Advances in Cognitive Systems X (20XX) 1–6 Submitted X/20XX; published X/20XX

XMRs: Uniform Semantic Representations for Intermodular Communication in Cognitive Robots

Jesse English ENGLIJ2@RPI.EDU

Sergei Nirenburg NIRENS@RPI.EDU

Department of Cognitive Science, Rensselaer Polytechnic Institute, Troy, NY 12180 USA

Abstract

We present the purpose, format and content of internal signals exchanged among the processing modules of the OntoAgent HRI-oriented architecture. These signals represent the agent's understanding of the situation at hand and motivate its potential response to it. Having been formulated in the same ontological framework, they support the use of interpreted input from multiple perception sources in the agent's decision-making and provide input for translating the agent's decisions into physical and verbal action.

1. Introduction.

This paper briefly introduces the purpose, format and content of a variety of internal signals exchanged among the processing modules of the OntoAgent HRI-oriented architecture. These signals represent the results of the agent's conscious reasoning operations and thus reflect the agent's understanding of the situation at hand and motivate its potential response to it. In other words, these signals convey meaning and are, therefore, collectively referred to as XMRs (MR stands for meaning representation). All the XMRs are interoperable because they are based on a uniform ontological substrate. XMRs can be divided into input- and output-oriented ones. In a robotic system that incorporates OntoAgent (see Figure 1), input XMRs are generated by the interpreters of the results of available external perception processing modules, while output XMRs provide input to the available effector modules. For example, the recent furniture building application, required only the perception modules for vision and language and the effector modules for verbal and physical actions. While all these input/output modules are operational at this time, they are under continuous development.

The interpreters of the percepts yield *visual MR*s (VMRs) and (input) *text MRs* (TMRs), respectively. On the output side, the decision-making modules must provide *action MRs* (AMRs) to the physical action system and (output) TMRs used by the language generator. Intermodular communication is also supported by *mental MRs* (MMRs) that encode mental actions and are used, among other purposes, to update the agent's situation model, to signal attention needs (e.g., the need to instantiate a goal instance), and to trigger any of the several truth maintenance and learning actions for the augmentation and improvement of the agent's memory models.

The format of inputs to the perception interpreters in Figure 1 is outside the scope of this paper, as is the format of outputs of the two action generators. In the current implementation of the robotic system that incorporates OntoAgent, the visual perception interpreter (VPI) that generates VMRs from the output of the dedicated visual perception interpreter and the physical action generator that translates AMRs into commands to a robotic action system, are developed to interface with a particular robotic system used in this application (Nirenburg et al. 2018). This type of work will have to be repeated for each environment in which OntoAgent must be integrated. This requirement does not hold for the natural language understander, which is tightly integrated into OntoAgent and uses the same decision-making infrastructure and the same set of knowledge resources as the core

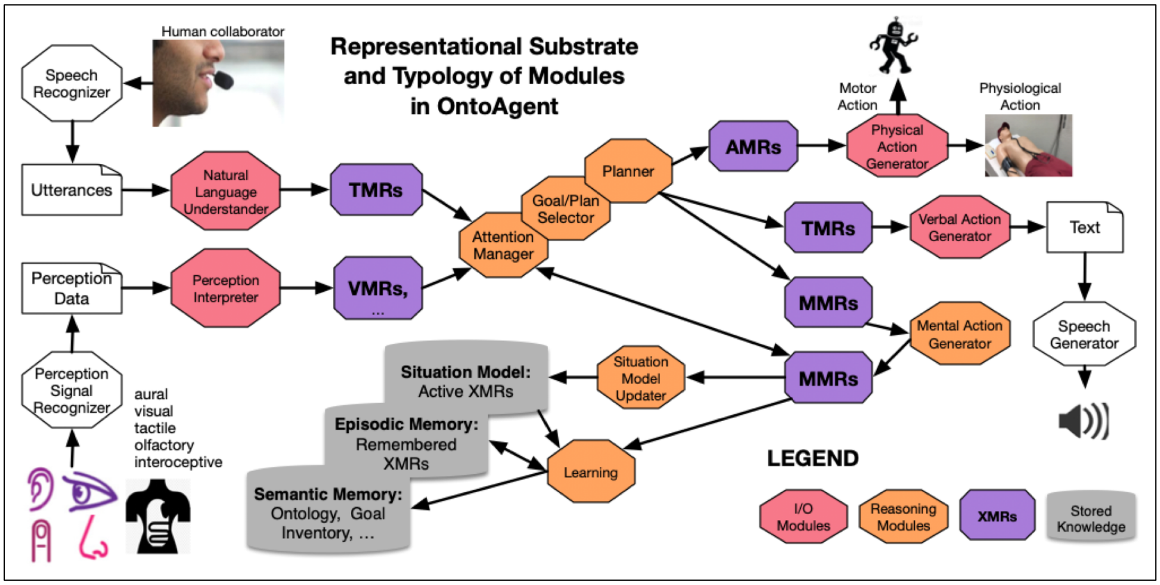

Figure 1. XMRs in OntoAgent. XMRs are interoperable meaning representations (MRs) that integrate the input/output modules with the system's internal processing modules and its static knowledge resources. We illustrate text MRs (TMRs) and visual MRs (VMRs) on the perception side. The system's planner generates action MRs (AMRs) as input for the external motor action module and TMRs as input for the verbal action generator module. The planner sends mental MRs (MMRs) to the mental action generation module to signal a need for replanning and/or instantiating (new) goals. The mental action generator can also trigger replanning independently, through metareasoning. This module also supports truth maintenance (updating the situation model) and learning (consolidation of episodic memory and learning of ontology). Specific applications might require only a subset of the input and output modalities. Thus, in the current furniture-building application the input involves only vision and language and the output, only language and physical action. Language processing and simulated interoception/physiological action modules are fully integrated with the agent, while vision and other perception modalities and motor action are imported and require custom interpreters to interact with OntoAgent. The perception interpreters are needed to translate output of the native perceptual systems into their corresponding XMRs; on the output side, AMRs must be translated into the representation native to a specific robotic motor action system.

decision-making modules of the OntoAgent. In fact, historically, XMRs are extensions of the TMRs that have been used in OntoSem, OntoAgent's natural language understander, for many years.

All of the agent's knowledge is represented in a uniform frame-based representation language. This knowledge covers all of the agent's internal knowledge resources: its long-term semantic memory (its ontology), its long-term episodic memory and its situation model (working memory) that comprises information about environment, the agent's goal/plan agenda, input and output XMRs, perspectives on other agents, and more. Properties (slots) in all of the frames, irrespective of the type of memory for which they are defined, are taken from a single repository in the agent's ontological world model. This ensures semantic cohesion, as these properties are reused across the types of XMRs. All individual frames are connected through metalevel relations, forming a single directed graph. The resulting graph database is flexible and facilitates an efficient query process. We use namespaces to provide soft division of frames in the graph: output XMRs are found in the @OUTPUT namespace, ontology is found in the @ONT namespace (which is itself effectively a subdivision of the @LTM, or long-term memory, namespace), the situation model is found in the @ENV namespace, and so forth. All XMRs also include an anchor section that contains metadata that is important for the agent's reasoning, such as timestamps (when the input was observed, or the output was generated), sources (e.g., the speaker for a TMR), and more. In the sample XMRs below, we omit the anchor section for brevity. The examples below are from the joint human-robot furniture building application.

In the sections that follow we present examples of a TMR, a VMR, an MMR and an AMR and then illustrate the input and output XMRs as they support human-agent interaction.

2. XMRs: TMRs, VMRs, MMRs and AMRs

TMRs: Text Meaning Representations. We illustrate an input TMR. The process whereby the planner generates an output TMR for the text (dialog turn) generation will be reported separately. Below is the TMR for the input *Get a foot bracket* issued by the user Jake whom the agent knows[1]:

```
@TMR.REQUEST-ACTION.1 = {          /speech act meaning is essential for reasoning and decision-making
   AGENT          @LTE.HUMAN.35;   /this is the concept instance in episodic memory corresponding to Jake
   BENEFICIARY    @TMR.ROBOT.0;    /this is a reference to self
   THEME          @TMR.TAKE.3; };  /the situation model already includes two instances of TAKE
@TMR.TAKE.3 = {
   AGENT          @LTE.ROBOT.0;
   THEME          @TMR.BRACKET.1; }; /this is the first time a bracket was mentioned in the situation
@TMR.BRACKET.1 = {
   SIDE-TB        BOTTOM; };       /this is an example of the use of a literal property filler from a predefined set
```

VMRs: Visual Meaning Representations. VMRs are produced by a visual perception interpreter (VPI). VMRs can represent two main kinds of information: a) timed spatial representations of the objects in a scene (that is, what can be seen, where and when) and b) visual events generated by reasoning over sequences of scene representations. Suppose the VPI generated the event of Jake affixing the foot and front brackets to a dowel. The VMR for this event would look as follows:

```
@VMR.ATTACH.1 = {
   AGENT          @LTE.HUMAN.35;  /Jake was recognized visually.
   THEME          @LTE.BRACKET.1;
   THEME          @LTE.BRACKET.2;
   DESTINATION    @VMR.DOWEL.1; };
@VMR.LOCATION.1 = {     /only results of visual perception are included; the system knows much more about Jake!
   DOMAIN         @LTE.HUMAN.35;
   RANGE          @VMR.ABSOLUTE-LOCATION.1;
   IN-FRONT-OF    @ENV.TABLE.1; };
@VMR.ABSOLUTE-LOCATION.1 = {…}; /quantitative data location data, with respect to a spatial origin point
```

[1] Grounding instances of concepts in the agent's situation model and long-term episodic memory is an important task in input interpretation. It is not discussed in this paper. We will report on our work on grounding separately.

The input to the VPI is expected to be a set of quantitative (Kennedy et al. 2007), egocentric or allocentric (Klatzky 1998) streams of object data. It is the job of the VPI to understand the input in the context of previously received (and analyzed) visual data, to produce the VMR. We expect this input to include features such as location, heading, axis, and bearing - each with respect to a defined spatial origin - these would be represented in the @VMR.ABSOLUTE-LOCATION.1 frame above. The VPI will generate agent-oriented relational locations, such as the filler for the property IN-FRONT-OF.

MMRs: Mental Meaning Representations. MMRs are generated by the agent to support any number of its reasoning tasks, such as learning or prioritization of agenda. MMRs are considered output XMRs, as they are results of the agent's reasoning. MMR is used to affect a change, albeit one that is internal to the agent's memory. Thus, the MMR reflecting the decision the agent makes to start building a chair (made, for example, as a result of processing a human team member's utterance *It's time to start working* or one of its paraphrases) will take the form:

```
@MMR.ADD-GOAL-INSTANCE.1 = {
   AGENT        @SELF.ROBOT.0;
   THEME        @ONT.HAVE-BUILT-A-CHAIR; };
```

Here the MMR is specifying that a new instance of a goal should be added to the agent's agenda. The goal type (non-instanced) is specified. Once the MMR is executed, the agent's agenda will be updated with a new instance of the goal.

AMRs: Action Meaning Representations. AMRs are signals to the agent's physical effector systems to carry out an action in the world. AMRs represent the meaning of this action, not the fine-grained motor control information. The AMR conveying the meaning "I will move the screwdriver to the table" looks as follows:

```
@AMR.CHANGE-LOCATION.1 = {
   AGENT           @SELF.ROBOT.0;
   THEME           @ENV.SCREWDRIVER.13;
   DESTINATION     @ENV.TABLE.2; };
```

The AMR will need to be post-processed into a series of robot-specific motor control commands. The same holds true for output TMRs that are used as inputs to the verbal action generator module of the system.

3. An Example of Human-Agent Interaction

The example we present shows the XMRs involved in running a portion of a script in our furniture assembly domain. The agent receives both TMR and VMR as inputs, and generates MMRs, AMRs and TMRs as necessary. Most of the XMRs reference entities in the agent's long-term memory, which emphasizes the integration of all of the types of agent knowledge to support reasoning. In the example scenario, the agent has previously learned (through instruction in natural language, see Nirenburg and Wood 2017) how to build a chair. Furthermore, all components required for that task are readily available in the environment. The scenario steps are presented below[2]; the XMRs generated while processing it are illustrated in Figure 2.

[2] Space constraints necessitate many simplifications in this presentation. Thus, we do not discuss the mechanism for prioritizing the goal/plan agenda, or the particulars of the planning mechanism, or the nature of the various heuristic decision functions, or treatment of impasses, or how parallelism is supported by tracking availability of perceptors and effectors, or whether XMR signals are transmitted across modules individually or in batches, etc.

1. Jake (@LTE.HUMAN.35): <Enters the room>.
2. Robot (@SELF.ROBOT.0): <Perceives that Jake is in the room>
3. Robot: <Modifies its agenda: instantiates the goal HAVE-GREETED-HUMAN>
4. Robot: <prioritizes the above goal, selects a plan for it and carries it out: generates an output TMR representing a greeting>
5. Jake: Issues the utterance: *Let's build a chair*.
6. Robot: <Understands the utterance>
7. Robot: <Modifies its agenda: instantiates the goal HAVE-BUILT-A-CHAIR>
8. Robot: <prioritizes the above goal, selects a plan to attain the above goal and generates an AMR representing the first action in the plan: to pick up a dowel>

/VMR output at Step 2. Generated by VPI

```
@VMR.ENTER.1 = {
   AGENT         @LTE.HUMAN.35;
   THEME         @ENV.ROOM.1; };
```

/MMR output at Step 2. Generated by Attention Manager and sent to Situation Model Updater

```
@MMR.REMEMBER.1 = {
   AGENT         @SELF.ROBOT.0;
   THEME         @MMR.LOCATION.1; };

@MMR.LOCATION.1 = {
   DOMAIN        @LTE.HUMAN.35;
   RANGE         @ENV.ROOM.1; };
```

/MMR output at Step 3. Generated by Attention Manager

```
@MMR.ADD-GOAL-INSTANCE.1 = {
   AGENT         @SELF.ROBOT.0;
   THEME         @ONT.HAVE-GREETED-
HUMAN; };
```

/TMR output at Step 4. Generated by Planner

```
@TMR.GREET.1 = {
   AGENT         @SELF.ROBOT.0;
   THEME         @LTE.HUMAN.35; };
```

/TMR output at Step 6. Generated by Natural Language Understander

```
@TMR.REQUEST-ACTION.1 = {
   AGENT         @LTE.HUMAN.35;
   BENEFICIARY   @SELF.ROBOT.0;
   THEME         @TMR.BUILD.1; };
@TMR.BUILD.1 = {
   AGENT         @ENV.HUMAN.1;
   AGENT         @SELF.ROBOT.0;
   THEME         @TMR.CHAIR.1; };
@TMR.CHAIR.1 = {};
```

/MMR output at Step 7. Generated by Attention Manager

```
@MMR.ADD-GOAL-INSTANCE.1 = {
   AGENT         @SELF.ROBOT.0;
   THEME         @ONT.HAVE-BUILT-A-CHAIR;
};
```

/AMR output at Step 8. Generated by Planner

```
@AMR.CHANGE-LOCATION.1 = {
   AGENT         @SELF.ROBOT.0;
   THEME         @ENV.DOWEL.1;
   DESTINATION   @ENV.TABLE.1; };
```

Figure 2. XMR signals generated by OntoAgent as integrated into the application in the domain of furniture assembly by a human-robotic team.

The robot observes Jake enter the room (Step 1) and passes the signal to the VPI that generates the corresponding VMR (Step 2). This causes Attention Manager to update the situation model by issuing the MMR signal (consisting in this case of two frames) and instantiate a new goal on the agent's goal/plan agenda (Step 3). When the agenda is prioritized at the next cycle, the associated plan to greet the human is selected, and a TMR is generated and passed to the Verbal Action

Generator (Step 4). Next, the human makes a request of the robot to begin a task (Step 5). An input TMR is produced by the language understander, (Step 6), which causes the agent's attention manager to update the agent's situation model and instantiate a new goal to build a chair on the agent's goal/plan agenda (Step 7). When the agenda is prioritized at the next cycle, one of the agent's stored plans for building a chair is selected, and an AMR is generated for the first action in the plan: to pick up a dowel (Step 8).

4. Conclusion

Our previous work in the area of interpreting "raw" perception signals into signals carrying meaning and thus facilitating the agent's reasoning, concentrated on the analysis of natural language. (We have also worked on interpreting interoception signals from a simulated model of human physiology, but that task did not involve adaptation to different world views, different semantics of knowledge elements or different formalisms: the simulation was developed using the same knowledge infrastructure as OntoAgent.) Over the past several years we have expanded our attention to interpreting "raw" vision percepts. Conversely, while in the past we addressed only verbal action (through our work in text generation), we have expanded into generation of instructions for a robotic action system. The XMR infrastructure supports this extended scope of work. Our work on natural language has amply demonstrated that language understanding sufficient for supporting human-like behavior in robots is only possible when language analysis has access to the entire space of knowledge resources in an artificial intelligent agent system. The amount of work involved in this task is significant. One of the issues we would like to investigate in incorporating other perception interpreters is the extent of the agent's stored knowledge that is necessary for interpreting "raw" perception results to the level when they become actionable.

Acknowledgement. This research was supported in part by grant #N00014-16-1-2218 from the U.S. Office of Naval Research. Any opinions or findings expressed in this material are those of the author and do not necessarily reflect the views of the Office of Naval Research.

References

Nirenburg, S. and P. Wood. 2017. Toward human-style learning in robots. AAAI Fall Symposium on Natural Communication with Robots.

Nirenburg, S., M. McShane, P. Wood, B. Scassellati, O. Mangin and A. Roncone. 2018. Robots learning through language communication with users. Proccedings of NLDB-18. Paris. June.

Kennedy, W., M. Bugajska, M. Marge, W. Adams, B. Fransen, D. Perzanowski, A. Schultz and J. G. Trafton. (2007). Spatial representation and reasoning for human-robot collaboration. Proceedings of the National Conference on Artificial Intelligence. 2. 1554-1559.

Klatzky, R. L. 1998. Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections. In C. Freksa, C. Habel, & K. F. Wender (Eds.), Spatial cognition - An interdisciplinary approach to representation and processing of spatial knowledge (Lecture Notes in Artificial Intelligence 1404) (pp. 1-17). Berlin: Springer-Verlag.