

# Extensions and applications of Pogamut 3 platform

Rudolf Kadlec, Jakub Gemrot, Michal Bída, Ondřej Burkert, Jan Havlíček, Lukáš Zemčák, Radek Pibil, Radim Vansa, Cyril Brom.

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic  
 {rudolf.kadlec, jakub.gemrot, michal.bida}@gmail.com, brom@ksvi.mff.cuni.cz

<http://artemis.ms.mff.cuni.cz/pogamut>

## Overview

Pogamut 3 is a platform for easy creation of IVA's behavior. This poster presents extensions of the Pogamut platform that are making it more suitable for storytelling, animations, cognitive modeling, emotional modeling and education. All these works are downloadable from:

<http://artemis.ms.mff.cuni.cz/pogamut>

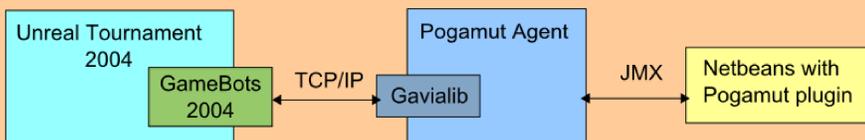
## Pogamut 3 platform

### Goal

Create platform for creation of IVA's behavior suitable for both research and education.

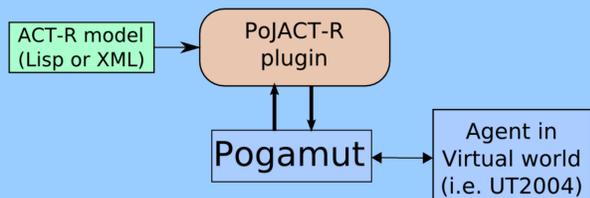
### Features

- Java API for coding IVA's behavior
- Netbeans™ IDE Pogamut plugin with support for tuning and debugging of IVAs
- Set of example IVAs, tutorials
- Opensource, online support on forums



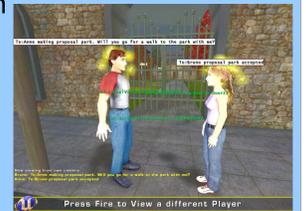
## Cognitive modeling with ACT-R

ACT-R [1] is a cognitive architecture of human cognition. PoJACT-R is an extension of Pogamut that can be used by cognitive researchers to test their models in virtual worlds. It uses jACT-R implementation of ACT-R. PoJACT-R currently consists of basic perceptual and motor modules and runtime that maintains bootstrap, connection and communication with virtual worlds of Pogamut. PoJACT-R supports both Lisp and XML syntax for ACT-R models.



## Emotions

Emotional AI in UT is a prototype implementation of Emohawk educational toolkit [2] based on Pogamut. This toolkit should facilitate education in basic and advanced IVA development (locomotion, coordination, storytelling, emotion modeling, etc.)



In the prototype we have:

- created a binding of ALMA [3] emotional model to Pogamut
- implemented affect-driven architecture based on ALMA affects
- created a working example scenario featuring three IVAs controlled by affect-driven architecture
- proposed and used evaluation methodology of this example scenario

## Gestures and facial expressions

We have implemented a base for a simple way of defining gestures for virtual agents to express themselves nonverbally in a believable manner.

For that we have provided an interpreter of (a subset of) BML [4] language for automatic construction of animation frames of modified MPEG-4 standard, as well as for their manual construction.

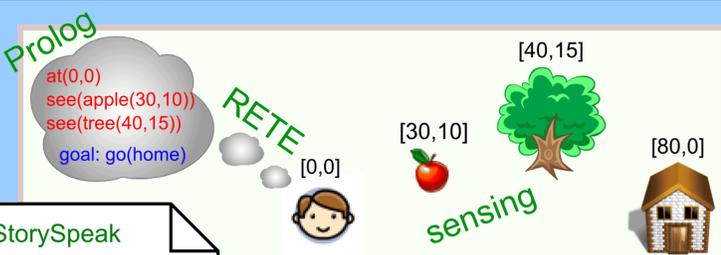
```

<gaze id="gaze1" start="0" end="10" target="DM-Flux2.InventorySpot49" type="AT"/>
<gaze id="gaze2" start="gaze1:end + 1" end="gaze2:start + 10" target="DM-Flux2.PathNode52" type="AVERT"/>
<gesture id="gesture1" start="gaze2:end + 2" end="gesture1:start + 12" type="LEXICALIZED" lexeme="honest_right.xml"/>
<gesture id="gesture2" start="gesture1:end + 2" end="gesture2:start + 2" type="LEXICALIZED" lexeme="kick_dance.xml"/>
  
```

```

<start_to_ready>
<animation_block length="2000">
<parameter name="LeftArm1/l_shoulder_flexion" intensity="16383" type="BAP"/>
<parameter name="LeftArm1/l_elbow_flexion" intensity="16383" type="BAP" />
<parameter name="LeftHand1/l_thumb1_flexion" intensity="8000" type="BAP" />
  
```

## StorySpeak



**StorySpeak**  
 +!near(me, Object)  
 : ?wants(Object)  
 <- grab(Object)

StorySpeak is a language for the specification of agents' programs. It is yet-another formalization of the BDI+P paradigm. This time, the language is especially designed for the authoring of short virtual stories.

The aim of the language is to provide means for both character-centric and author-centric approach. The StorySpeak interpreter distinguishes between virtually embodied agents (*actors*) and bodiless omniscient agent (*story director*).

Firstly, the author has means to create roles (the set of believes, desires and plans) for every actor that should participate in the story. Secondly, the author may specify plans for the story director, that can alter the BDI intention stack of any actor at any given time allowing the author to switch between two mentioned approaches.

## Educational scenarios

Educational scenarios is Pogamut based framework for development of educational applications by non-programmers (teachers, sociologists etc.). Scenarios are scripted in restricted natural language. Drools Business Rules was used as an inference engine.



```

rule "Returned"
when
  performing act "wrong direction"
  player is in area "allowed area"
then
  stop timer "wrong direction"
  show message "Great, you have returned back." for 8 seconds
  play "EducationalScenarios.ExampleScenario.Returned"
  add 1 penalty point to player
  add text "returned by himself" to report "report" as "Correction"
end
  
```

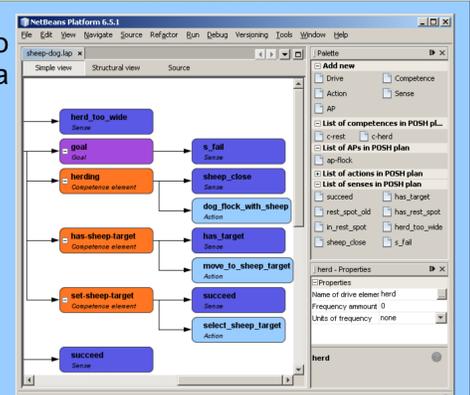
Example scenario aimed at testing of orientation in urban areas.

## POSH visual editor

POSH visual editor enables you to create a new POSH plan [5] using a simple point-and-click GUI.

### Features:

- Multiple editing modes (1 text and 2 graph modes)
- Generated palette with acts and senses
- Integrated Drag&Drop of elements
- Token highlighting in source code



### References:

- [1] ACT-R. <http://act-r.psy.cmu.edu/>
- [2] Brom, C., Bída, M., Gemrot, J., Kadlec, R., Plch., T.: Emohawk: Learning Virtual Characters by Doing. In: Proceeding of TIDSE 2009. (2009) (accepted, to appear)
- [3] Gebhard, P.: ALMA – A Layered Model of Affect. <http://www.dfki.de/~gebhard/alma/index.html>
- [4] BML – Behaviour Markup Language. <http://wiki.mindmakers.org/projects:bml:main>
- [5] Bryson, J.: POSH Action Selection. <http://www.cs.bath.ac.uk/~jjb/web/posh.html>

### Acknowledgment:

This work was supported by the GACR 201/09/H057 grant, by the Program "Information Society" under project 1ET100300517 and by GA UK no. 21809.