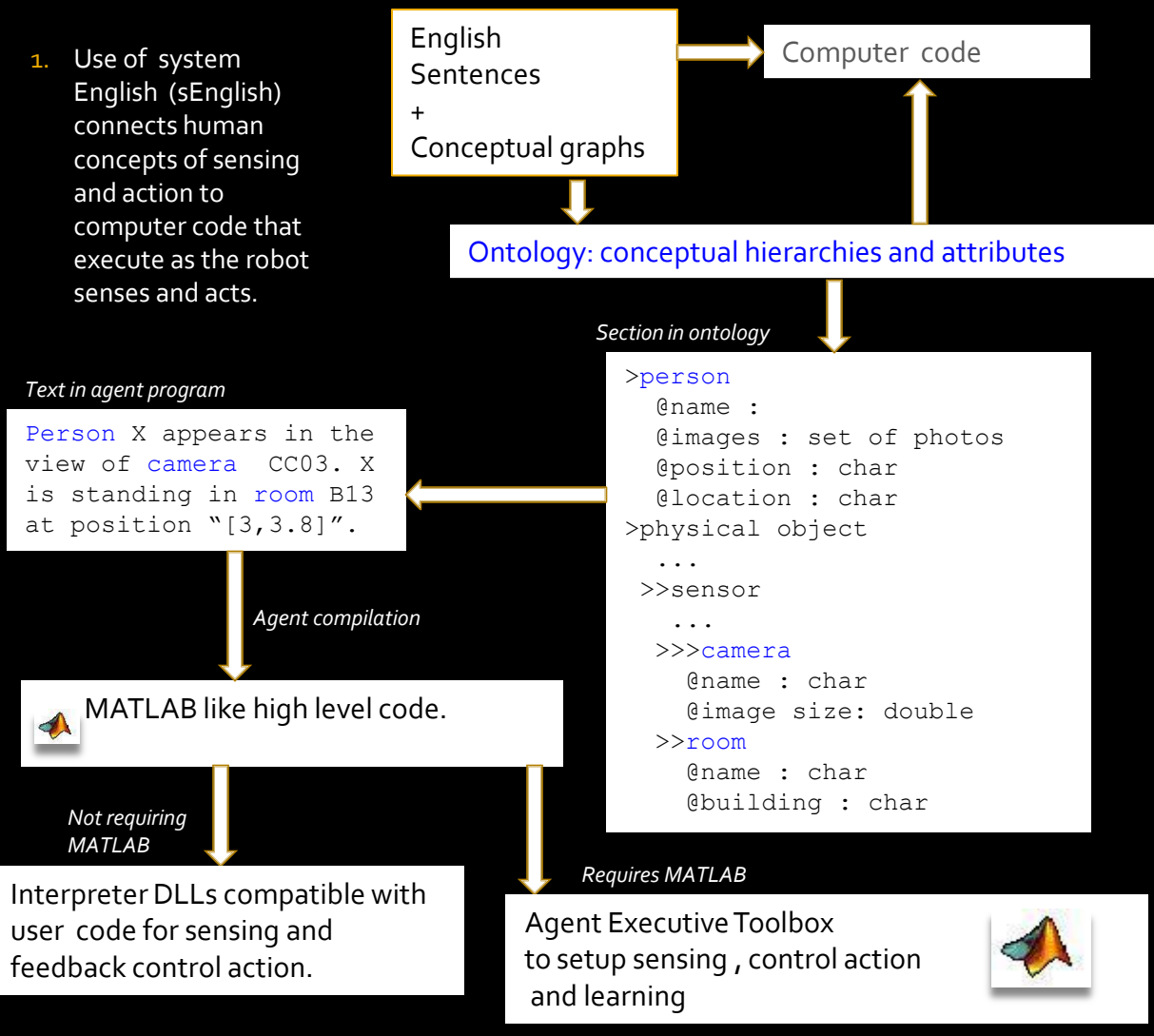
Why does the use of Cognitive Agent Toolbox results in significant time savings and in more efficient agent programming?

1. Use of system English (sEnglish) connects human concepts of sensing and action to computer code that execute as the robot senses and acts.



This leads to shared knowledge of the world between humans and the machine interpreting sEnglish.

Why does the use of the Cognitive Agent Toolbox results in significant time savings and in more efficient agent programming and agent operations?

2. Belief-desire-intention agents programming in terms of sEnglish sentences

```
//c basic rules
^[Analyse message Ms.] :-[ Ms is a new message.] & [The sender  of Ms is trustworthy.].
^[Update world model W by Ms.] :- [Message Ms informs.] .
^[Analyse message Ms.]  :- [Message Ms requests.].

//c basic plans
+^[Analyse message Ms.]<- [Get the type of message Ms.] ; [Interpret message Ms in view of world model W.];
                [Get world change Dws associated with message Ms .];[Update world model W using Ms.];
                [Generate answer Ans to Ms.];
                [Establish whether feasible plan Pla exists to achieve world change Dws using capabilities Cap. ].
//r +^[No plan can achieve Dws.]
//r +^[There is plan Pla to achieve Dws.]
//r +^[World change Dws is desirable.]
. . .
```

Belief-desire-intention agents programming in terms of sEnglish/Jason sentences permits definition of behaviour rules in terms of Prolog like rules in sentences. Plans for goal achievement are stated in sentences with Jason syntax in between sentences.
Notations:
BLUE = sEnglish sentences defined in associated sEnglish document
GREEN = comments
RED = annotations for model checking of the agent/environment hybrid system by MCMAS

3. The ontology/sentences based programming permits the programming of agent operations in high complexity environments.
4. Shared understanding of operations is established between human operators of the system and the agent system itself. Apart from ease of operations, the advantage of shared understanding is crucial for fast efficient joint solutions to be found by the agent(s) and human(s).
5. The agents are able to solve highly complex problems that otherwise are only accessible to humans today.

This leads to an easier to setup and to maintain agent system. In addition, the operators of the agents can get day to day insight why the agents made decisions and actions the way they did. Within bounds operators can also redefine the logic of agents.

How is an agent class defined in the Cognitive Agent Toolbox ?

Phase 1. Use the sEnglish Eclipse plugi-n to create the core concepts of your agent class for world modelling . An agent class is all the high level code not implemented yet into an agent individual.

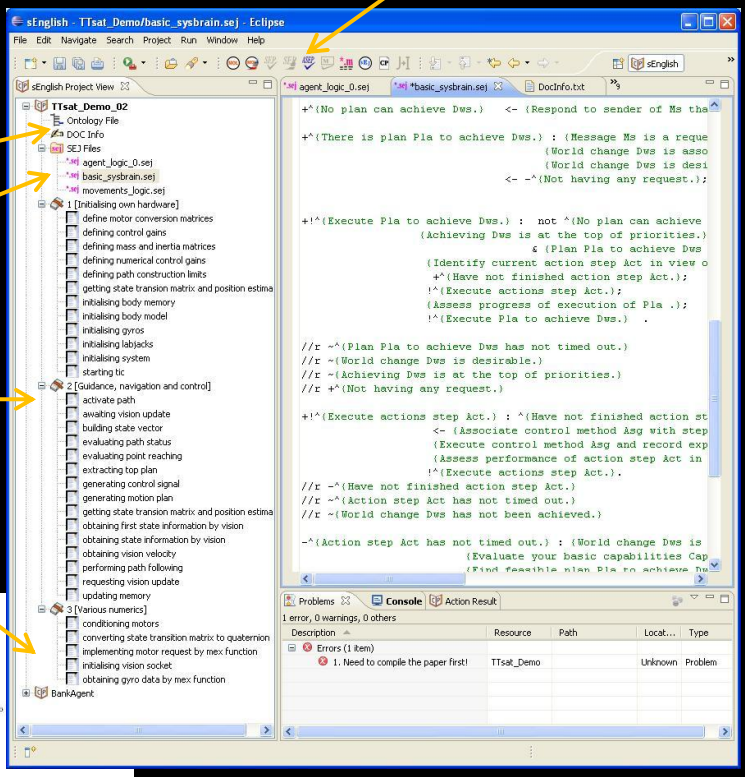
sEnglish specific editing , debugging and compilation buttons

A single sEnglish document contains all the concepts the agent knows and its logic and physical abilities

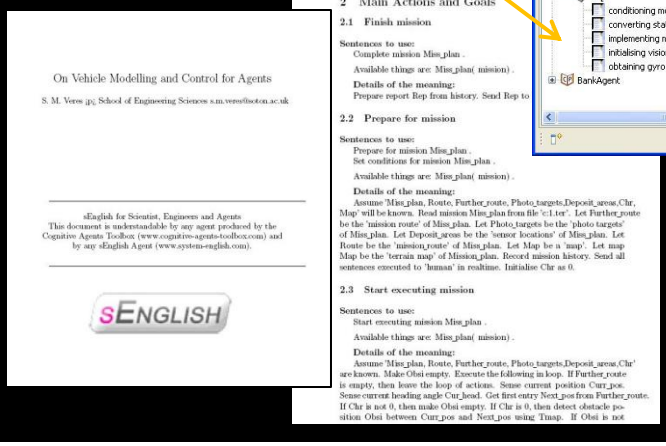
Ontology (.ont file),

sEnglish/Jason logic (.sej files)

and sections/subsections of **sEnglish Document** to store procedural details on sensing and action taking in the physical world in terms of sentence definitions



Several agents can be edited at the same time


On Vehicle Modelling and Control for Agents

S. M. Veres, School of Engineering Sciences s.m.veres@soton.ac.uk

sEnglish for Scientist, Engineers and Agents
This document is understandable by any agent produced by the Cognitive Agents Toolbox (www.cognitive-agents-toolbox.com) and by any sEnglish Agent (www.system-english.com).

sENGLISH

2 Main Actions and Goals

2.1 Finish mission

Sentences to use:
Complete mission Miss_plan .
Available things are: Miss_plan(mission) .
Details of the meaning:
Prepare report Rep from history. Send Rep to

2.2 Prepare for mission

Sentences to use:
Prepare for mission Miss_plan .
Set conditions for mission Miss_plan .
Available things are: Miss_plan(mission) .
Details of the meaning:
Assume 'Miss_plan, Route, Further_route, Photo_targets,Deposit_areas,Chr, Map' will be known. Read mission Miss_plan from file 'c:1.ter'. Let Further_route be the 'mission route' of Miss_plan. Let Photo_targets be the 'photo targets' of Miss_plan. Let Deposit_areas be the 'sensor locations' of Miss_plan. Let Route be the 'mission_route' of Miss_plan. Let Map be a 'map'. Let map Map be the 'terrain map' of Mission_plan. Record mission history. Send all sentences executed to 'human' in realtime. Initialise Chr as 0.

2.3 Start executing mission

Sentences to use:
Start executing mission Miss_plan .
Available things are: Miss_plan(mission) .
Details of the meaning:
Assume 'Miss_plan, Route, Further_route, Photo_targets,Deposit_areas,Chr' are known. Make Obsi empty. Execute the following in loop. If Further_route is empty, then leave the loop of actions. Sense current position Curr_pos. Sense current heading angle Cur_head. Get first entry Next_pos from Further_route. If Chr is not 0, then make Obsi empty. If Chr is 0, then detect obstacle position Obsi between Curr_pos and Next_pos using Tmap. If Obsi is not

The edited files compile into .html or .tex LaTex documents at the click of the CP button in Eclipse.

A first phase of agent-group programming consists of defining the agents in an efficient way in a single document where agent share concepts and some sensing and actions skills executed by sentences (.sep files). This phase also defines the reasoning for goal achievement, planning and execution capabilities of an agent (.sej files).