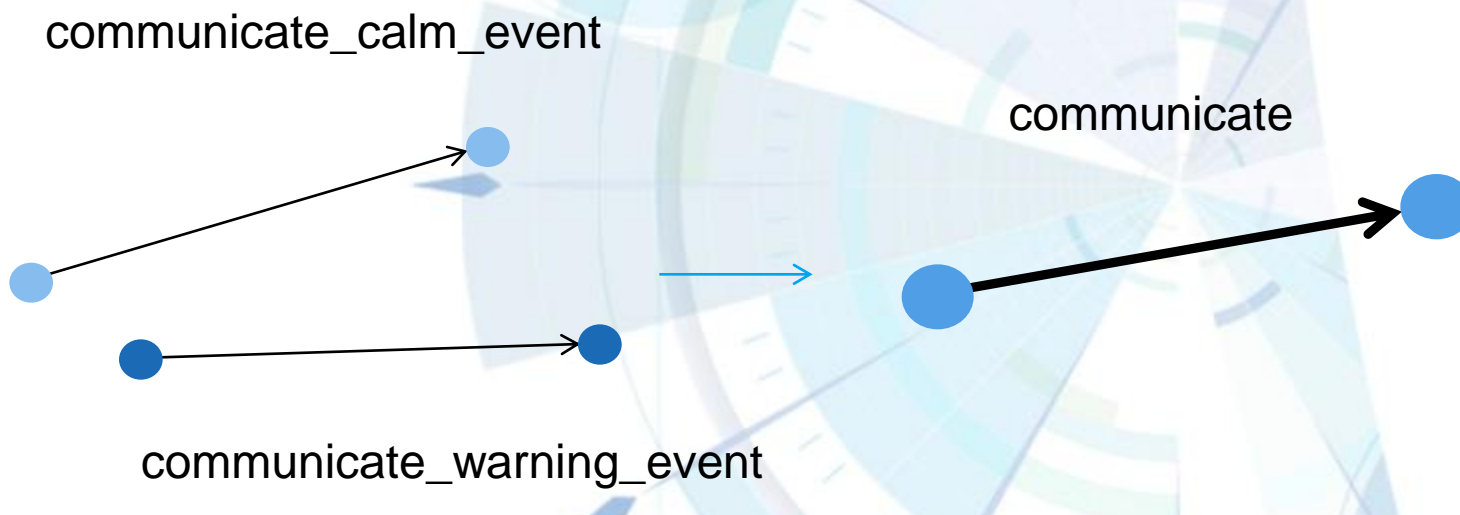


AN EADS COMPANY

Cognitive MASDA

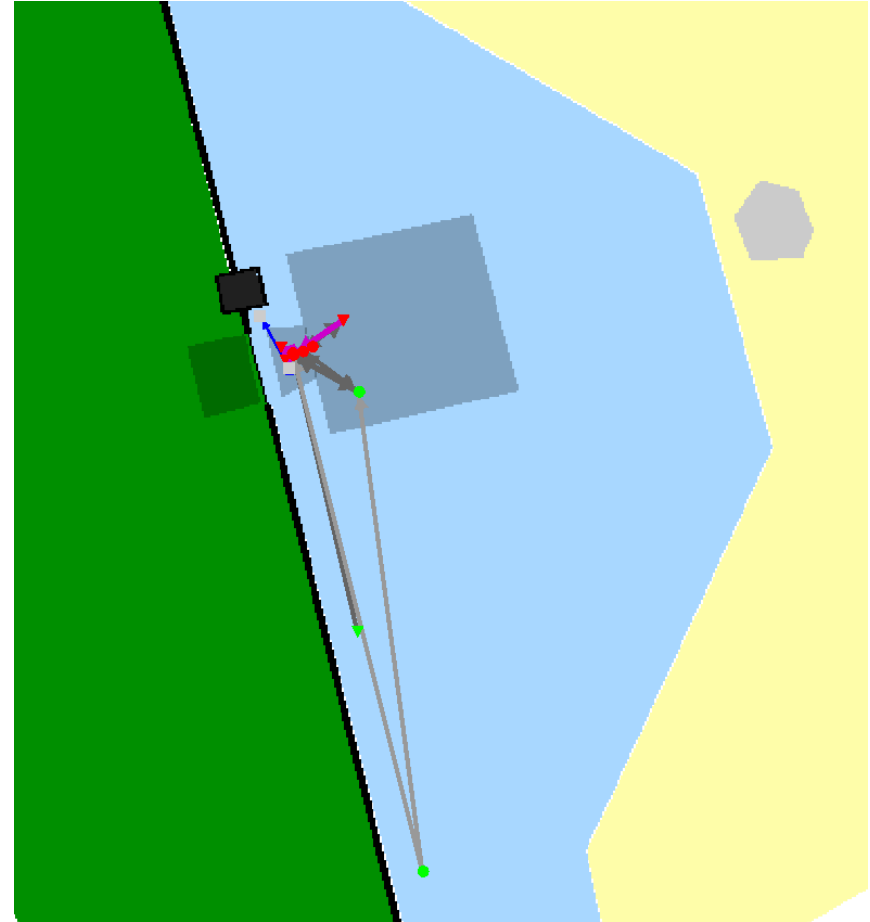
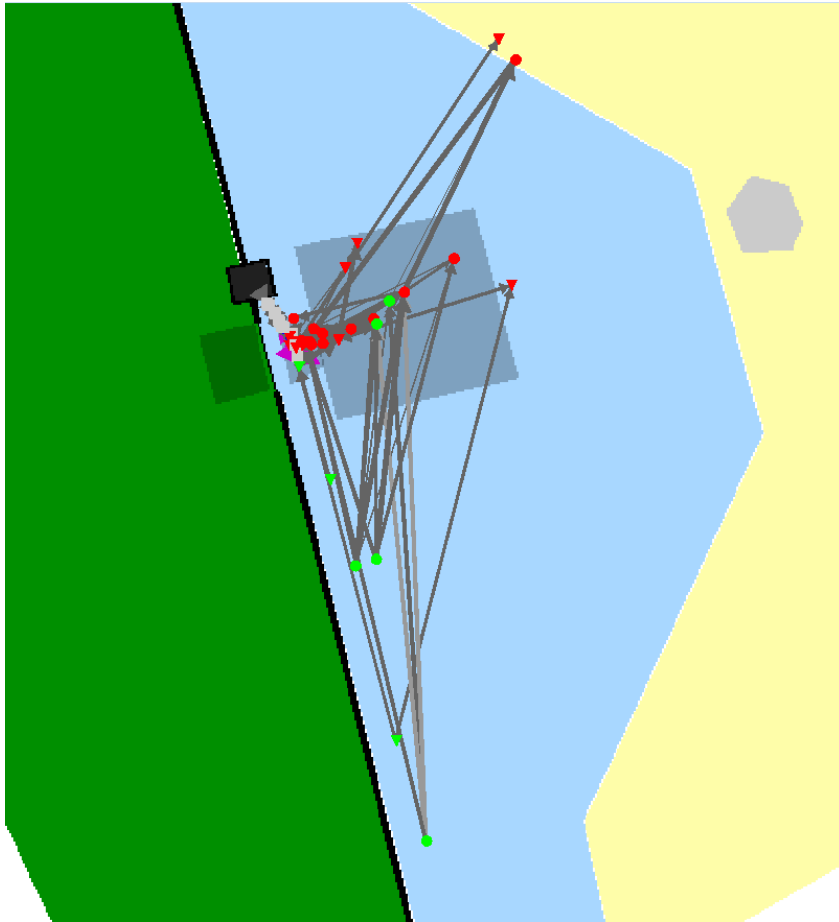
Abstract action graph

- An example:



Multi-Agent Strategy Discovering Algorithm

Abstract action graph



Institut "Jožef Stefan"





Language attributes

anger	inSoldierAreaSPA
anger_level	inFightAreaEntrance
fear	inFightArea
fear_level	inFightAreaInterest
need	inFightAreaInterestEntrance
need_level	inTowerArea
evaluation	inTowerFightArea
evaluation_level	Moving
standby	HasCACLNear
energy_level	HasCommunicatedCalmEvent
average_civilian_anger	HasGesticulateEvent
CALC_anger_value	HasShowedWeapon
CALC_motive_value	HasLoadedGun
NearProvokeSlightlyEvent	HasPerformedWarningShot
inSoldierArea	HasPerformedEffectiveShot





Multi-Agent Strategy Discovering Algorithm

Machine learning

- Machine learning: providing symbolic explanation of a graphical pattern
- Machine learning from abstract action graph + taxonomies + description of attributes
- Area of interest around graphical patterns
- ML learns all interesting patterns inside area of interest



Multi-Agent Strategy Discovering Algorithm

Civilian pattern



The screenshot shows a software interface with a map on the left and several panels on the right. The map displays a green area on the left and a blue area on the right, with a path of grey squares connecting a red triangle to a green triangle. The interface includes tabs for 'Abstraction', 'Strategy viewer', and 'Soldier panel'. The 'Settings' panel on the right has 'Path length' set to 3 and 'Minimum path occurrence' set to 1, with a 'Process' button. The 'Filter' panel shows a tree structure of actions, with 'negotiate_event' selected. The 'Strategic paths' panel lists a sequence of 'negotiate_event' actions with increasing indices, with the 24th action highlighted. A central text box contains the following rules:

IF CACL Needy THEN moving_event
IF energy_level >= 100 THEN moving_event

An 'Info' box below the rules provides details:

- Number of actions: 3
- Abstracted action: moving_event
- Rule:
 - IF energy_level >= 100 THEN moving_event
 - IF CACL Needy THEN moving_event
- Actions:
 - 3 x moving_event



Multi-Agent Strategy Discovering Algorithm

Civilian pattern



The screenshot displays the Multi-Agent Strategy Discovering Algorithm (EDA) interface. The main window is divided into several panels:

- Abstraction:** Shows a map with a green area (friendly) and a black area (enemy). A path is highlighted with a green dot and arrows.
- Strategy viewer:** Displays the discovered strategy rules:

```
IF energy_level = 92 AND CAO Angry THEN attack_event_ww
IF energy_level = 97 AND anger_level >= 100
AND need_level >=68 THEN attack_event_ww
```
- Soldier panel:** Shows the current state of the soldier, including energy and anger levels.
- Settings:** Includes fields for Path length (2) and Minimum path occurrence (1), with a Process button.
- Filter:** Shows a tree view of actions, including verbal_interactions and communicate.
- physical_interactions:** Shows a tree view of physical interactions, including attack_event_ts, attack_event_ww, attack_failed, and gesticulate_event.
- Strategic paths:** Lists the discovered strategic paths, such as:

```
7: attack_event_ww(20)
7: attack_event_ww(21)
6: attack_event_ww(25)
5: attack_event_ww(38)
5: attack_event_ww(39)
4: attack_event_ww(56)
4: attack_event_ww(57)
2: attack_event_ww(154)
2: attack_event_ww(184)
2: attack_event_ww(187)
2: attack_event_ww(190)
2: attack_event_ww(194)
2: attack_event_ww(195)
2: attack_event_ww(197)
2: attack_event_ww(198)
2: attack_event_ww(200)
```


An info panel provides details about the current path, including the number of actions (45), the abstracted action (attack), the rule used, and the actions performed (40 x attack_event_ww, 2 x attack_failed, 3 x attack_event_ts).



Multi-Agent Strategy Discovering Algorithm

Soldier pattern

Abstraction Strategy viewer Soldier panel



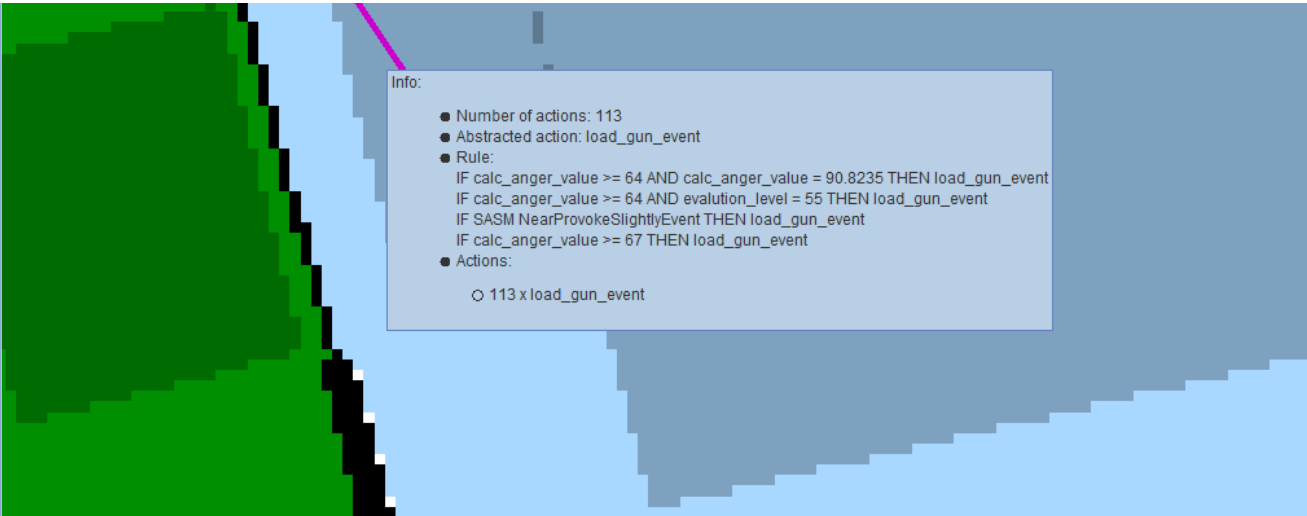
Settings

Path length:

Minimum path occurrence:

Process

IF calc_anger_value >= 64 AND evaluation_level = 55 THEN load_gun_event
 IF SASM NearProvokeSlightlyEvent THEN load_gun_event
 IF calc_anger_value >= 67 THEN load_gun_event



Info:

- Number of actions: 113
- Abstracted action: load_gun_event
- Rule:
 - IF calc_anger_value >= 64 AND calc_anger_value = 90.8235 THEN load_gun_event
 - IF calc_anger_value >= 64 AND evaluation_level = 55 THEN load_gun_event
 - IF SASM NearProvokeSlightlyEvent THEN load_gun_event
 - IF calc_anger_value >= 67 THEN load_gun_event
- Actions:
 - 113 x load_gun_event

Filter

- physical_interactions
 - attack
 - gesticulate_event
 - moving
 - gun_actions

Strategic paths

- 7: communicate_calm_event(34)
- 2: communicate_calm_event(127)
- 2: communicate_calm_event(148)
- 2: communicate_calm_event(149)
- 2: communicate_calm_event(150)
- 1: communicate_calm_event(343)
- 1: communicate_calm_event(346)
- 1: communicate_calm_event(349)

Visualize



Multi-Agent Strategy Discovering Algorithm

Soldier pattern



Abstraction Strategy viewer Soldier panel

Settings

Path length: 2

Minimum path occurrence: 1

Process

Filter

- communicate_warning_event
- communicate_warning_event
- negotiate_event
- provoke
- threaten_event
- physical_interactions
- attack
- gesticulate_event
- moving
- gun_actions
 - gun_handling
 - shot
 - gun_shot_effective
 - gun_shot_warning_event

Info:

- Number of actions: 31
- Abstracted action: shot
- Rule: IF calc_anger_value >= 100 AND SASM HasLoadedGun THEN gun_shot_warning_event
- Actions:
 - 29 x gun_shot_warning_event
 - 2 x gun_shot_effective

Strategic paths

- 3: gun_shot_warning_event(86)
- 3: gun_shot_warning_event(87)
- 2: gun_shot_warning_event(134)
- 2: gun_shot_warning_event(135)
- 1: gun_shot_warning_event(500)
- 1: gun_shot_warning_event(501)

Visualize

IF calc_anger_value >= 100 AND SASM HasLoadedGun THEN gun_shot_warning_event

Multi-Agent Strategy Discovering Algorithm

Soldier pattern



The screenshot shows the 'Settings' panel with 'Path length' set to 2 and 'Minimum path occurrence' set to 1. The 'Filter' panel shows a tree view of event categories: negotiate_event, provoke, provoke_slightly_event, provoke_aggressive_event, threaten_event, physical_interactions, attack, attack_event_ts, attack_event_ww, and attack_failed.

IF evaluation_level \geq 62.5 THEN gun_shot_warning_event

IF evaluation_level \geq 65 AND SASM HasLoadedGun THEN gun_shot_warning

The 'Strategic paths' panel lists various events with their occurrence counts, such as '2: gun_shot_effective(139)' and '1: gun_shot_warning_event(290)'. An 'Info' popup is visible, providing details about the number of actions (17), the abstracted action (gun_shot_warning_event), and the rules that trigger it.

Info:

- Number of actions: 17
- Abstracted action: gun_shot_warning_event
- Rule:
 - IF evaluation_level \geq 62.5 THEN gun_shot_warning_event
 - IF evaluation_level \geq 55 AND SASM HasLoadedGun THEN gun_shot_warning_event
- Actions:
 - 17 x gun_shot_warning_event

Pattern editor



Abstraction Strategy viewer Soldier panel

Run

load_gun_event:IF evaluation_level >= 55 THEN load_gun_event;	<input checked="" type="checkbox"/>
load_gun_event:IF evaluation_level >= 70 THEN load_gun_event;	<input checked="" type="checkbox"/>
load_gun_event:IF evaluation_level >= 55 AND calc_anger_value <= 63 THEN load_gun_event;	<input checked="" type="checkbox"/>
load_gun_event:IF evaluation_level >= 35 AND evaluation_level >= 64.2857 THEN load_gun_event;IF evaluation_level >= 35 THEN load_gun_event;	<input checked="" type="checkbox"/>
communicate_calm_event:IF evaluation_level >= 35 AND SASM HasPerformedWarningShot THEN communicate_calm_event;	<input checked="" type="checkbox"/>
communicate_calm_event:IF evaluation_level >= 39 AND SASM Standby THEN communicate_calm_event;IF evaluation_level >= 39 THEN communicate_calm_event;	<input checked="" type="checkbox"/>



Institut "Jožef Stefan"



XML output of patterns

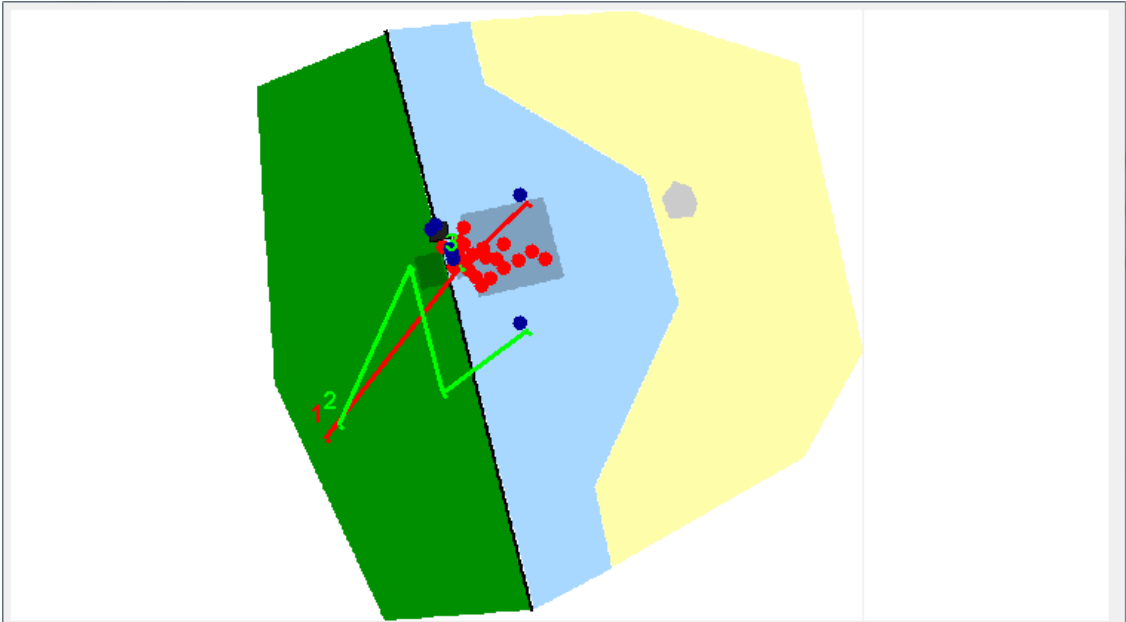
Contains graphical and symbolic description

ⓐ	xmlns:library	http://www.eusas.org/masda/library
ⓐ	xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
ⓐ	xsi:schemaLocation	http://www.eusas.org/masda/library library.xsd
▲	library:pattern	
ⓐ	id	1
e	library:description	Test
e	library:result	ALARM
▲	library:graph	
▲	library:edges	
ⓐ	destination	1
ⓐ	rule	0
ⓐ	source	0
▷	library:edges	
▷	library:edges	
▲	library:vertices	
ⓐ	id	0
e	library:action	communicate_calm_event
e	library:role	SASM
▷	library:vertices	
▷	library:vertices	
▷	library:vertices	
▲	library:rules	
▲	library:ruleSet	
ⓐ	id	0
▷	library:rule	
▷	library:rule	
▷	library:ruleSet	
▲	library:ruleSet	





Pattern discovery replay / vignette1



- 3. OK
00:01:07 - 00:02:08
Soldier SASM@data\vignette1\test_ok.logSASM#3 Communicate, load gun, warning shot, shot effective.
- 1. ALARM!
00:01:33 - 00:01:33
Soldier SASM@data\vignette1\test_ok.logSASM#4 Communicate, load gun, WITHOUT warning shot, shot effective.
- 2. OK
00:01:33 - 00:01:33
Soldier SASM@data\vignette1\test_ok.logSASM#5 Communicate, load gun, warning shot, shot effective.

Time: -



Conclusion and discussion



- Explanation how CMASDA works
- The CMASDA algorithm discovers relevant patterns in the Vignette1.1 scenario as planned (civilian and soldier)
- MASDA discovered patterns on its own from raw data and taxonomies

Thank you



Institut "Jožef Stefan"

© EADS 2010 – All rights reserved



INSTITUTE OF INFORMATICS
SLOVAK ACADEMY OF SCIENCES



FOI



CASSIDIAN
AN EADS COMPANY