

Two diverse systems built using generic components for spoken dialogue (Recent Progress on TRIPS)

**James Allen, George Ferguson, Mary Swift, Amanda Stent, Scott Stoness,
Lucian Galescu, Nathan Chambers, Ellen Campana, and Gregory Aist**

University of Rochester
Computer Science Department
UR Comp Sci RC 270226
Rochester NY 14627 USA
{james, ferguson, swift, stoness,
campana, gaist}
@cs.rochester.edu

Institute for
Human and Machine Cognition
40 South Alcaniz St.
Pensacola FL 32502
{lgalescu,nchambers}@ihmc.us

State University of New York at
Stony Brook
1418 Computer Science
Stony Brook University
Stony Brook NY 11794 USA
stent@cs.sunysb.edu

Abstract

This paper describes recent progress on the TRIPS architecture for developing spoken-language dialogue systems. The interactive poster session will include demonstrations of two systems built using TRIPS: a computer purchasing assistant, and an object placement (and manipulation) task.

1 Introduction

Building a robust spoken dialogue system for a new task currently requires considerable effort, including extensive data collection, grammar development, and building a dialogue manager that drives the system using its "back-end" application (e.g. database query, planning and scheduling). We describe progress in an effort to build a generic dialogue system that can be rapidly customized to a wide range of different types of applications, primarily by defining a domain-specific task model and the interfaces to the back-end systems. This is achieved by using generic components (i.e., ones that apply in any practical domain) for all stages of understanding and developing techniques for rapidly customizing the generic components to new domains (e.g. Aist, Allen, and Galescu 2004). To achieve this goal we have made several innovations, including (1) developing domain independent models of semantic and contextual interpretation, (2) developing generic dialogue management components based on an abstract model of collaborative problem solving, and (3) extensively using an ontol-

ogy-mapping system that connects the domain independent representations to the representations/query languages used by the back-end applications, and which is used to automatically optimize the performance of the system in the specific domain.

2 Theoretical Underpinnings: The Problem-Solving Model of Dialogue

While many have observed that communication is a specialized form of joint action that happens to involve language and that dialogue can be viewed as collaborative problem solving, very few implemented systems have been explicitly based on these ideas. Theories of speech act interpretation as intention recognition have been developed (including extensive prior work in TRIPS' predecessor, the TRAINS project), but have been generally considered impractical for actual systems. Planning models have been more successful on the generation side, and some systems have used the notion of executing explicit task models to track and drive the interactions (e.g., Sidner and Rich's COLLAGEN framework). But collaborative problem solving, and dialogue in general, is much more general than executing tasks. In our applications, in addition to executing tasks, we see dialogue that is used to define the task (i.e., collaborative planning), evaluate the task (e.g., estimating how long it will take, comparing options, or likely effects), debug a task (e.g., identifying and discussing problems and how to remedy them), learn new tasks (e.g., by demonstration and instruction).

In the remainder of the paper, we'll first discuss the methods we've developed for building dialogue systems using generic components. We'll then describe two systems implemented using the TRIPS architecture that we will demonstrate at the interactive poster session.

3 Generic Methods: Ontology Mappings and Collaborative Problem Solving

The goal of our work is to develop generic spoken dialogue technology that can be rapidly customized to new applications, tasks and domains. To do this, we have developed generic domain independent representations not only of sentence meaning but also of the collaborative actions that are performed by the speech acts as one engages in dialogue. Furthermore, we need to be able to easily connect these generic representations to a wide range of different domain specific task models and applications, ranging from data base query systems to state-of-the-art planning and scheduling systems. This paper describes the approach we have developed in the TRIPS system. TRIPS is now being used in a wide range of diverse applications, from interactive planning (e.g., developing evacuation plans), advice giving (e.g., a medication advisor (Ferguson et al. 2002)), controlling teams of robots, collaborative assistance (e.g., an assistant that can help you purchase a computer, as described in this paper), supporting human learning, and most recently having the computer learn (or be taught) tasks, such as learning to perform tasks on the web. Even though the tasks and domains differ dramatically, these applications use the same set of core understanding components.

The key to supporting such a range of tasks and applications is the use of a general ontology-mapping system. This allows the developer to express a set of mapping rules that translate the generic knowledge representation into the specific representations used by the back-end applications (called the KR representation). In order to support generic discourse processing, we represent these mappings as a chain of simpler transformations. These representations are thus transformed in several stages. The first, using the ontology mapping rules, maps the LF representation into an intermediary representation (AKRL - the abstract KR language) that has a generic syntax but whose content is expressed in terms of the KR ontology. The second stage is a

syntactic transformation that occurs at the time that calls to the back-end applications actually occur so that interactions occur in the representations the back-end expects. In addition to using ontology mapping to deal with the representational issues, TRIPS is unique in that it uses a generic model of collaborative problem solving to drive the dialogue itself (e.g. Allen, Blaylock, and Ferguson 2002). This model forms the basis of a generic component (the collaboration manager) that supports both intention recognition to identify the intended speech acts and their content, planning the system's actions to respond to the user (or that take initiative), and providing utterance realization goals to the generation system. To develop this, we have been developing a generic ontology of collaborative problem solving acts, which provide the framework for managing the dialogue. The collaboration manager queries a domain-specific task component in order to make decisions about interpretations and responses.

4 TRIPS Spoken Dialogue Interface to the CALO Purchasing Assistant

The CALO project is a large multisite effort which aims at building a computerized assistant that learns how to help you with day-to-day tasks. The overarching goal of the CALO project is to

... create cognitive software systems, that is, systems that can reason, learn from experience, be told what to do, explain what they are doing, reflect on their experience, and respond robustly to surprise (Mark and Perreault 2004).

Within this broad mandate, one of our current areas of focus is user-system dialogue regarding the task of purchasing - including eliciting user needs, describing possibilities, and reviewing & finalizing a purchase decision. (Not necessarily as discrete stages; these elements may be interleaved as appropriate for the specific item(s) and setting.) Within the purchasing domain, we began with computer purchasing and have branched out to other equipment such as projectors.

How to help with purchasing? The family of tasks involving purchasing items online, regardless of the type of item, have a number of elements in common. The process of purchasing has some common

dialogue elements - reporting on the range of features available, allowing the user to specify constraints, and so forth. Also, regarding the goal that must be reached at the end of the task, the eventual item must:

Meet requirements. The item needs to meet some sort of user expectations. This could be as arbitrary as a specific part number, or as compositional - and amenable to machine understanding - as a set of physical dimensions (length, width, height, mass, etc.)

Be approved. Either the system will have the authority to approve it (cf. Amazon's one-click ordering system), or more commonly the user will review and confirm the purchase. In an office environment the approval process may extend to include review by a supervisor, such as might happen with an item costing over (say) \$1000.

Be available. (At one time a certain electronics store in California had the habit of leaving out floor models of laptops beyond the point where any were

actually available for sale. (Perhaps to entice the unwitting customer into an “upsale”, that is, buying a similar but more expensive computer.)) On a more serious note, computer specifications change rapidly, and so access to online information about available computers (provided by other research within CALO) would be important in order to ensure that the user can actually order the machine he or she has indicated a preference for.

At the interactive poster session, we will demonstrate some of the current spoken dialogue capability related to the CALO task of purchasing equipment. We will demonstrate a number of the aspects of the system such as initiating a conversation, discussing specific requirements, presenting possible equipment to purchase, system-initiated reminders to ask for supervisor approval for large purchases, and finalizing a decision to purchase.

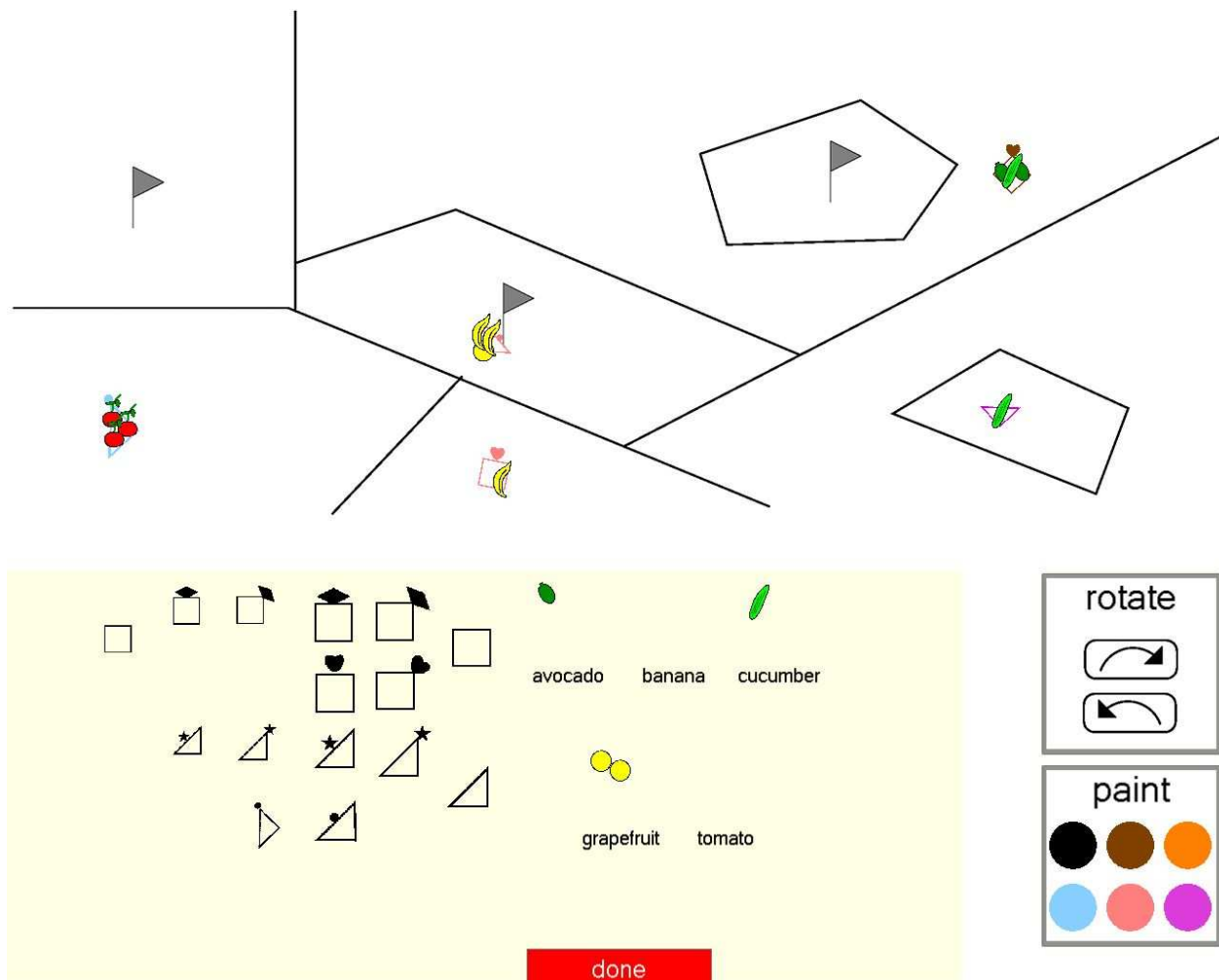


Figure 1. Fruit carts display.

5 TRIPS Spoken Dialogue Interface to choosing, placing, painting, rotating, and filling (virtual) fruit carts

TRIPS is versatile in its applications, as we've said previously. We hope to also demonstrate an interface to a system for using spoken commands to modifying, manipulating, and placing objects on a computer-displayed map. This system (aka "fruit carts") extends the TRIPS architecture into the realm of continuous understanding. That is, when state-of-the-art dialogue systems listen, they typically wait for the end of the utterance before deciding what to do. People on the other hand do not wait in this way – they can act on partial information as it becomes available. A classic example comes from M. Tanenhaus and colleagues at Rochester: when presented with several objects of various colors and told to "click on the yel-", people will already tend to be looking relatively more at the yellow object(s) even before the word "yellow" has been completed. To achieve this type of interactivity with a dialogue system – at least at the level of two or three words at a time, if not parts of words – imposes some interesting challenges. For example:

1. Information must flow asynchronously between dialogue components, so that actions can be triggered based on partial utterances even while the understanding continues
2. There must be reasonable representations of incomplete information – not just "incomplete sentence", but specifying what is present already and perhaps what may potentially follow
3. Speech recognition, utterance segmentation, parsing, interpretation, discourse reasoning, and actions must all be able to happen in real time

The fruit carts system consists of two main components: first, a graphical interface implemented on Windows 2000 using the .NET framework, and connected to a high-quality eyetracker; second, a TRIPS-driven spoken dialogue interface implemented primarily in LISP. The actions in this domain are as follows:

1. Select an object ("take the large plain square")
2. Move it ("move it to central park")
3. Rotate it ("and then turn it left a bit – that's good")
4. Paint it ("and that one needs to be purple")
5. Fill it ("and there's a grapefruit inside it")

Figure 1 shows an example screenshot from the fruit carts visual display. The natural language in-

teraction is designed to handle various ways of speaking, including conventional definite descriptions ("move the large square to central park") and more interactive language such as ("up towards the flag pole – right a bit – more – um- stop there.")

6 Conclusion

In this brief paper, we have described some of the recent progress on the TRIPS platform. In particular we have focused on two systems developed in TRIPS: a spoken dialogue interface to a mixed-initiative purchasing assistant, and a spoken interface for exploring continuous understanding in an object-placement task. In both cases the systems make use of reusable components – for input and output such as parsing and speech synthesis, and also for dialogue functionality such as mapping between language, abstract semantics, and specific representations for each domain.

References

- Aist, G. 2004. Speech, gaze, and mouse data from choosing, placing, painting, rotating, and filling (virtual) vending carts. International Committee for Co-ordination and Standardisation of Speech Databases (COCOSDA) 2004 Workshop, Jeju Island, Korea, October 4, 2004.
- Aist, G.S., Allen, J., and Galescu, L. 2004. Expanding the linguistic coverage of a spoken dialogue system by mining human-human dialogue for new sentences with familiar meanings. Member Abstract, 26th Annual Meeting of the Cognitive Science Society, Chicago, August 5-7, 2004.
- James Allen, Nate Blaylock, and George Ferguson. A problem-solving model for collaborative agents. In *First International Joint Conference on Autonomous Agents and Multiagent Systems*, Bologna, Italy, July 15-19 2002.
- George Ferguson, James F. Allen, Nate J. Blaylock, Donna K. Byron, Nate W. Chambers, Myrsolava O. Dzikovska, Lucian Galescu, Xipeng Shen, Robert S. Swier, and Mary D. Swift. *The Medication Advisor Project: Preliminary Report*, Technical Report 776, Computer Science Dept., University of Rochester, May 2002.
- Mark, B., and Perrault, R. (principal investigators). 2004. Website for Cognitive Assistant that Learns and Organizes. <http://www.ai.sri.com/project/CALO>