A Practical Guide to Studying Emergent Communication through Grounded Language Games

Jens Nevens and **Paul Van Eecke** and **Katrien Beuls** [1]

Abstract. The question of how an effective and efficient communication system can emerge in a population of agents that need to solve a particular task attracts more and more attention from researchers in many fields, including artificial intelligence, linguistics and statistical physics. A common methodology for studying this question consists of carrying out multi-agent experiments in which a population of agents takes part in a series of scripted and task-oriented communicative interactions, called 'language games'. While each individual language game is typically played by two agents in the population, a large series of games allows the population to converge on a shared communication system. Setting up an experiment in which a rich system for communicating about the real world emerges is a major enterprise, as it requires a variety of software components for running multi-agent experiments, for interacting with sensors and actuators, for conceptualising and interpreting semantic structures, and for mapping between these semantic structures and linguistic utterances. The aim of this paper is twofold. On the one hand, it introduces a high-level robot interface that extends the Babel software system, presenting for the first time a toolkit that provides flexible modules for dealing with each subtask involved in running advanced grounded language game experiments. On the other hand, it provides a practical guide to using the toolkit for implementing such experiments, taking a grounded colour naming game experiment as a didactic example.

1 INTRODUCTION

How can a population of agents self-organise an effective and efficient communication system that allows them to communicate about their native environment? This fundamental research question concerning the mechanisms underlying human-like communication systems has for a long time sparked the interest of researchers from many fields, including artificial intelligence (e.g. [22, 14]), linguistics (e.g. [36, 9]), and statistical physics (e.g. [1, 17]). Well-attended workshops at important conferences, such as the NeurIPS emergent communication workshop, indicate that the community interested in models of emergent communication is growing ever more rapidly.

A common methodology for studying emergent communication consists of carrying out multi-agent experiments in which a population of agents takes part in a series of scripted and task-oriented communicative interactions, called 'language games' [22]. Each game is typically played locally by two agents in the population without any form of central control and without the agents having any mind-reading capacities. Through self-organisation, the population converges on a shared communication system after playing a large number of games [7]. The most widely studied language game is the Naming Game [22, 33, 1], in which the task involves referring to individual objects and thereby establishing a shared lexicon of proper nouns. More advanced scenarios include games in which the agents refer to the properties of objects or events [35], use multi-word expressions [34], or develop grammatical structures [2].

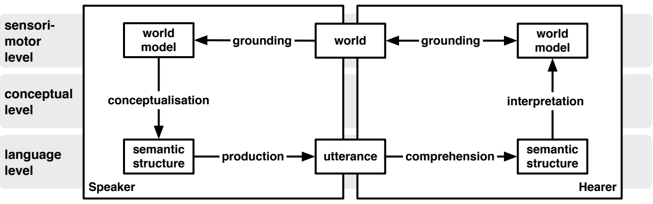


Figure 1: Grounded robot interactions follow a semiotic cycle that involves three main levels: the sensori-motor level, the conceptual level and the language level. The speaker agent (on the left hand-side) and the hearer agent (on the right hand-side) only share the world in which they are situated and the utterance that the speaker produces.

Mathematical investigations and computer simulations help making the assumptions of a specific theory explicit, allowing researchers to study the emergence of a particular communication system in a simulated world, comparing different scenarios and parameter settings. Yet, the step from such a simulated world to the real world with noisy sensori-motor values is crucial to make and requires the use of physical robots, as has been advocated in the work of many researchers studying the emergence and evolution of speech and language [11, 28, 15]. The increased realism leads to the need for more robust and fine-grained models, as has for instance been shown when moving from the Naming Game in a simulated world to the Grounded Naming Game in the real world [31, 32].

Setting up such grounded language game experiments requires taking into consideration a set of processes that has been referred to as *the semiotic cycle* [25], and implementing each of the processes involved. During each game, the speaker and hearer move through this cycle as depicted in Figure 1. First, both agents perceive the world through their own sensors and construct an internal world model (*grounding*). Then, the speaker determines which information needs to be conveyed to the hearer in order for the task to succeed and conceptualises it into a semantic structure (*conceptualisation*). This meaning representation is then expressed through a linguistic expression that is passed to the hearer (*production*). The hearer then parses the utterance into a meaning representation (*comprehension*). He interprets the resulting semantic structure in relation with his world model and performs the relevant action (*interpretation*). Finally, the speaker provides feedback on the outcome of the

[1] Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium, email: ehai@ai.vub.ac.be

game, allowing both agents to update their individual knowledge. The described processes take place on three main levels: grounding on the sensori-motor level, conceptualisation and interpretation on the conceptual level, and production and comprehension on the language level.

There are a number of tools available that can be used for implementing language game experiments. A general-purpose, widely used platform is *NetLogo* [38], which was developed as an educational tool to teach students about agent-based modelling. It is mainly targeting the complex systems science community and contains a large number of sample models on many topics. NetLogo provides an excellent architecture for setting up and monitoring multi-agent simulations but does not contain any built-in functionalities for implementing the processes involved in the semiotic cycle.

NaminggamesAL is a recent tool for implementing a variety of basic naming games in simulated worlds [18]. It includes a multi-agent simulation framework and a number of built-in learning strategies, but lacks modules for implementing more advanced versions of the three levels of the semiotic cycle.

A software tool that stems from the linguistic community is *MoLE* (*Modelling Language Evolution*) [10]. MoLE focuses on the language level and was especially designed for conducting experiments on the emergence of case [9]. It includes the necessary building blocks for setting up multi-agent language games in which lexical items can be recruited as grammatical markers. It does not include an advanced semantic processing engine, an elaborate language processing engine, and interfaces to physical robots or rich world models.

Finally, *Babel* is a software package that was originally implemented as a testbed for research on the origins of language [13]. In its first version, it provided users with a basis for running computer simulations and allowed the rapid construction of experiments and a flexible visualisation of the results. Later versions of Babel (see [30, 12]) introduced more elaborate tools for setting up language games with advanced modules for dealing with the conceptual level (IRL – [23, 20]) and the language level (FCG – [24, 26]). Although Babel has often been used in grounded experiments, involving amongst others the AIBO dog-like robot [29], the QRIO humanoid [19] and the PERACT vision system [37], it has never included a standardised interface to connect to robotic platforms.

The contribution of this paper is twofold. On the one hand, it introduces a high-level robot interface that extends the Babel software system, presenting for the first time a toolkit that provides flexible modules for dealing with each subtask involved in running advanced grounded language game experiments. On the other hand, it provides a practical guide to using the toolkit for implementing such experiments, taking a grounded colour naming game experiment as a didactic example.

The remainder of this paper is structured as follows. Section 2 introduces the challenges involved in establishing a shared colour lexicon and discusses the grounded colour naming game as a solution. Section 3 serves as a practical explanation of how this solution can be implemented on a high level using the Babel system. Finally, Section 4 provides more technical detail on the architecture and main features of the newly developed robot interface.

2 EMERGENT COMMUNICATION FOR THE COLOUR DOMAIN

The goal of the colour naming game experiment is to show how a shared communication system for referring to objects by their colour can emerge in a population of autonomous agents. The agents start without any concepts or words, perceiving only the average colour values of the objects in the scene. In a real-world setting, transmitting raw sensor values does not lead to successful communication, because the sensors of each agent will always record slightly different values due to differences in the agents' perspectives on the scene, changes in lighting conditions and in some cases differences in robot morphology[2]. Therefore, concepts and words form the necessary layers to abstract away from sensor data, in order to achieve more robust communication.

A large body of previous work has shown how colour categories and words can emerge through self-organisation in a population of autonomous agents, including robots [27, 17, 4, 3, 6]. In essence, the solution resides in the agents dividing their continuous colour space into convex regions that correspond to colour categories that are functional in the world, and in establishing a shared lexicon to refer to each region. An operationalisation of this solution has been proposed in the form of *the grounded colour naming game experiment* [27].

Figure 2 shows an instantiation of a grounded colour naming game. In this game, the world consists of a number of toy monsters, each with a different colour. Two randomly selected agents from the population are physically embodied in the two robots, one playing the role of speaker and the other the role of hearer. The task of the speaker is to use a vocalisation to draw the attention of the hearer to one of the monsters in the world. The task of the hearer is to point to this monster, signalling that he has understood the message. Finally, the speaker signals success if the hearer pointed to the right monster, or points himself if this was not the case.

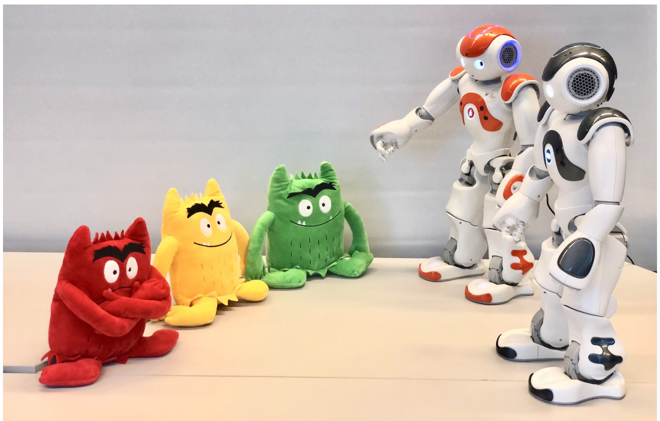


Figure 2: Two Nao robots play a grounded colour naming game with a scene consisting of coloured monsters. For each game, two agents from the population are physically instantiated in the two robot bodies.

Once the basic interaction script is in place, we can start experimenting with different mechanisms for inventing, adopting and aligning colour categories and words. Suppose that the orange robot in the back of Figure 2 needs to refer to the green monster in front of him. As he enters the experiment without any colour categories or words, he needs to invent both a new category and a new word to express this category. He takes the observed colour value as the first prototypical value of this new category (e.g. [(7, 246, 9) ↔ `CATEGORY-1`]) and associates the category with a newly generated

[2] Traditional sensor calibration is undesirable here, as it requires a notion of central control which conflicts with the autonomous nature of the agents.

word form, in this case "fusemo", assigning to the association a default initial score (e.g. [`CATEGORY-1` ↔ *fusemo*]). He then utters the word to the hearer. The hearer does not know this newly invented word and is therefore unable to determine which monster the speaker is referring to. The speaker provides feedback by pointing to the green monster. At that point, the hearer associates the colour value that he observed for this object to a new category (e.g. [(5, 243, 2) ↔ `CATEGORY-2`]) and associates this category to the word "fusemo" with a default initial score (e.g. [`CATEGORY-2` ↔ *fusemo*]). Crucially, each association between a sensor value and a category, as well as between a category and a word form is internal to the individual agent and cannot be shared as such.

While agents are able to invent new categories and words throughout the experiment, they will prefer to reuse existing ones. When a novel observation comes in, the agents will determine whether the category that is the closest in sensory distance to the observation discriminates the topic monster from the other monsters in the world. In other terms, they will calculate the distance between the observed sensory value and each of their categories, and select the closest one. Then, they verify that no other object is closer to this category, which means that the category uniquely discriminates the monster in the world. If no such category can be found, a new category is invented following the procedure explained in the previous paragraph. When it comes to the words, the speaker will always choose the word form most strongly associated with the selected category and the hearer will choose the category most strongly associated with the selected word form.

When agents invent, adopt and reuse colour categories and words as described above, the categories of the agents never align and their vocabularies become enormous in size. To overcome this issue, speaker and hearer go through an alignment phase at the end of each interaction. If the interaction succeeds, the agents reinforce the association between the category and the word form that was used and punish competing associations. Moreover, they also slightly shift the prototypical value of the used category towards the observed sensor value. If the game fails, the agents punish the association that was used.

Using these mechanisms for invention, adoption and alignment, a population of agents will eventually converge on a stable colour category system, and on a shared inventory of words for referring to these categories. Importantly, the emerged system is tailored towards the distinctions that are functional in the world, both in terms of number of categories and in the way in which the colour space is subdivided. The concrete mechanisms described in this section are the ones most commonly used in the literature. A complete overview of invention, adoption and alignment strategies that have been explored for the colour naming game falls outside the scope of this paper, but can be found in earlier work by Joris Bleys [3].

3 IMPLEMENTING A GROUNDED COLOUR NAMING GAME EXPERIMENT

We will now demonstrate how the Babel toolkit can be used to implement a grounded colour naming game experiment like the one that was introduced in the previous section. Babel's experiment framework and submodules for dealing with the sensori-motor level, conceptual level and language level provide abstractions that allow specifying the game on a high level and in an intuitive way, while most technical detail is taken care of by the system itself [3]. Actual source code that corresponds to the explanation in this section has become an integral part of the Babel toolkit[4], which can be obtained via `https://emergent-languages.org`. Additionally, an online web demonstration of the grounded colour naming game experiment is available at `https://ehai.ai.vub.ac.be/demos/babel-grounded-colour-naming-game-experiment`.

3.1 Multi-agent architecture

Implementing a language game experiment involves keeping track of the agents in the population, selecting the agents to participate in each game, assigning them the role of speaker or hearer, and, most importantly, specifying the language game script according to which the agents will interact. Within Babel, the multi-agent simulation part of the experiment is handled by the 'experiment-framework' submodule. The experiment framework is entirely customisable when it comes to how the population is structured, which and how many agents are selected for each interaction, how their role is determined and what an interaction looks like. For this grounded colour naming game experiment, we will make use of the experiment framework's default settings: a fully connected population structure, one speaker and one hearer per game, both randomly selected, and communicative success as a measure for evaluation. The interaction script itself is specified as shown in Listing 1 and consists of the following steps (with the Babel function names between parentheses):

1. Two agents are downloaded into the robot bodies (EMBODY)
2. Both agents scan the world and construct their world model (AGENT-OBSERVE-WORLD)
3. The speaker chooses an object to refer to (CHOOSE-TOPIC)
4. The speaker conceptualises the topic in relation to his world model (CONCEPTUALISE)
5. The speaker chooses a word for the topic (PRODUCE-UTTERANCE)
6. The speaker utters the word while the hearer is listening (PASS-UTTERANCE)
7. The hearer parses the observed word into a semantic structure (COMPREHEND-UTTERANCE)
8. The hearer interprets the semantic structure in relation to his world model (INTERPRET)
9. The hearer points to the hypothesized topic (POINT-AND-OBSERVE)
10. The speaker provides feedback by nodding (AGENT-NOD) in case of success, or pointing (POINT-AND-OBSERVE) in case of failure
11. Both agents align (ALIGN-AGENT)

EMBODY, AGENT-OBSERVE-WORLD, PASS-UTTERANCE, POINT-AND-OBSERVE and AGENT-NOD all happen at the sensori-motor level; CHOOSE-TOPIC, CONCEPTUALISE and INTERPRET at the conceptual level; and PRODUCE-UTTERANCE and COMPREHEND-UTTERANCE at the language level. Finally, ALIGN-AGENT takes place on both the conceptual and the language level.

[3] In this example experiment, all processes in the semiotic cycle are implemented using standard Babel modules. It is however perfectly possible to only use Babel modules for implementing some of these processes, and different software for the others.

[4] The complete source code for running the Grounded Colour Naming Game Experiment is part of the Babel toolkit and can be found in the subfolder `experiments/grounded-colour-naming-game-experiment`. A simulator mode has also been provided, to run the experiments if you do not have a Nao robot at your disposal.

Listing 1: Interaction Script

```
(defmethod interact ((experiment gcng-experiment) interaction &key)
  (let ((speaker (speaker interaction))
        (hearer (hearer interaction)))
    ;; 1
    (embody speaker (first (robots experiment)))
    (embody hearer (second (robots experiment)))
    ;; 2
    (agent-observe-world speaker)
    (agent-observe-world hearer)
    ;; 3
    (choose-topic speaker (world speaker))
    ;; 4
    (conceptualise speaker (topic speaker) (world speaker))
    ;; 5
    (produce-utterance speaker (meaning-representation speaker))
    ;; 6
    (pass-utterance speaker hearer (utterance speaker))
    ;; 7
    (comprehend-utterance hearer (observed-utterance hearer))
    ;; 8
    (interpret hearer (meaning-representation hearer))
    ;; 9
    (point-and-observe hearer (hypothesized-topic hearer))
    ;; 10
    (if (communicated-successfully interaction)
      (agent-nod speaker)
      (point-and-observe speaker (topic speaker)))
    ;; 11
    (align-agent speaker)
    (align-agent hearer)))
```

3.2 Sensori-motor level

The agents' action and perception capabilities are handled at the sensori-motor level by Babel's 'robot-interface' submodule, which is presented for the first time in this paper. The robot interface defines a standard set of functions that are particularly useful for conducting language games, for example scanning the robot's environment, speaking, listening and pointing. It abstracts away these high-level instructions from their specific implementation, which heavily depends on the hardware that is used, and is different for each type of robot. More technical detail can be found in Section 4 below, with an overview of the available functionality in Table 1.

In the example experiment presented in this paper, the robot interface makes use of the sensors and actuators of the Nao robotic platform. Concretely, the EMBODY step embodies the speaker and hearer agents into the available robots. The AGENT-OBSERVE-WORLD step uses the camera of the robot to make a picture of the scene, and uses the OpenCV library [5] to construct a world model by segmenting the scene and extracting certain features for the objects, including their average colour value. PASS-UTTERANCE lets one robot speak via text-to-speech while the other listens and performs speech recognition. POINT-AND-OBSERVE is used by the hearer to indicate the hypothesized object. Finally, either AGENT-NOD or POINT-AND-OBSERVE is used by the speaker at the end of the game to signal success or provide feedback.

3.3 Conceptual level

Bridging the gap between the world model and the meaning that needs to be expressed by the speaker or interpreted by the hearer is handled at the conceptual level by Babel's 'IRL' (Incremental Recruitment Language) submodule [23, 20]. IRL implements a form of procedural semantics, which means that semantic representations consist of primitive operations that directly correspond to actual function calls, and which can be combined into semantic networks for expressing more complex meanings. In conceptualisation, the IRL engine uses a search process to compose such a semantic network that singles out a given topic in the current scene. In interpretation, the IRL engine executes the semantic network by calling the functions underlying the primitive operations and propagating the resulting values.

Suppose that in this example experiment the speaker needs to refer to the green monster. CONCEPTUALISE will then trigger the IRL engine to compose the smallest possible semantic network that uniquely discriminates the object by its colour, relying on the agent's ontology. As the present experiment is only concerned with basic colour categories, the semantic network will always consist of a single FILTER-BY-CLOSEST-COLOUR operation, in this case using CATEGORY-1, as shown in Figure 3. On the hearer's side, INTERPRET calls the IRL engine to execute the semantic network that results from the comprehension process, also consisting here of a single FILTER-BY-CLOSEST-COLOUR operation, in order to retrieve the topic object. During the alignment phase at the end of a successful game, the prototypical value of the used categories in the speaker's and hearer's ontologies are updated by slightly shifting them towards the values that were observed in this game.

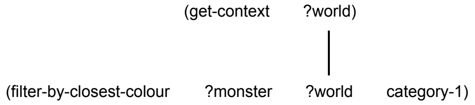


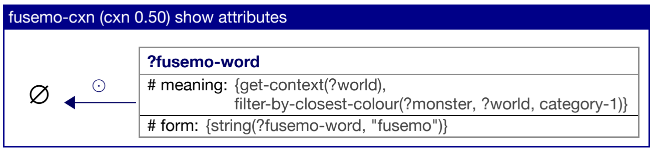
Figure 3: The speaker's semantic network that singles out an object discriminating it by CATEGORY-1.

For didactic purposes, only the very basic functionality of IRL is used in this experiment. For experiments that necessitate more complex semantic structures, we refer the reader to earlier work by Spranger (on spatial language) [21], Bleys (on colour) [3] and Pauw (on quantification) [16].

3.4 Language level

The task of mapping between a semantic structure and an utterance is taken care of by Babel's 'FCG' (Fluid Construction Grammar) submodule [24, 26]. FCG performs this mapping based on emergent form-meaning pairings, in this context called constructions. On the form side, a construction can include any form-related features, such as word forms, morphological properties and word order constraints. On the meaning side, it can include any type of semantic information, for example (parts of) a semantic network composed at the conceptual level.

In the example above, the speaker invented the word "fusemo" to refer to the colour of the green monster. He will use FCG to create a new construction that maps between this word form and the semantic network that was the outcome of the conceptualisation process. The construction is initialised with a default entrenchment score of 0.50, as illustrated by Figure 4. As the hearer had never heard this word before, after feedback he will create his own construction that maps between the observed word form "fusemo" and the semantic network that results from conceptualising in his world model the object that was pointed at. If the necessary constructions are already in place, the speaker uses PRODUCE-UTTERANCE to find the word form most strongly associated to his meaning network and the hearer uses COMPREHEND-UTTERANCE to retrieve the meaning network most strongly associated to the word form that he observed.


fusemo-cxn (cxn 0.50) show attributes

∅ ← ⊙

?fusemo-word

meaning: {get-context(?world), filter-by-closest-colour(?monster, ?world, category-1)}

form: {string(?fusemo-word, "fusemo")}

Figure 4: The speaker's construction that maps between the word form "fusemo" and its meaning.

During the alignment phase after a successful game, both agents will increase the entrenchment score of the constructions that they have used. The agents will also decrease the score of competing constructions, i.e. constructions that map the same meaning to other word forms in the case of the speaker, and constructions that map the same word form to other meanings in the case of the hearer. After a failed game, both agents decrease the score of the constructions that were applied. When the score of a construction reaches zero, the construction is removed from the agent's inventory.

The constructions that are used in this didactic example are always direct mappings between a single word form and a complete meaning network. Moreover, a single construction always suffices to comprehend and produce an utterance. Examples of experiments that involve more complex linguistic structures can for example be found in previous work by Garcia Casademont (on hierarchical structures) [8] and Beuls (on grammatical agreement) [2].

3.5 Running and monitoring experiments

The Babel toolkit comes with a 'monitors' submodule that is designed to track a multitude of experimental parameters in real time. The data recorded during a series of experiments can be displayed using dynamically updating graphs or exported to data files for later data exploration. Experimental parameters that are typically tracked include communicative success, the number of constructions in the inventories of the interacting agents, the categories in their ontologies and the size of the semantic (IRL) and syntactic (FCG) search spaces.

Figure 5 shows a graph that was created by the monitoring system during a single experimental run of the grounded colour naming game experiment. There were five agents in the population, communicating about six distinctly coloured monsters of which three were shown during each game. The x-axis represents the time dimension, indicating the total number of games that were played. The turquoise line indicates the average communicative success that was achieved over the last fifty games (left y-axis). At the beginning of the run, the communicative success equals zero as the agents start without any categories or words. Over the course of 250 games, it rises to 1, as the emerged communication system becomes powerful enough to solve the task. The ontology size (red line), i.e. the average number of categories per agent, starts at zero and goes to six in just over 100 games (right y-axis). This number is optimal for this experimental set-up, as there are indeed six colour distinctions that are useful in the world. The average lexicon size (dark yellow line) clearly shows how the agents locally introduce new words (leading to 13 different forms), before gradually converging on the optimal number of six words, one for each category (right y-axis). The blue line tracks the average number of forms per meaning in the population (right y-axis). In the phase in which many words are being invented, this number reaches its maximum, after which it gradually declines to a single form for each meaning. The green line shows the opposite, namely the average number of meanings per form (right y-axis). While the agents are still building up their ontologies, it can happen that two word forms get associated to the same meaning. As an effect of alignment, the meaning-per-form ratio gradually decreases to 1.

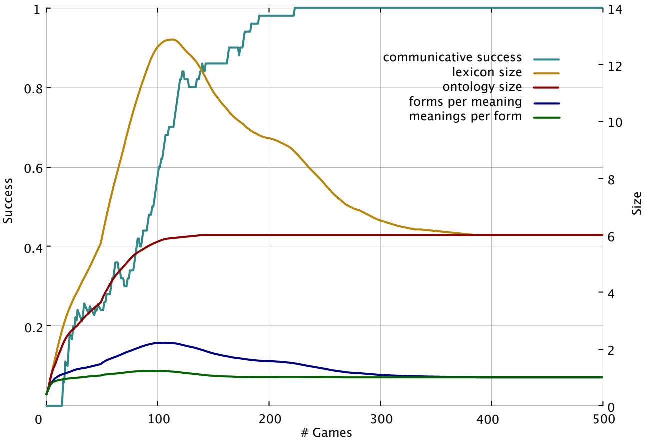


Figure 5: A graph showing the results of monitoring a single experimental run of the grounded colour naming game experiment. The population consists of five agents that develop a communication system to refer to six distinctly coloured monsters.

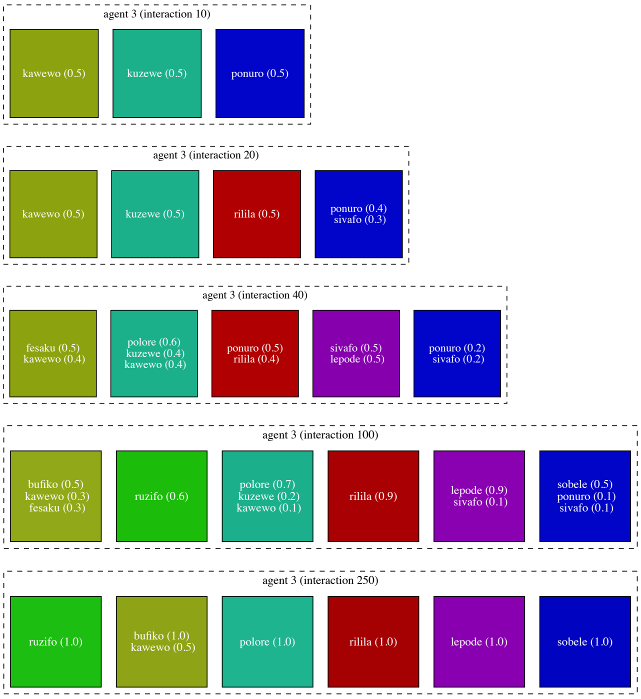


Figure 6: A visualisation of agent 3's colour lexicon after 10, 20, 40, 100 and 250 games. Note that the word "ponuro" was first exclusively used to refer to blueish objects (interaction 10) and was later also used to refer to reddish objects (interaction 40). In the end, the form did not survive, as the population converged on "rilala" and "sobele" for referring to reddish and blueish objects respectively (interaction 250).

A different kind of visualisation that was created using the monitoring system is presented in Figure 6. Each row in the figure shows a snapshot of single agent's ontology of colour categories and their associated word forms at a specific point in time. In this case, the ontology and word forms of agent 3 are shown after 10, 20, 40, 100 and 250 interactions. We can see how the agent gradually distinguishes more colour categories, until he reaches the optimal number of six. We can also observe that after 100 interactions, the agent has learned multiple words for certain colour categories, but that most have already disappeared after 250 interactions. The rise and fall of the word "ponuro" is particularly interesting. It was first exclusively used to refer to blueish objects (interaction 10), was then also associated to the colour category used to refer to reddish objects (interaction 40), but dies out as the population has converged on the words "ribala" and "sobele" to refer to reddish and blueish objects respectively (interaction 250).

Babel's monitoring system can in real-time track, aggregate and visualise series of experiments that are run in parallel, and can easily be extended to record other experimental parameters or measures.

4 THE ROBOT INTERFACE: TECHNICAL SPECIFICATION

The robot interface is a newly developed part of the Babel software system, facilitating the implementation of processes that take place at the sensori-motor level of the semiotic cycle. It allows Babel users to seamlessly integrate the use of physical robot bodies in their language game experiments, by providing a hardware-independent interface to the functionality that is most frequently used in language games. This section first gives an overview of the general architecture of the robot interface, and then describes how it can be concretely used in combination with the Nao hardware that was employed in the experiment reported in the previous section.

4.1 General architecture

The robot interface standardises a number of core capabilities that can be exerted by a wide range of robotic platforms, abstracting away from their low-level implementation details. An overview of the most relevant capabilities, such as speaking, listening and pointing, is presented in Table 1.

Table 1: Selected functions from the Babel robot interface API.

Function	Arguments	Return value
MAKE-ROBOT	ip-address, port, type	robot-connection
OBSERVE-WORLD	robot-connection	world-model
SPEAK	robot-connection, utterance	boolean
HEAR	robot-connection	perceived-utterance
POINT	robot-connection, arm	boolean
NOD	robot-connection	boolean
SHAKE-HEAD	robot-connection	boolean
LOOK-DIRECTION	robot-connection, dir, angle	boolean

In order to be able to use the robot interface, a robot-connection object of a specific type needs to be created first. This is done using the function MAKE-ROBOT, which takes an IP address, a port number and a type of robot (e.g. 'nao') as input and returns a robot-connection object specialised towards this type of robot. Each of the available capabilities is then implemented as a Common Lisp generic function, with methods specialising on the subtype of the robot-connection object. This means for instance that when a certain capability is called with a robot-connection of type 'nao' as its first argument, the call will automatically be dispatched to the method that implements this capability specifically for the Nao robot. A didactic example of how such a capability can be implemented is shown in Listing 2. The example shows how the general SPEAK capability is implemented as a generic function, while a call to this function with as first argument a connection of type 'nao' will automatically be routed to the method just below. This general architecture ensures that the robot interface is easily extensible, both in terms of adding additional functionality and in terms of extending the existing functionality to different robotic platforms.

Listing 2: Didactic example of the implementation of the speaking capability on a Nao robot.

```
(defgeneric speak (robot-connection utterance)
  (:documentation ``The robot says the utterance.''))

(defmethod speak ((nao nao)
                  (utterance string))
  ``Sending the utterance to the Nao's speech endpoint,
      returning a boolean that indicates success or
      failure.''
  (rest (assoc :success
           (nao-send-data nao
                          :endpoint ``/speech/say''
                          :data `((speech . ,utterance))))))
```

When setting up a grounded language game experiment like the one reported on in this paper, it suffices to create one robot-connection object per robot body, at the beginning of the experiment. At the start of each communicative interaction, the EMBODY step (see Section 3.1) will then assure that the speaker and hearer agents sense and act through the right robot body during this game, by associating them to one of these robot-connection objects. This avoids opening and closing a connection for every interaction.

4.2 Using the Nao robot

The experiment described earlier in this paper used the robot interface to play grounded colour naming games using two humanoid robots of the Nao type[5]. Nao robots run a GNU/linux-based operating system, called *NaoQi OS*, and can be controlled from an external computer using the *NaoQi framework*, available either as a C++ or a Python library.

On the computer that runs the Babel software system, we set up a Docker container running a Python (Flask) server. This server exposes a RESTful API that continuously listens to HTTP requests, transforming them into concrete instructions that are passed to the right Nao robot using the Python version of the NaoQi framework. When a function from the Babel robot-interface API is called during an experiment, an HTTP POST request containing the necessary information is sent to the Python server's endpoint that handles the capability associated to this function. Suppose for example that the function SPEAK is called with as arguments a robot-connection object and an utterance. Babel's robot interface will then send an HTTP POST request containing a JSON object holding the IP address and port of the Nao associated to the robot-connection object, as well as the utterance to pronounce, to the `/speech/say` endpoint of the Python server running in the Docker container. The Python server will parse the request and call a function from the NaoQi framework that makes the robot say the utterance and will return a JSON object containing a key 'success' with a boolean value. A visual depiction of this system architecture is shown in Figure 7.

[5] https://www.softbankrobotics.com/emea/en/nao

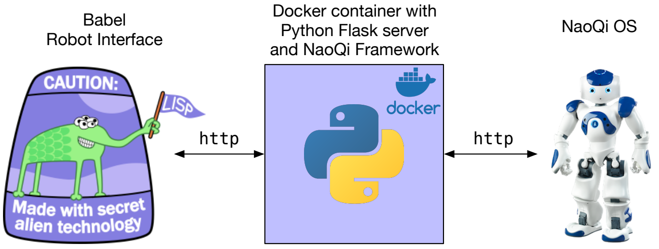


Figure 7: When a function of the Babel robot interface API is called during an experiment, an HTTP POST request is send to a Python server running in a Docker container. The Python server then uses the NaoQi framework to communicate the request to the Nao robot.

5 CONCLUSION

Grounded language game experiments form an excellent tool to study emergent communication and its underlying mechanisms. Setting up such experiments, in which a rich system for communicating about the real word emerges, requires implementing each process involved in the semiotic cycle, encompassing the sensori-motor, conceptual and language levels. This paper has introduced a high-level interface that allows making use of physical robots for operationalising the grounding processes on the sensori-motor level. This interface has been fully integrated into the Babel software system, which, as a result, now includes software modules that facilitate the implementation of all processes involved in the semiotic cycle. This paper has also presented a practical guide to using the Babel toolkit for setting up full-cycle experiments, taking the grounded colour naming game as didactic example.

ACKNOWLEDGEMENTS

We would like to thank Remi van Trijp, Katie Mudd and Yannick Jadoul for their valuable comments on earlier versions of this paper. We are also grateful to the three anonymous reviewers of the AISB Symposium on Grounded Language Learning for Artificial Agents for their encouraging feedback and appreciation. This work was supported by the Research Foundation Flanders (FWO) through grants 1SB6219N and G0D6915N (CHIST-ERA ATLANTIS).

REFERENCES

[1] Andrea Baronchelli, Maddalena Felici, Vittorio Loreto, Emanuele Caglioti, and Luc Steels, 'Sharp transition towards shared vocabularies in multi-agent systems', *Journal of Statistical Mechanics: Theory and Experiment*, **2006**(06), P06014, (2006).

[2] Katrien Beuls and Luc Steels, 'Agent-based models of strategies for the emergence and evolution of grammatical agreement', *PloS one*, **8**(3), e58960, (2013).

[3] Joris Bleys, *Language strategies for the domain of colour*, Language Science Press, 2016.

[4] Joris Bleys, Martin Loetzsch, Michael Spranger, and Luc Steels, 'The grounded colour naming game', in *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (Ro-man 2009)*, (2009).

[5] G. Bradski, 'The OpenCV Library', *Dr. Dobb's Journal of Software Tools*, (2000).

[6] Miquel Cornudella, Thierry Poibeau, and Remi van Trijp, 'The role of intrinsic motivation in artificial language emergence: a case study on colour', in *26th International Conference on Computational Linguistics (COLING 2016)*, pp. 1646–1656, (2016).

[7] Bart De Vylder and Karl Tuyls, 'How to reach linguistic consensus: A proof of convergence for the naming game', *Journal of theoretical biology*, **242**(4), 818–831, (2006).

[8] Emilia Garcia Casademont and Luc Steels, 'Insight grammar learning', *Journal of Cognitive Science*, **17**(1), 27–62, (2016).

[9] Sander Lestrade, 'The emergence of argument marking', in *The Evolution of Language: Proceedings of the 11th International Conference (EVOLANGX11)*, eds., S.G. Roberts, C. Cuskley, L. McCrohon, L. Barceló-Coblijn, O. Fehér, and T. Verhoef. Online at `http://evolang.org/neworleans/papers/36.html`, (2016).

[10] Sander Lestrade. Mole: Modeling language evolution. `https://CRAN.R-project.org/package=MoLE`, 2017.

[11] Martin Loetzsch and Michael Spranger, 'Why robots?', in *Proceedings of the 8th International Conference on the Evolution of Language (EVOLANG 8)*, eds., Andrew D.M. Smith, Marieke Shouwstra, Bart de Boer, and Kenny Smith, pp. 222–229. World Scientific, (2010).

[12] Martin Loetzsch, Pieter Wellens, Joachim De Beule, Joris Bleys, and Remi van Trijp, 'The babel2 manual', *Technical Report AI-Memo 01-08*, (2008).

[13] Angus McIntyre, 'Babel: A testbed for research in origins of language', in *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*, COLING '98, pp. 830–834, Stroudsburg, PA, USA, (1998). Association for Computational Linguistics.

[14] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations, 2017.

[15] Pierre-Yves Oudeyer and Frédéric Kaplan, 'Discovering communication', *Connection Science*, **18**(2), 189–206, (2006).

[16] Simon Pauw and Joseph Hilferty, 'The emergence of quantifiers', *Experiments in cultural language evolution*, **3**, 277, (2012).

[17] Andrea Puglisi, Andrea Baronchelli, and Vittorio Loreto, 'Cultural route to the emergence of linguistic categories', *Proceedings of the National Academy of Sciences*, **105**(23), 7936–7940, (2008).

[18] William Schueller, *Active Control of Complexity Growth in Language Games*, Ph.D. dissertation, University of Bordeaux, 2018.

[19] Michael Spranger, Martin Loetzsch, and Luc Steels, 'A perceptual system for language game experiments', in *Language Grounding in Robots*, eds., Luc Steels and Manfred Hild, 89–110, Springer, (2012).

[20] Michael Spranger, Simon Pauw, Martin Loetzsch, and Luc Steels, 'Open-ended Procedural Semantics', in *Language Grounding in Robots*, eds., L. Steels and M. Hild, 153–172, Springer, (2012).

[21] Michael Spranger and Luc Steels, 'Emergent Functional Grammar for Space', in *Experiments in Cultural Language Evolution*, ed., L. Steels, number 3 in Advances in Interaction Studies, 207—232, John Benjamins, (2012).

[22] Luc Steels, 'A self-organizing spatial vocabulary', *Artificial life*, **2**(3), 319–332, (1995).

[23] Luc Steels, 'The emergence of grammar in communicating autonomous robotic agents', in *ECAI 2000: Proceedings of the 14th European Conference on Artificial Life*, ed., W. Horn, pp. 764–769, Amsterdam, (August 2000). IOS Publishing.

[24] *Design Patterns in Fluid Construction Grammar*, ed., Luc Steels, John Benjamins, Amsterdam, 2011.

[25] Luc Steels, 'Grounding language through evolutionary language games', in *Language Grounding in Robots*, 1–22, Springer, (2012).

[26] Luc Steels, 'Basics of Fluid Construction Grammar', *Constructions and frames*, **9**(2), 178–225, (2017).

[27] Luc Steels and Tony Belpaeme, 'Coordinating perceptually grounded categories through language: A case study for colour', *Behavioral and brain sciences*, **28**(4), 469–488, (2005).

[28] Luc Steels and Manfred Hild, *Language Grounding in Robots*, Springer, Berlin, 2012.

[29] Luc Steels and Frédéric Kaplan, 'Aibo?s first words: The social learning of language and meaning', *Evolution of communication*, **4**(1), 3–32, (2000).

[30] Luc Steels and Martin Loetzsch, 'Babel: A tool for running experiments on the evolution of language', in *Evolution of communication and language in embodied agents*, 307–313, Springer, (2010).

[31] Luc Steels and Martin Loetzsch, 'The grounded naming game', *Experiments in cultural language evolution*, **3**, 41–59, (2012).

[32] Luc Steels, Martin Loetzsch, and Michael Spranger, 'A boy named sue', *Belgian Journal of Linguistics*, **30**(1), 147–169, (2016).

[33] Luc Steels and Angus McIntyre, 'Spatially distributed naming games', *Advances in complex systems*, **1**(04), 301–323, (1998).

[34] Joris Van Looveren, 'Multiple word naming games', in *Proceedings of the 11th Belgium-Netherlands Conference on Artificial Intelligence. Universiteit Maastricht, Maastricht, the Netherlands*, (1999).

[35] Remi van Trijp, 'The emergence of semantic roles in Fluid Construction

Grammar', in *The Evolution Of Language*, 346–353, World Scientific, (2008).

[36] Remi van Trijp, 'Linguistic assessment criteria for explaining language change: A case study on syncretism in german definite articles', *Language Dynamics and Change*, **3**(1), 105–132, (2013).

[37] Remi van Trijp, *The evolution of case grammar*, Language Science Press, 2016.

[38] Uri Wilensky. Netlogo. `http://ccl.northwestern.edu/netlogo/`, 1999.