

Using Discourse to Modify Control Procedures

Pete Bonasso

Texas Robotics & Automation Laboratories (TRACLabs)
1012 Hercules, Houston, TX
r.p.bonasso@jsc.nasa.gov

Abstract

We outline recent discourse work using a shared plans approach to modify control procedures for advanced life support systems. This effort was motivated during a recent water recovery test at NASA by a need to temporarily suspend automatic shutdown responses in the 3T control system while the crew conducted manual procedures. Discourse plans are reactive procedures executed in the RAPs reactive planning system. The parser generates descriptions of concepts in the memory of the control system. Discourse management runs concurrently with the control system and can handle a variety of interruptions and non-sequiturs.

Introduction

For a number of years we have been developing artificial intelligence control systems for advanced life support (Schreckenghost et al 1998, Bonasso 2001). These systems, built with the 3T control system (Bonasso et al 1997), were for the most part autonomous. Based on an increasing need to accommodate variable autonomy, however, we augmented our control system with a discourse management component called the Dynamic Predictive Memory Architecture (DPMA) (Fitzgerald & Wiseman 1997). DPMA parses an utterance not lexically but conceptually based on phrasal patterns in the control system memory in order to recognize relevant knowledge structures and to use those structures to identify action and sensing requests. With DPMA, discourse plans can be constructed using the procedure language of the control system, the RAPs system (Firby 1999).

Using DPMA in a recent water recovery environment we have been able to provide users with a more flexible interface to the control system, one that uses interactive discourse plans to guide the user into the concept space of the control system (Bonasso 2000).

The advanced water recovery system (WRS) (see Figure 1) developed at Johnson Space Center (JSC) is comprised of 1) a biological water processor (BWP) to remove organic compounds and ammonia, 2) a reverse osmosis (RO) system to remove inorganic compounds from the effluent of the BWP, 3) an air evaporation system (AES) to

recover additional water from the brine produced by the RO, and 3) a post processing system (PPS) to bring the water to within potable limits. From January 2001 through April 2002, the 3T system controlled the WRS autonomously in a continuous 24/7 integrated test.

Toward the end of that test, it became necessary for life support engineers to perform certain manual maintenance tasks that would inadvertently trigger automatic shutdown (ASD) procedures in the autonomous control system. Thin tubes in the nitrifier assembly would periodically become clogged with biomass and require the engineers to manually purge them. During the manual purge, the pressures in the tubes would drop below the minimum pressure being maintained by 3T, causing an automatic shutdown of the nitrifier water and air pumps for the affected tubes. As a temporary solution, we commented out the action part of the ASD in the code, leaving only the warning output. After the maintenance procedure, we uncommented the code.

Anticipating similar requirements in the future, we developed a set of discourse plans to allow the engineers to interactively modify any procedure which might be affected by manual tasks.

Our Approach to Interactive Procedure Modification

We take as our theoretical model of discourse the Shared Plans (SP) approach (Grosz and Sidner 1990). Because we had at our disposal a reactive planning engine in the RAPs system, we were able to implement a modified version of the algorithm developed by Lochbaum (Lochbaum 1998) to manage SP discourses.

RAPs (reactive plans) that carry out the SP processing use the RAPs agenda to represent the discourse segments and their interrelationships and a pushdown stack to hold the objects and data associated with each segment. We use the term discourse management (DM) to include the stack and the set of RAPs used to manage the dialog.

The DPMA parser transforms a given user utterance into a description of a language concept in memory, e.g., an object or a speech act. When the DM enters into a dialog with the user, each DM query sets up a monitor task to wait for the class of response expected. For example, the DM question "Which temperature sensor are you referring to?" sets up a monitor expecting the response to be a description

of an instance of the class of temperature sensor. The collection of event monitors and their associated SP processing RAPs comprises the DM's expectations of the user's next utterance. When a user response is detected,

one of the event monitors will trigger, allowing the continued development of the SP. If no event triggers, the utterance represents the user's desire to do something new and processing of a new SP is begun.

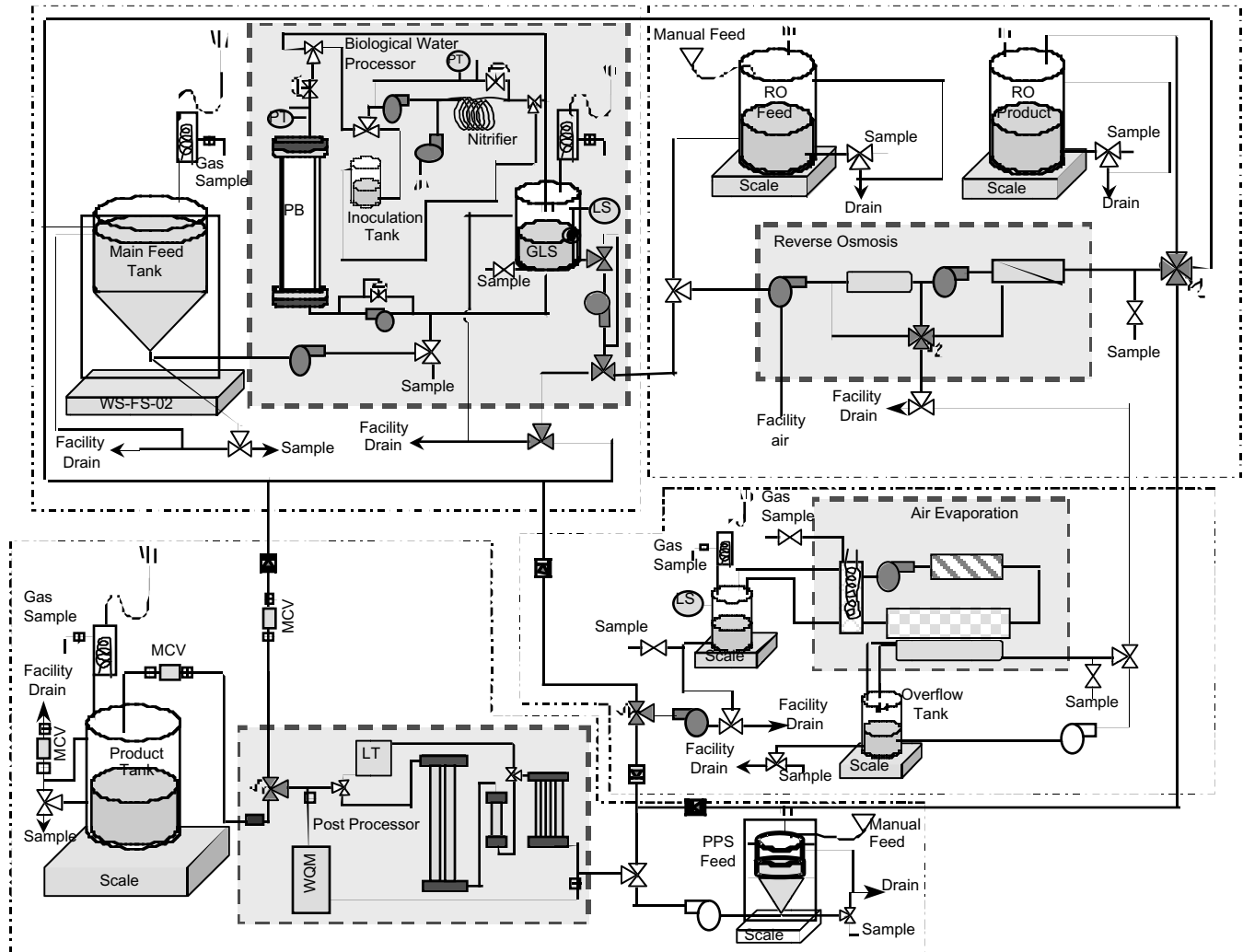


Figure 1 The Advanced Water Recovery System (WRS)

Until there is enough information for the discourse manager to invoke a RAP (recipe) to carry out the speech act, the SPs are only partially shared plans (PSPs). For example, the user may have asked the value of a temperature sensor in a heater, but there is more than one temperature sensor and the discourse manager will need clarification. The SP processing terminates for a PSP when a recipe to carry out the desired action -- also known as an individual plan, or IP -- can be successfully invoked. The SP becomes fully shared (FSP) at this point since the control system understands and concurs with the user's intentions and needs no further information to carry out the

task. We call IPs action RAPs. Action RAPs already exist for the control system, e.g., "Turn on the feed pump.", or can be developed for adjusting the control system, as in the case of procedure modification.

Finally, the SP RAPs usually make a last query to the user to insure that the action taken has indeed had the desired effect and the FSP is terminated. If the desired effect was not obtained, a new PSP is established for the same speech act and SP processing continues.

Flow of Control

Figure 2 shows the general flow of control for SP processing. "Service" is a general reference to an act that can be taken by the control system without further SP dialog. A user utterance either via text or point-and-click

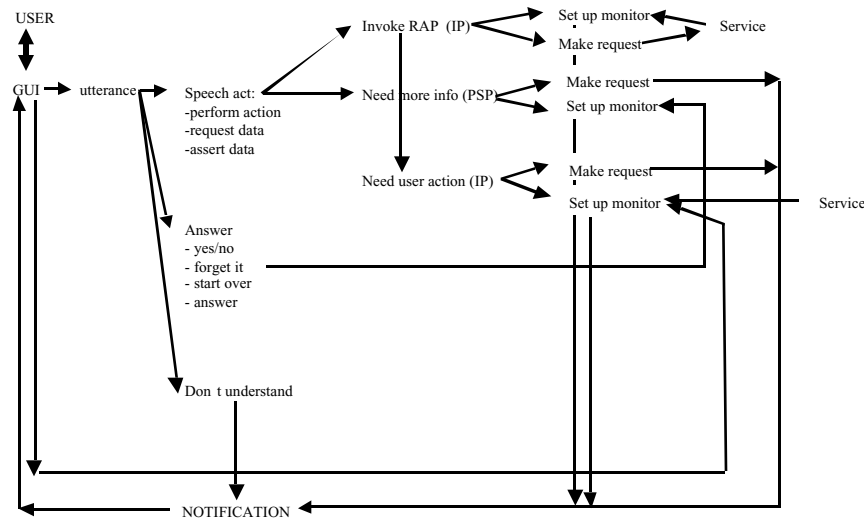


Figure 2: Shared Plans Processing

But the user can also change his mind about doing the task as in "forget it" or "never mind", as well as start the SP processing over again when he sees that the discourse direction is not to his liking. The user can also start up a new SP at anytime, since the RAP engine is designed to manage multiple reactive plans simultaneously (e.g., the DM runs concurrently with RAPs that control the physical system).

Speech acts can generate the invocation of an action RAP directly if the utterance and the state of affairs as understood by the DM contain all the information needed to carry out the action RAP. In these cases, the DM puts the action RAP on the agenda and a monitor is set up to watch for the desired effect. Additionally, the action RAP may require the user to execute a step in the recipe (a user IP), such as turning on a piece of equipment or entering an authorization code in a pop-up window. In these cases the user acts as a control agent and the action RAP sets up monitors to watch for the result of the user action.

If there is no recipe for the action, the SP fails and the user is notified accordingly. If the action RAP cannot be invoked because of missing information or the need for clarification, the user is asked for the missing information or clarification and a monitor is invoked to wait for the user's response.

Finally, the parser may fail to produce a description from the user's utterance. As mentioned earlier the issue here is one of guiding the user into a corpus of discourse circumscribed by the knowledge bases of the DM, which in turn is dependent on the set of RAPs and memory concepts

mode can be a speech act to perform an action, a request for data, an assertion of data or a response to a DM query for more information. Affirmative and negative answers are common to a number of queries such as "Do you mean the downstream transducer?" as well as "Is this the information you requested?"

for the given application. Our system searches the utterance for any subset of words that match a memory concept and responds with a guess at what the user required. Failing this, our system displays a list of suggested utterances (see Bonasso 2000).

An Example Procedure Modification

In this section, we provide an annotated transcript of a dialog taken from a session in which the user wishes to prevent the shut down of a pump when the pressure is too low. Procedure modification involves finding the task step in a relevant RAP and modifying it according to the user's desires. In this session, the user 1) starts over in the middle of the modification process, 2) asks an unrelated question, 3) decides the final deletion isn't to her liking and 4) eventually abandons the task altogether.

12/04 14:16:46 - I want to modify a procedure
 12/04 14:16:51 - USI: There is more than one procedure that I could modify. Which are you referring to?
 [Note : lines with the USI : tag are from the DM].

The user believes a recipe for modifying a procedure exists in the RAP library because in our implementation there is a GUI display of suggested utterances the user might employ while interacting with the system, one of which is "I want to modify a procedure." In order to clarify the general reference to "a procedure", the DM prompts the user and

sets up a monitor to watch for an utterance describing an instance of a procedure.

12/04 14:17:09 - How many BWP procedures are there?

12/04 14:17:31 - USI: There are 19 procedures. I will list them. [The DM lists the procedures one of which is named by the string "handle bioreactor alerts".]

12/04 14:17:37 - USI: There is more than one procedure that I could modify. Which are you referring to?

12/04 14:18:11 - handle bioreactor alerts

The monitor to watch for an utterance describing an instance of a procedure is still active and the DM refreshes its request to the user. This monitor will eventually time out if the needed parameters cannot be obtained in a reasonable amount of time.

12/04 14:18:22 - USI: Here are the methods for handle bioreactor alerts :

NITRIFIER-PROBLEM-FIXED
NITRIFIER-PRESSURE-LOW-MARK
NITRIFIER-PRESSURE-LOW-STOP-PUMP
NITRIFIER-PRESSURE-HIGH
NORMAL

Which one do you want to modify?

At this point the DM is working on an SP to glean from the user the parameters necessary to carry out a modification on the identified procedure. Two RAPs are active, one to interactively prompt the user for the parameters, and another to invoke the modify RAP when all the parameters are obtained.

12/04 14:18:48 - nitrifier-pressure-low-stop-pump

12/04 14:18:49 - USI: I understand you wish to modify the NITRIFIER-PRESSURE-LOW-STOP-PUMP method. Here are the tasks for nitrifier-pressure-low-stop-pump method.

task-net
sequence
Step t1 is NITRIFIER-TUBE-FLOW STOP
Step t2 is PUT-OUT-PBBWP-WARNING

Do you want to add to, delete or change the NITRIFIER-PRESSURE-LOW-STOP-PUMP method?

12/04 14:22:19 - start over

12/04 14:22:19 - USI: Okay, I'll start over.

The DM deletes the parameters for the SP obtained thus far and begins again.

12/04 14:22:23 - USI: Here are the methods for handle bioreactor alerts <previous listing omitted>. Which one do you want to modify?

12/04 14:22:51 - How many AES temperature sensors are there?

12/04 14:22:57 - USI: There are 24 A E S system temperatures. I will list them. <temperature sensor list omitted>.

12/04 14:23:02 - USI: Here are the methods for handle bioreactor alerts <previous listing omitted>

12/04 14:23:05 - USI: Which one do you want to modify?

The class of user response at 14: 22: 51 did not match any current expectation of the DM. Thus a new SP (an FSP, since the DM understood the intent of the speech act and knows a recipe for carrying out that act) is begun to answer the user's new query. Once that SP is completed, the original SP processing continues. The RAPs planning engine handles all of the refresh/interrupt details for interleaving reactive plans.

12/04 14:23:14 - nitrifier-pressure-low-stop-pump

12/04 14:23:16 - USI: I understand you wish to modify the nitrifier-pressure-low-stop-pump method. Here are the tasks for nitrifier-pressure-low-stop-pump method.

<previous listing omitted>

12/04 14:23:26 - USI: Do you want to add to, delete or change the nitrifier-pressure-low-stop-pump method?

12/04 14:23:35 - delete

12/04 14:23:47 - USI: Which task do you want to delete?

12/04 14:23:54 - t1

12/04 14:23:55 - USI: I understand you wish to delete the t1 task.

12/04 14:23:58 - USI: Here are the new tasks for nitrifier-pressure-low-stop-pump method.

task-net
sequence
Step t1 is NO-OP
Step t2 is PUT-OUT-PBBWP-WARNING

When the task was identified, the DM had all the information needed to carry out the procedure modification.

12/04 14:24:04 - USI: Are you satisfied with the delete modification to the nitrifier-pressure-low-stop-pump method of the handle bioreactor alerts rap?

12/04 14:24:14 - no

12/04 14:24:14 - USI: O.K. Here are the original tasks for nitrifier-pressure-low-stop-pump method.

task-net
sequence
Step t1 is NITRIFIER-TUBE-FLOW STOP
Step t2 is PUT-OUT-PBBWP-WARNING

Since the modification is actually made on a copy of the original procedure, a new SP is begun to undo whatever is wrong.

12/04 14:24:21 - USI: Which do you wish to re-select, the task, the modification or the method?

12/04 14:24:31 - the method

12/04 14:24:34 - USI: Here are the methods for handle bioreactor alerts <previous listing omitted>

12/04 14:24:37 - USI: Which one do you want to modify?

12/04 14:24:45 - forget it

12/04 14:24:46 - USI: Okay, I'll forget it.

"Forget it" signals the complete abandonment of the shared plan. Memory variables are left in tact for when the user

wants to "try again", in which case the SP processing picks up where it left off, or the user can "start over".

The above interaction takes place on-line while 3T is controlling the WRS. When the modifications are completed, at the next cycle of the RAPs agenda, the new procedure is invoked. After that, during the maintenance operations, 3T displays the low-pressure warning but takes no action. When the maintenance operations are complete, the user requests that the original procedure be restored. If a low-pressure situation occurs subsequently, the ASD will trigger. A laptop demonstration of the entire interactive procedure modification and control system responses is available.

Discussion

From the example dialog shown (and others which space does not allow) we believe two tentative conclusions can be drawn. First, the DPMA system is a viable candidate for a discourse management engine. It not only provides for easily devised and adjustable discourse plans, but also allows the interweaving of those plans with domain plans. Second, our streamlined shared plans processing approach as outlined in Figure 1, is a useful model of human-machine interaction, where the machine is considered a kind of idiot-savant and the user must be guided into areas of discourse related to what the machine knows.

The implications of being able to modify the procedural reasoning of an agent, while the agent is "on-line" are many, ranging from the ability to make temporary changes during a development tests to modifying a procedure in a deployed system after learning more about the environment. During our efforts in the integrated water test, there were several cases similar to the given example where we were asked to temporarily suspend automatic shutdowns due to crew intervention. Instead of building and integrating ad hoc interface buttons for the user, procedure modification discourse would have been less costly and the users could be trained to follow the discourses themselves.

One can also imagine a possible scenario on a manned outpost on Mars. The WRS control system has been programmed to shutdown the biological water processor (BWP) using the following procedure:

- 1) Turn off the effluent pump
- 2) Turn off the feed pump
- 3) Turn off the recycle pump
- 4) Close the recirculation valve

Later, the crew determines that the normally open air vent from the gas liquid separator (GLS) tends to clog during dust storms and needs to be closed as part of the shutdown procedure. With the DPMA discourse management system, the following dialog would eventually be possible, via either keyboard or voice.

ASTRONAUT: WRS, I want to change the BWP shut down procedure.
WRS: Okay. I have located the BWP shut down procedure. (Procedure steps are displayed in text)
ASTRONAUT: After you close the recirculation valve, close the air valve.
WRS: I am not familiar with the air valve. Here is a list of the BWP valves. (A list with names known by the system is shown.)
Which one are you referring to?
ASTRONAUT: The GLS vent valve.
WRS: Okay. I have added the step, close the GLS vent valve. (The new procedure is displayed in text.)
Anything else?
ASTRONAUT: No.
WRS: Okay.

We have not yet conducted user trials, but we have given a laptop demonstration of this system to a number of both knowledgeable and naive users. In all cases the users understand the concepts and what we are trying to accomplish, but report that there is more of a learning curve to understand the procedure structures than to understand how the discourse plans operate. These preliminary results point to a need for more graphical representations of the procedures and their methods.

References

- Bonasso, Pete. 2001. Intelligent Control of a NASA Advanced Water recovery System, in *Proceedings of The 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space: A New Space Odyssey*, Montreal, June.
- Bonasso, Pete. 2000. Teaching Humans to Talk to Robots, in Working Notes of the AAAI Spring Symposium on Natural Dialogues with Practical Robotic Devices, Stanford, March.
- Bonasso, Pete, Firby, R.J., Gat E., Kortenkamp, D., Miller, D. and Slack, M. 1997. Experiences with an Architecture for Intelligent, Reactive Agents, in *Journal of Experimental and Theoretical Artificial Intelligence*, 9, 237-256.
- Firby, James R.. 1999. The RAP System Language Manual, Version 2.0 Neodesic, Inc., Evanston, IL.
- Fitzgerald, W. and Wiseman, J. 1997. Approaches to Integrating Natural Language Understanding and Task Execution in the DPMA AERCam Testbed, Neodesic, Inc., October.
- Grosz, B. and Sidner, C. L. 1990. Plans for Discourse. In Cohen, Morgan, and Pollack, editors, *Intentions and Plans in Communication and Discourse*. MIT Press.

Lochbaum, Karen E. 1998. A Collaborative Planning Model of Intentional Structure. *Computational Linguistics* 24 (4): 525-572.

Schreckenghost, D., Bonasso, P., Kortenkamp, D. and Ryan, D. 1998. Three-Tiered Architecture for Controlling Space Life Support Systems. IEEE Symposium on Intelligence in Automation and Robotics. May.