ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-929

Open-ended Grounded Semantics

Michael Spranger[1] and **Martin Loetzsch**[2] and **Simon Pauw**[1]

Abstract. Artificial agents trying to achieve communicative goals in situated interactions in the real-world need powerful computational systems for conceptualizing their environment. In order to provide embodied artificial systems with rich semantics reminiscent of human language complexity, agents need ways of both conceptualizing complex compositional semantic structure and actively reconstructing semantic structure, due to uncertainty and ambiguity in transmission. Furthermore, the systems must be open-ended and adaptive and allow agents to adjust their semantic inventories in order to reach their goals. This paper presents recent progress in modeling open-ended, grounded semantics through a unified software system that addresses these problems.

1 INTRODUCTION

In the past ten years, computational experiments have demonstrated how to ground language in the real world via perception and action, as well as culturally in populations of agents. In these experiments agents embodied in real robots engage in communicative interactions about things in the physical world. Tremendous progress has been made in explaining the coordination of sensorimotor categorical systems and their co-evolution with language [28, 24, 29, 30, 18, 4]. These studies convincingly demonstrate that open-ended, grounded communication is possible and that the resulting systems are resilient to perceptual noise, changes in the environment and perturbations of the structure of the agent community.

However, most of this work focuses on simple utterances consisting of one word or multiple words, without syntactic structure. But natural language obviously is more complex than mere bags of words. Rather, human language is compositional: the syntactic structure of an utterance (besides the lexical meaning of all items in it) encodes semantic structure and hence is itself meaningful. For instance, interpreting the utterance "the yellow block to the right of you" requires not only decoding the words for the categories involved but also understanding what to do with them (e.g. transforming spatial perspective). Acknowledging this fact, there is now a rich body of research that goes beyond the formation of purely lexical communication systems and focuses on the self-organization of grammar, e,g by using *Fluid Construction Grammar* [6, 27] as a formalism for representing, processing and learning linguistic knowledge (see e.g [22] and [33] for investigations into the emergence of case grammars for expressing argument structure and [9] for self-organization of grammars for aspect).

These approaches, while taking the complexity of grammar seriously, fall short in two ways. First, they largely neglect, i.e. scaffold, the problem of grounding representations in the real world, a necessary requirement for human-robot or robot-robot interaction. But secondly, they consider the domain of semantics static and fixed. That is, while grammar and the syntactic system of language are seen as adaptive and open-ended systems, meaning or semantic structure is not. Now, if one sees language as an open-ended, adaptive system, why would one stop and only consider the translation of semantic structure into syntactic as an adaptive system? The reason is that the necessary tools linking language to embodiment in the right level of complexity are missing.

In this paper we report on the latest progress towards open-ended, grounded semantics using a fully operational computational system called *Incremental Recruitment Language* (IRL). We will discuss its design choices using concrete examples from different semantic domains, stressing its application in language production and interpretation on artificial systems interacting in the real world. The work presented here bases itself on substantial previous work. Key ideas of the IRL system have been laid out by [21], and a first version of the system was introduced by [25] with further advancements reported in [32]. This article presents the current state of IRL and focuses on its first widespread application to different semantic domains and interaction scenarios.

2 SITUATED INTERACTION

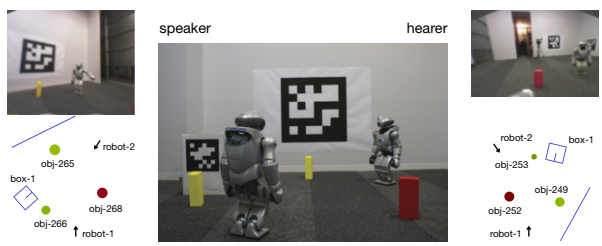

Figure 1. Example scene. Two robots autonomously perceive and act in an office environment that contains different types of objects. Both robots autonomously create world models reflecting the state of the environment (see bottom left and right schematics), that include objects with spatial and color properties, the carton boxes as well as the robots.

The notion of situated interactions and specifically of language games has come to be a prime vehicle for researching communication in robotic systems. Language games are interactions of two or more agents in the real world, with one of the agents having a particular communicative goal, for instance that the other agent points to an object, agrees with the description of an event, performs an action or a complex sequence of actions and so on. In this model of communication agents use language to achieve certain ends. Hence,

[1] SONY Computer Science Laboratory Paris, 6, Rue Amyot, 75005 Paris, France, email: spranger@csl.sony.fr
[2] AI-Lab, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium

language is seen as a tool and its development, adaptation and acquisition are based on the concrete usage scenarios agents face in the real world. Consequently, situated interactions necessitate a whole systems approach, in which a multitude of systems typically studied in isolation, such as perception, action, semantics and syntax, are operationalized and orchestrated in the right way. Another important feature of such a setup is that noise and uncertainty are immediately part of the problem and not a mere afterthought.

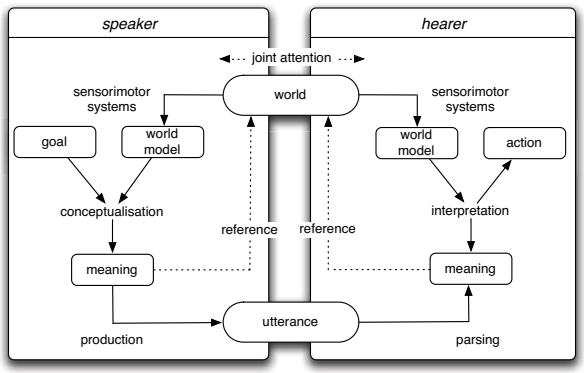

Figure 2. The semiotic cycle is a model of situated communicative interactions between two interacting agents.

Figure 1 shows an example of an encounter of two artificial agents (we use humanoid robots [7]) that engage in a situated interaction. Figure 2 shows the necessary processing steps for agents in such a communicative interactions. One of the agents is the speaker, the other the hearer. Both agents independently process sensorimotor data stemming from the onboard cameras and proprioceptive sensors in order to construct world models of the environment [20]. Based on the particular communicative goal and the current state of the world represented in the world model, the speaker conceptualizes a meaning which is then rendered into an utterance by the language system. The hearer parses the utterance to determine its meaning and interprets it with respect to his current model of the world in order to infer the speaker's communicative goal and, for instance, perform a desired action.

3 GROUNDED PROCEDURAL SEMANTICS

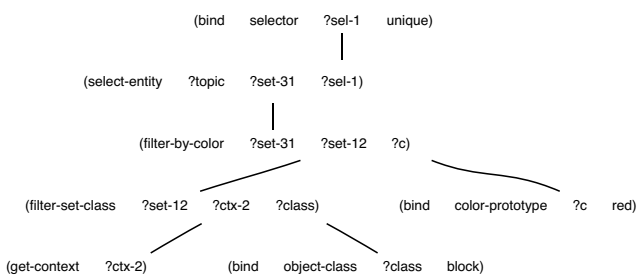

Figure 3. Semantic structure underlying the utterance *the red block*.

In order for a hearer to interpret an utterance, he has to apply the meanings conveyed in the linguistic structure to his perception of the context of the interaction. Consequently, a speaker who uses language to achieve a certain communicative goal wants the hearer to execute a program [10], i.e. a set of operations that allow the hearer to, for example, discriminate an object in the environment or perform an action. Thus we model semantics, i.e. what it is a speaker wants the hearer to execute, as a program linking operations and data.

Let us start with an example. The meaning of the utterance *the red block* is most likely a set of operations that will lead a hearer of this utterance to first filter the context for blocks, followed by the application of the color category red, in order to arrive at the set of red blocks, which should only consist of a single entity. A possible program, also called an *IRL-network*, is shown in Figure 3. This network explicitly represents the chain of the four operations `get-context`, `filter-set-class`, `filter-by-color` and `select-entity` by linking their arguments through variables (starting with `?`). Semantic entities, which represent concepts, prototypes or selectors (like `unique` in this network) are introduced into the network with *bind statements*, as in `(bind color-prototype ?c red)`.

When such a program is *evaluated*, for instance by a speaker to test the semantic structure with respect to the particular communicative goal of the speaker, or by a hearer in order to interpret an utterance, the following happens. First `get-context` gets the current world model from the perceptual processes that are constantly monitoring the environment for events and objects and *binds* it to the variable `?ctx-2`. This is followed by the evaluation of the `filter-set-class` operation, which filters the objects in the context using the class `block` to yield a set which contains all the blocks in the environment. This set is bound to the variable `?set-12`. This variable is then input to the operation `filter-by-color`, which yields the set of red objects from the input set. Hence, in `?set-31` now are all red blocks. Lastly, `select-entity` checks whether in `?set-31` there is only a single object, and if that is the case, will bind that object to the variable `?topic`, which is the referent of the phrase *the red block*. Note that the word "object" here refers to an agent's private representation of things he has perceived in the world and only indirectly refers to the physical object which is the referent.

The story just told is not quite complete. As has been argued elsewhere [21], language requires that semantic structure does not encode control flow, but rather that data flows in all directions and is computed where possible. For this, operations need to be able to function in different directions. For instance, the operation `filter-set-class` which has three arguments, computes a set of blocks when given a set and the object class block. It is also, however, able to compute, given two sets, the object class most likely to transform one set into the other. Moreover, when only passed a set, it will compute pairs of object classes and sets that encode possible segmentations of the input set using all object classes known to an agent. This *multidirectionality* of operations proves important for dealing with missing items, for instance due to partial parsing of an utterance, but it is also needed when constructing semantic structure.

Another important issue, not discussed so far, is that of grounding. There are now many proposals of how agents can ground lexicons and categorical systems in sensorimotor interaction with the environment [3, 34, 23] and the mechanism discussed in this paper are designed to allow such insights to be applied straightforwardly. For instance, the implementation of the operation for `filter-by-color` is based on recent findings about how basic

color categories can be grounded in the sensor data streams of digital cameras [24, 4]. Here, color categories are represented as prototypical points in color space and filtering a set of objects for a specific color amounts to finding all objects that are closest to that category in terms of their distance in the color space. Similarly, other grounding mechanisms such as for events [16, 2] are easily instantiated in IRL operations. However, note that IRL is agnostic as to what specific cognitive operations are implemented as semantic building blocks – IRL itself doesn't provide any cognitive operations but rather provides mechanisms for combining them in compositional semantic structures.

4 COMPOSITION

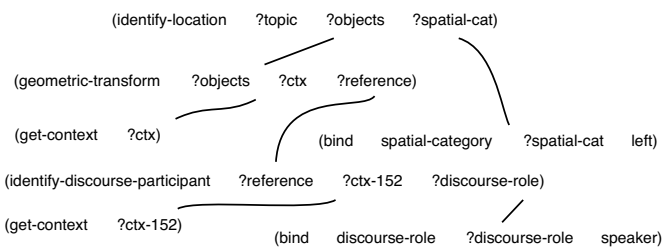

Figure 4. Semantic structure potentially underlying the utterance *left of you*. This network consists of four different operations: `get-context` introduces the current state of the world into the network, `identify-discourse-participant` picks out the hearer robot from the context, `geometric-transform` geometrically transforms the complete context to the perspective of the hearer robot, and `identify-location` applies the spatial category `right` and computes the leftmost object from the perspective of the hearer.

There are two scenarios in which agents autonomously compose semantic structure like that just described. In the first one, speakers have a particular communicative goal and need to construct semantic structure which is then processed by the language engine to compute an utterance. This process is called *conceptualization*. In the second scenario, hearers use their knowledge about the current context of the interaction to actively reconstruct meanings from the potentially partial structures parsed by the language system. We call this process *interpretation*. Both cases are equally important and they both conceive the process of building semantic structure as a heuristically guided search process that explores the space of possible IRL-networks and is driven by the agent's particular goal, for instance in conceptualization to discriminate a certain object.

In conceptualization, in other words while "planning what to say" [26], a speaker searches for an IRL-network that, when executed by the hearer, will reach a particular given communicative goal in a particular context. IRL-programs are constructed in an approach quite similar to genetic programming [11]. The basic building blocks are IRL-programs packaged into chunks and the search process progressively combines chunks, hence IRL-programs, into more and more complex semantic structures. Each built structure is immediately tested for compatibility with the communicative goal of the speaker (recall we are describing conceptualization), as well as the context. Figure 5 shows an example of such a search process that has produced the program in Figure 4 for discriminating the red block in Figure 1. This structure could be expressed by the utterance *left of you* (which actually will prove problematic as we will see later).

The search process for 'good' semantic structure is guided by many different heuristics, one being that the structure can be expressed using the language system available to an agent. Others are more focused on the particular character of the communicative goal. If the goal is to discriminate an object or event in the environment, then it is beneficial to use more discriminative categories, i.e. categories that enlarge the distance between the topic and all other objects in the context. Let us consider spatial language semantics, where it has been shown that not only are there different ways of conceptualizing spatial reality [12], but humans have a strong tendency to combine discriminating spatial categories with salient landmarks in order to construct utterances such as *left of you* [5].

Such principles can be implemented in IRL via scoring mechanisms that are tailored to particular communicative goals. For instance, the operation `identify-location` computes a location from a source set and a category and assigns the result a score which is based on how close (or similar) the identified location is to the category prototype, compared to all other objects in the source set This score, also called *discrimination score*, thus reflects how discriminating the category is. The search process will then prefer those semantic structures that yield highly discriminating categories. The same holds for landmark objects, whose saliency also can be marked using scores. IRL makes no specific claims about how to score semantic structure; in fact, it is up to the user of IRL to implement these heuristics. IRL is agnostic as to whether it can be used in discrimination scenarios, where language is used to discriminate an object in the environment. Other scenarios are conceivable and will mainly differ in the particular scoring mechanism applied. In fact many different scoring mechanisms can happily coexist in agents.

Search is also applied in interpretation. If an agent is parsing an utterance, then the same process of searching semantic structure is used by the hearer to reconstruct the meaning, i.e. the semantic structure the speaker had in mind. There are three principle reasons as to why interpretation is a search process. First, language transmission is inherently noisy. Language parsing in the real world is subject to noise and words might be rendered unparsable and consequently parts of the network might be missing. Second, especially in open-ended interaction scenarios, an agent might not know all the words and constructions used by the speaker, which also leads to partial semantic structures. And third, human language is in many cases ambiguous. Already the seemingly unproblematic utterance *left of you* has two possible semantic structures underlying it. One is *left of you from your perspective*, which is conceptualizing the hearer as an *intrinsic* frame of reference. The other is *left of you from my perspective*, which uses the hearer as a *relative* frame of reference [12]. Consequently, these two possibilities are represented by two different IRL-networks. For the first option the network is depicted in Figure 4. The second possible interpretation can be obtained from that structure by replacing the `geometric-transform` operation, which takes the robot and transforms the context using the robot as intrinsic landmark, by the operation `geometric-transform-from-viewpoint`, which implements a relative transformation. However, looking at the scene in Figure 1, the second interpretation does refers to one of the yellow blocks `obj-249` (in the hearer's world model). The first interpretation refers to the red block `obj-252` (in the hearer's world model). Whether or not the interpretation of semantic structure is ambiguous in terms of reference depends on the particular communicative context. For instance, when both robots view the scene from a similar perspective both intrinsic and relative readings of the phrase might lead to the same referent. However, it should be clear from the exam-

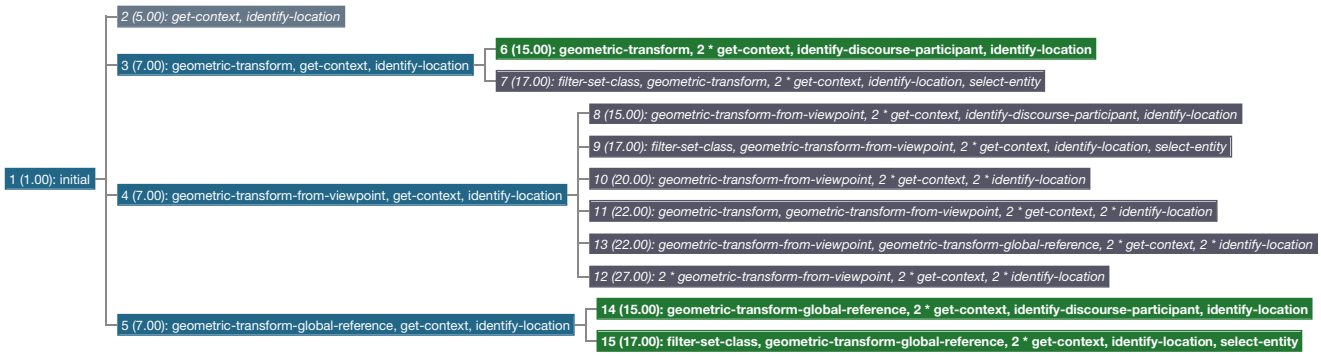

Figure 5. The search tree involved in conceptualizing the semantic structure seen in Figure 4. From left to right, nodes represent progressively growing programs combined from several chunks, which are each tried out and in some cases lead to solutions (green nodes).

ple that ambiguity is a ubiquitous feature of natural language.

IRL deals with ambiguity in a uniform way. Whatever structure the language system is able to parse, IRL will actively try and reconstruct sensible semantic structure that adheres both to the semantic structure provided by the language system and the particular communicative context. The same scoring mechanisms as for conceptualization will ensure that for example only structure that is discriminating for a particular object (implicitly assuming that the speaker constructs structure based on these principles) will be considered and the best of all possible results is chosen as the interpretation of an utterance.

5 OPEN-ENDED SEMANTICS/ LEARNING AND ADAPTATION

Search spaces quickly become intractable when multiple semantic domains such as tense, aspect, mood and so forth are combined, because the number of possibilities for composing semantic structures increases exponentially with the number of cognitive operations involved. But grammar can help here, because it is a sophisticated tool that highly structures human language in order to manage not only the search space of possible syntactic structure [31] but perhaps more importantly the vast space of possible conceptual structures. Parts of meaning that are covered by a particular construction of a language can be stored as a *chunk* and from then on be used as an atomic unit in composition. From this perspective grammar reflects a deep semantic bias towards using certain semantic structures, one of the main claims in cognitive linguistics (see [13] for insightful investigations into the cultural diversity of spatial conceptualization).

For example, in Russian every verb is marked by an Aktionsart and tense. In IRL, a linguistic construction for verbalizing events would be accompanied by a chunk that includes the temporal relation to the moment of speaking or hearing (tense), as well as the highlighting of a specific portion of an event by an Aktionsart (see Figure 6 for an example).

There are two main reasons why using chunks is beneficial. The first is that chunks can interact tightly with grammatical structure. As another example, the network in Figure 7 reflects the meaning constraints for a transitive construction, which essentially needs some agent, patient and event (the verb). This structure does not, however, constrain what the particular agent, patient or event is that fills the open variables. Figure 8 shows a network that was constructed using such a 'transitive' chunk by combining it with other chunks, for instance the semantic structure of a determined adjective noun phrase

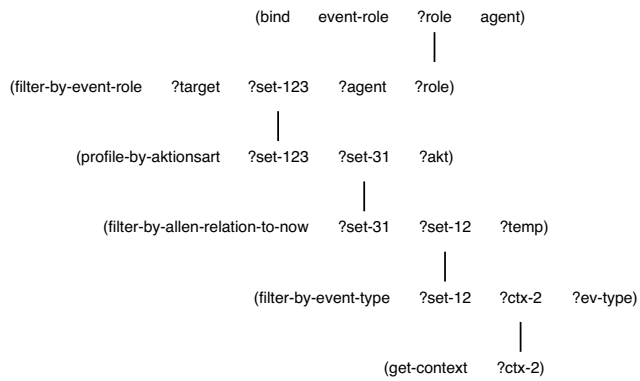

Figure 6. A chunk that contains a network of operations underlying intransitive verbs in languages that require speakers to mark tense and Aktionsart (potentially also aspect).

and a common noun. The second reason is that readily made structure dramatically reduces the search space. If a structure like the one final structure in Figure 8 is constructed from scratch using only simple operations, the search tree would have a search depth of eight (essentially one step in depth per operation). However, every time an operation is added to a program, it can be linked to the current structure in multiple ways, which leads to an explosion of nodes on every layer of depth. Hence, the system soon has to deal with a wide search tree, where every node will be executed and tested against the context. Consequently, using chunking dramatically increases the performance of the system, even in simple examples.

Chunking is possible in IRL because of two concepts touched upon earlier: the data flow representation of programs and the multidirectionality of operations (see Section 3). These two tightly intertwined concepts form the basis of chunking. In parsing, semantic structure such as in Figure 3 can be completely computed by the grammar engine, including all semantic entities, via bind statements and so forth. Hence, when executing this structure against the context most of these filtering operations will be passed input sets, as well as prototypes, object classes and the like. Their job in this case is to apply these categories and object classes onto the input set. On the other hand, in conceptualization the information about the particular discriminating classes and prototypes actually needs to be computed.

The chunk here only specifies the operations and leaves the concrete adjective, e.g. red or blue, underspecified and the operations are typically computing output-set-prototype pairs in order to provide hypotheses about possibly sensible meaning with respect to the current communicative goal.

So what about open-ended adaptation then? In lexicon formation studies, open-ended semantics refers to the fact that the number of prototypes, categories and names are not a priori fixed in the inventories of the agent, but rather that agents self-organize their categorical segmentation of the sensorimotor space based on success in communication. This idea has led to convincing insights that have been readily incorporated by IRL and successfully applied in lexical development scenario. Ergo, the system can incorporate ideas present in the community on lexical development and adaptation and leverage results. But, the system allows for adaptation on a different level as well. With IRL agents are able to autonomously evolve semantic structure itself, i.e. the linking of operations. The basis for this kind of adaptation is set with the basic structure of chunks and their entangled use in conceptualization and interpretation. In conceptualization agents naturally build semantic structure, hence they can easily store parts of the conceptualized structure. The same holds for interpretation, where chunks can be not only used, but also constructed from interpretation search trees. Moreover, when building conceptualization and interpretation search trees, agents know which chunks were used in the creation of a particular network. Hence, chunks can be independently rated and scored based on their usage in networks and based on the success of such networks in communicative interactions. To that end, chunks feature a score, which reflects their success. Moreover, since structure is represented explicitly in IRL-networks, graph-based similarity measures can be used to guide analogy and generalization learning operators which further compress structure and extract significant parts. Such extracted and compressed networks can immediately become chunks in their own right.

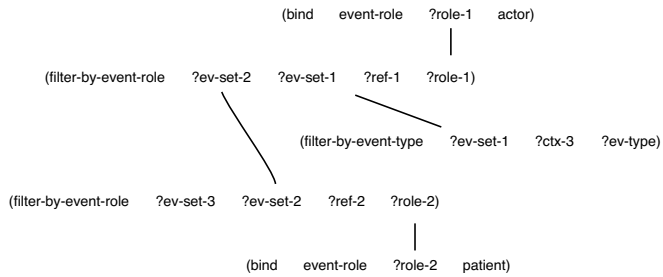

Figure 7. IRL-network of a chunk that is the meaning of a 'transitive construction', which is a grammatical rule that requires an event and two participants: actor and patient. Note, this basic distinction between the event participants is typically referred to as agent and patient or actor and undergoer. Here, however, we call it actor and patient to avoid confusion with the other use of the word agent in this paper. The operations in this network filter the context for events of a specific type as specified by the verb (`filter-by-event-type`), followed by filtering operations for a specific actor event participant, followed by the filter for a specific patient of that event (both represented by `filter-by-event-role` and particular bind statements). Figure 8 shows a concrete meaning constructed using this chunk, which can be verbalized by using a transitive construction.

6 DISCUSSION AND CONCLUSION

We have already touched upon different domains where the presented approach has proven useful. Studies have been conducted and are currently underway to model syntax and semantics of Russian verbs and event structure, as well as on the co-evolution of syntax and semantics in spatial language [19]. Moreover, the system is used to explore issues of grounding and computational modeling long missing from certain traditions in cognitive linguistics such as image schema theory. All these studies for the first time unite perception and language processing while grounding increasingly complex syntactic structure within an integrated system framework that is connected to the real world

There is another important observation related to the discussion in this paper. Ultimately the system was developed with a particular usage scenario in mind, namely to fill the gap between perception and language (see Figure 2) in the processing of situated interaction given certain communicative tasks. This is a rather general use case which in principle relates to a large array of applications from human-robot interaction to robot-robot interaction and computational models of the evolution of language. Hence, it should be noted that the system can be used to study ontogeny, i.e. developmental artificial systems [8], equally well as phylogeny, i.e. evolution of language [19]. Moreover, different levels of adaptation are conceivable, from systems that start with completely empty semantic inventories to handcrafted scenarios in which the grammar as well as semantics are provided by an engineer, for instance to develop human-robot interaction systems.

IRL is meant to be tightly integrated with a grammar engine. In our experiments we rely on *Fluid Construction Grammar* [6, 27] as a formalism for processing language, but other approaches can be used as well. Nevertheless, it should be mentioned that most ideas presented here are rooted in usage-based theories of language and necessitate a tight syntax-semantics interface. For instance, IRL-networks as hinted in the discussion of Figures 7 and 8, IRL-networks can have a quite direct mapping to syntax and desirably language is well-integrated with the representations constructed on the semantic side.

The oldest and in some sense most similar system to what we have presented here is Winograd's SHRDLU [35], which however misses the key aspects of grounding, active interpretation and conceptualization as a search process. Other work such as [1, 17] focuses mostly on lexical meaning. Some approaches have taken more general approaches e.g. to event structure [14] but stay mostly tied with that particular domain. One of the few approaches talking about objects and events in the same framework is [15], which is comparable to ours, but so far has been a theoretical proposal only.

This paper has presented recent progress on a computational semantics system that supports open-ended, grounded communication. We have shown how the system can be used to tackle different semantic domains, e.g. space, event structure, time and Aktionsarten. Furthermore, current and potential future applications of the system were explored.

Acknowledgments

We are greatly indebted to Masahiro Fujita, Hideki Shimomura, and their team at Sony Corporation for making the humanoid robots available for the experiments reported here and also to Luc Steels without whom none of the work presented here would have been possible. This research was funded by Sony CSL Paris with additional funding from the ALEAR project (EU FP7, grant 214856). We greatly thank Nancy Chang for proofreading and the anonymous reviewers for their comments and suggestions.

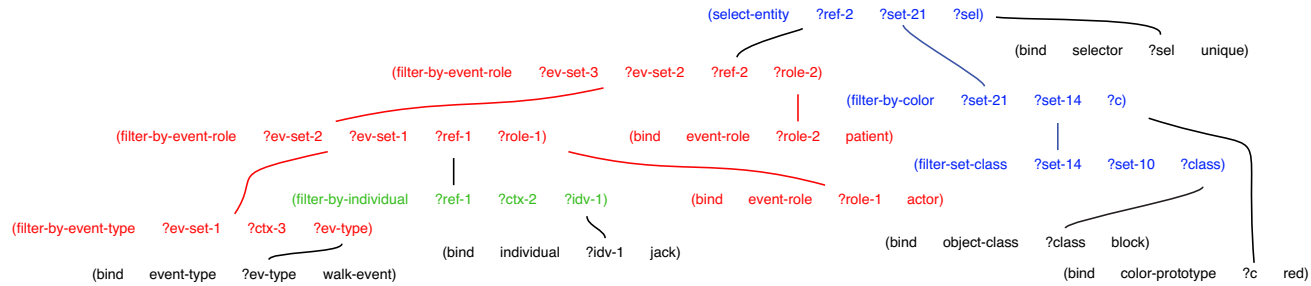

Figure 8. Semantic structure underlying the utterance *Jack pushes the red block*, which is constructed using the chunk in Figure 7 (highlighted in red), but also using chunks for what could be called a determined color adjective noun phrase (here marked in blue) and a common noun (marked in green). For clarity reasons `get-context` was omitted, but essentially variables starting with `?ctx-` are linked to that operation.

REFERENCES

[1] D. Bailey, J. Feldman, S. Narayanan, and G. Lako, 'Modeling embodied lexical development', in *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pp. 84–89, (1997).

[2] J-C. Baillie and J-G. Ganascia, 'Action categorization from video sequences', in *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*, ed., W. Horn, pp. 643–647. IOS Press, (2000).

[3] A. Billard and K. Dautenhahn, 'Grounding communication in autonomous robots: An experimental study', *Robotics and Autonomous Systems*, **24**(1), 71–79, (1998).

[4] J. Bleys, M. Loetzsch, M. Spranger, and L. Steels. The grounded colour naming game. Proceedings of Roman-09, 2009.

[5] L.A. Carlson and P.L. Hill, '7. Formulating Spatial Descriptions across Various Dialogue Contexts', *Spatial Language and Dialogue*, **1**(9), 89–104, (2009).

[6] J. De Beule and L. Steels, 'Hierarchy in fluid construction grammar', in *KI 2005: Advances In Artificial Intelligence. Proceedings of the 28th German Conference on AI*, ed., U. Furbach, volume 3698 of *LNAI*, pp. 1–15, Koblenz, (2005). Springer.

[7] M. Fujita, Y. Kuroki, T. Ishida, and T. T. Doi, 'Autonomous behavior control architecture of entertainment humanoid robot SDR-4X', in *Proceedings IROS '03*, pp. 960–967, vol. 1, Las Vegas, Nevada, (October 2003).

[8] K. Gerasymova and M. Spranger, 'Development of language competence in autonomous artificial systems', *Accepted at ECAI 2010*, (2010).

[9] K. Gerasymova, L. Steels, and R. van Trijp, 'Aspectual morphology of russian verbs in fluid construction grammar', in *Proceedings of the 31th Annual Conference of the Cognitive Science Society (Cogsci09)*, eds., N. A. Taatgen and H. van Rijn, pp. 1370–1375, Austin, TX, (2009). Cognitive Science Society.

[10] P. N. Johnson-Laird, 'Procedural semantics', *Cognition*, **5**(3), 189–214, (1977).

[11] J.R. Koza, *Genetic programming: on the programming of computers by means of natural selection*, The MIT press, 1992.

[12] S. C. Levinson, 'Language and space', *Annual Review of Anthropology*, **25**, 353–382, (1996).

[13] S. C. Levinson, *Space in language and cognition: Explorations in cognitive diversity*, Cambridge University Press, 2003.

[14] S. Narayanan, 'Moving right along: A computational model of metaphoric reasoning about events', in *Proc. Sixteenth National Conference of Artificial Intelligence (AAAI-99)*, (1999).

[15] D. Roy, 'Semiotic schemas: A framework for grounding language in action and perception', *Artificial Intelligence*, **167**(1-2), 170–205, (2005).

[16] J. M. Siskind, 'Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic', *Journal of Artificial Intelligence Research*, **15**, 31–90, (2001).

[17] J. M. Siskind, 'Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic', *Journal of Artificial Intelligence Research*, **15**(1), 31–90, (2001).

[18] M. Spranger and M. Loetzsch, 'The semantics of SIT, STAND, and LIE embodied in robots', in *Proceedings Cogsci 09*, eds., Niels A. Taatgen and Hedderik van Rijn, pp. 2546–2552, Austin, TX, (2009). Cognitive Science Society.

[19] M. Spranger, S. Pauw, and M. Loetzsch, 'Open-ended semantics co-evolving with spatial language', *The Evolution of Language. Proceedings of the 8th International Conference*, (2010).

[20] Michael Spranger, *World Models for Grounded Language Games*, German diplom (master) thesis, Humboldt-Universitaet zu Berlin, 2008.

[21] L. Steels, 'The emergence of grammar in communicating autonomous robotic agents', in *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*, ed., W. Horn, pp. 764–769, Berlin, Germany, (2000). IOS Press.

[22] L. Steels, 'Constructivist development of grounded construction grammars', in *Proceedings 42nd Annual Meeting of the Association for Computational Linguistics*, ed., W. Daelemans, pp. 9–19, (2004).

[23] L. Steels, 'The symbol grounding problem has been solved. so what's next?', in *Symbols and Embodiment: Debates on Meaning and Cognition*, ed., M. de Vega, Oxford University Press, Oxford, (2008).

[24] L. Steels and T. Belpaeme, 'Coordinating perceptually grounded categories through language: A case study for colour', *Behavioral and Brain Sciences*, **28**(4), 469–489, (August 2005).

[25] L. Steels and J. Bleys, 'Planning what to say: Second order semantics for fluid construction grammars', in *Proceedings CAEPIA '05*, Santiago di Compostella, (2005).

[26] L. Steels and J. Bleys, 'Planning what to say: Second order semantics for fluid construction grammars', in *Proceedings of CAEPIA 5*, (2005).

[27] L. Steels and J. De Beule, 'Unify and merge in fluid construction grammar', in *Proceedings EELC 06*, eds., P. Vogt, Y. Sugita, E. Tuci, and C. Nehaniv, volume 4211 of *Lecture Notes in Computer Science*, pp. 197–223. Springer Verlag, (2006).

[28] L. Steels and F. Kaplan, 'Bootstrapping grounded word semantics', in *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, ed., T. Briscoe, 53–73, Cambridge University Press, Cambridge, (2002).

[29] L. Steels and M. Loetzsch, 'Perspective alignment in spatial language', in *Spatial Language and Dialogue*, eds., K. R. Coventry, T. Tenbrink, and J. A. Bateman, 70–89, Oxford University Press, (2009).

[30] L. Steels and M. Spranger, 'The robot in the mirror', *Connection Science*, **20**(4), 337–358, (2008).

[31] L. Steels and P. Wellens, 'How grammar emerges to dampen combinatorial search in parsing', in *Symbol Grounding and Beyond: Proceedings of the 3rd EELC*, eds., P. Vogt, Y. Sugita, E. Tuci, and C. Nehaniv, LNAI 4211, pp. 76–88, Berlin, (2006). Springer-Verlag.

[32] W. Van Den Broeck, 'Constraint-based compositional semantics', in *Proceedings of the 7th International Conference on the Evolution of Language (EVOLANG 7)*, eds., Andrew D.M. Smith, Kenny Smith, and Ramon Ferrer i Cancho, pp. 338–345, Singapore, (2008). World Scientific Publishing.

[33] R. van Trijp, 'The emergence of semantic roles in fluid construction grammar', in *Proceedings of the 7th International Conference on the Evolution of Language (EVOLANG 7)*, eds., A. D.M. Smith, K. Smith, and R. Ferrer i Cancho, pp. 346–353, Singapore, (2008). World Scientific Publishing.

[34] P. Vogt, 'Bootstrapping grounded symbols by minimal autonomous robots', *Evolution of Communication*, **4**(1), 87–116, (2002).

[35] T. Winograd, *Procedures as a representation for data in a computer program for understanding natural language.*, Ph.D. dissertation, MIT, 1971.