The TRAINS 93 Dialogues

Peter A. Heeman and James Allen

The University of Rochester
Computer Science Department
Rochester, New York 14627

TRAINS Technical Note 94-2

March 1995

Abstract

This report describes a corpus of task-oriented dialogues set in the TRAINS domain. A user collaborates with a planning assistant to accomplish some task involving manufacturing and shipping goods in a railroad freight system. We include a description of the task, collection situation, and transcriptions conventions. The audio files, along with time-aligned word and phoneme transcriptions are available on CD-ROM from the Linguistic Data Consortium. Altogether, there are 98 dialogs included, collected using 20 different tasks and 34 different speakers. This amounts to six and a half hours of speech, about 5900 speaker turns, and 55000 transcribed words.

Funding gratefully received from the Natural Sciences and Engineering Research Council of Canada, from NSF under Grant IRI-90-13160, and from ONR/DARPA under Grant N00014-92-J-1512.

Contents

1 Introduction

One of the goals that we are pursuing at the University of Rochester is the development of a conversationally proficient planning assistant, which helps a user construct a plan to achieve some task involving the manufacturing and shipment of goods in a railroad freight system. To do this, we need to know what kinds of phenomena occur in such dialogues, and how to deal with them. To provide empirical data, we have collected a corpus of dialogues in this domain with a person playing the role of the system. The collection procedure was designed to make the setting as close as to human-computer interaction as possible, but was not a "wizard" scenario, where one person pretends to be a computer. Thus these dialogues provide a snapshop into an ideal human-computer interface that would be able to engage in fluent conversations.

In this paper, we present our current dialogue collection endeavor. Unlike our previous collections, we have used speech analysis tools in order to produce time-aligned word transcriptions. We supply both the transcriptions and the audio files on CD-ROM available from the Linguistic Data Consortium. This allows this corpus to be used for speech analysis purposes, such as speech recognition, and prosodic analysis.

This collection also expands on the number of different tasks, and the number of different speaker pairs. We have 20 different problem scenarios, and 25 different pairs of conversants. As well, for each pair of conversants, we have collected up to seven dialogues, each involving a different task. This should make this corpus suitable for studying problem solving strategies, as well as how agents collaborate in solving a task.

Since our corpus contains dialogues in which the conversants work together in solving the task, it provides natural examples of dialogue usage that computer systems will need to handle in order to carry on a dialogue with a user. For instance, our corpus contains instances of overlapping speech, back-channel responses, and turn-taking: phenomena that do not occur in collections of single speaker utterances, such as ATIS (MADCOW, 1992). Also, even for phenomena that do occur in single speaker utterances, such as speech repairs, our corpus allows the interactions with other dialogue phenomena to be examined.

In comparison to the HCRC Map Task corpus (Anderson et al., 1991), ours is also unique. The map task involves one agent trying to explain his route to another agent. Our domain involves two agents working together to construct a plan that solves some stated goal. So, the agents must do high-level domain planning in addition to communicative planning. Hence, our corpus allows researchers to examine language usage during collorative domain-planning—an area where human-computer dialogues will be very useful.

2 Overview of the Trains Project

The TRAINS project is a long-term research project to develop an intelligent planning assistant that is conversationally proficient in natural language (Allen et al., 1995). In particular, the TRAINS agent helps a person (the manager) construct and monitor plans about a railroad freight system. The manager is responsible for assigning cargo to trains and scheduling shipments, scheduling various simple manufacturing tasks, and for revising the original plan when unexpected situations arise during plan execution. The system aids the manager in all aspects of this task by interacting in natural language and (eventually) through a graphical interface. In particular, the system typically will perform the following tasks:

- evaluating suggested courses of action, such as calculating expected completion times, detecting conflicts that might interfere with the actions, and so on;
- filling in details of the proposed plan that do not require the manager's attention;
- suggesting ways to solve particular subproblems as they arise;
- presenting and describing the current state of the world and how the proposed plan may affect it, including answering questions from the manager;
- dispatching the plan to the different (simulated) agents in the world;
- interpreting reports back from the agents in the world in order to monitor the progress of the plan and to anticipate problems before they arise;
- coordinating the correction and modification of plans with the manager.

While we aim to produce a functional system, the system itself is not really the goal of the effort. Rather, the domain is a tool for forcing our research to address the problems that arise in building a complete dialogue system. For instance, the dialogue module must be able to handle a wide range of everyday discourse phenomena rather than handling a few selected problems of theoretical interest. While this approach focuses our research directions, the solutions that we seek are general solutions to the phenomena rather than specific solutions that happen to work in the TRAINS domain. The TRAINS project currently has several main foci of research:

- parsing and semantically interpreting utterances as they arise in spoken language, including sentence fragments, repairs and corrections, and long sequences of utterances that incrementally provide information that eventually combines to form a complete interpretation;
- accounting for the discourse behavior present in natural dialogue, *including* local coherence issues of reference and scope ambiguity, and global discourse issues such as topic flow and structure;
- representing the reasoning and control of the discourse agent, including reasoning about plans in the Trains domain as well as the reasoning processes that drive the system's behavior in the dialogue itself; and
- providing the knowledge representation and reasoning tools that are required to support planning and scheduling in realistic size domains, including reasoning about time, events, actions and plans.

3 Role of Dialogues in the Trains Project

There are a wide range of problems that we intend to use this data to help solve. These dialogues have been extensively used in our work on plan-based dialogue models and speech acts (e.g., Traum and Hinkelman, 1992), on discourse models for handling co-reference and scoping (e.g., Poesio, 1992), on detecting and correcting speech repairs (e.g., Heeman and Allen, 1994), and on other parts of the TRAINS project. Here are a few examples of how we have used, or are planning to use the data.

- parsing spontaneous dialogue. It is well known that spoken language is quite different in structure than written text. In addition, spontaneous dialogue seems very different than spoken language in a setting where speakers have time to plan and prepare their utterances in advance. This database will allow one to study the structure of interactive dialogue.

- speech repairs. In-line speech repairs are very common in interactive dialogue. We find about a quarter of the turns contain repairs: repetitions, corrections, and cancelations.

- prosody in spontaneous dialogue. While the study of intonation and prosody is an area of growing interest, there is little work done so far on spontaneous dialogue. More frequently, studies involve read speech, or utterances that are planned and prepared in advance of the utterance. The database will provide a rich source of data on repairs;

- turn-taking strategies. With the precise timing information available, more detailed studies of turn-taking will be possible than previously could be done using only analyses based on timing by hand. Effects of intonation, use of certain phrases, and so on, can be studied in relation to the turn-taking behavior observed;

- the use of referring expressions in dialogue. With a large database of dialogue, and some additional annotation on co-reference phenomena, various theories of reference resolution, such as centering, will be able to be empirically tested. In addition, the effect of segmentation into intonational phrases on reference phenomena can be studied;

- global discourse structure. Several theories of the global structure of discourse have been proposed, but none have been the subject of systematic study because of a lack of enough data in a well-behaved domain.

- speech act analysis. It is believed that common ways of phrasing utterances conventionally signal certain (indirect) speech acts. The database will allows us to study such issues quantitatively.

- the effect of general world knowledge on dialogue structure. A number of theories of topic flow and change have been proposed using plan-based models, but there has never been a systematic study of such theories based on a large corpora of dialogues.

4 Previous dialogue collections in the Trains Domain

There have been two prior dialogue collection efforts as part of the TRAINS project. Nakajima and Allen (1993) collected dialogues in the TRAINS domain as part of a study that examined prosodic clues to discourse structure. For the dialogues, a human played the role of the system, and both system and user were out of sight of each other communicating through microphones and headphones. The only common domain knowledge that the conversants shared was a map of the TRAINS domain, which included the cities, and locations of warehouses and factories.

The second dialogue collection, the TRAINS 91 dialogues, was undertaken by Gross, Allen, and Traum (1993). This collection employed the same collection paradigm as Nakajima and Allen, but with a simpler domain. Gross, Allen, and Traum found that this domain provided a complex enough setting for extended dialogues. For the dialogue collection, eight different users were recorded, with the same person playing the role of the system throughout. The system had information about the distance between cities and restrictions for shipping freight. Each pair did two problems: a warmup problem and a test problem. The same warmup and test problems were used by all pairs of participants.

5 The Trains 93 Dialogue Collection

The present dialogue collection is very similar to the second dialogue collection. The TRAINS map used in this collection, shown in Figure 1, differs only slightly from the one used in the 91 dialogues. However, there are some differences between the two corpora. First, more attention was payed to minimizing outside noise, and obtaining high-quality recordings. Second, the dialogues were transcribed using the Waves software (Ent, 1993), resulting in time-aligned transcriptions. Third, a larger number of speaker pairs and problems were used, resulting in a much larger corpus of dialogues. Third, less attention this time was spent in segmenting the dialogues into utterance units. Rather, we used a more pragmatically oriented view for segmenting the dialogues into reasonable sized audio files, suitable for use with Waves. This convention is described in Section 6.

The dialogues in this study were collected during three efforts. The first set were collected in the spring of 1992, encompassing dialogue **d92-1**, and was part of a trial run. In the fall of 1992, after small changes in the problem scenarios and in the user and system instructions, the **d92a** dialogues were collected. Then during the spring of 1993, after some further refinements in the instructions, we collected the bulk of the dialogues, the **d93** dialogues.

5.1 Setup

The dialogues were collected in an office which had partitions separating the two conversants. Dialogues were collected with Sennheiser HMD 414 close-talking microphones and headphones. The dialogues were recorded by a Panasonic SV-3900 Digital Audio Tape deck at a sampling rate of 48 kHz. In addition to the system and user, a coordinator, who ran

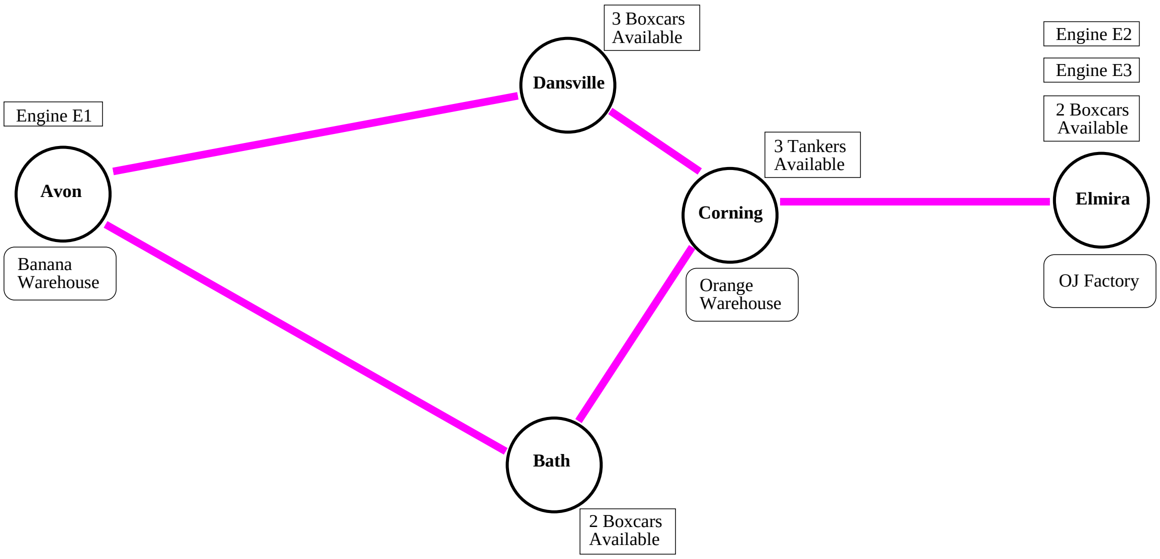


Figure 1: TRAINS Domain for 93 Dialogues

the experiments, was also present. All three could communicate with each other over microphones and headphones. But only the system and user's speech was recorded, each on a separate channel of a DAT tape. Both the user and system knew that the coordinator was overhearing, but would not participate in the dialogues, even if a problem arose.

At the start of the session, the user was given a copy of the consent form (Appendix A) to read and sign, as well as a copy of the user instructions (Appendix B) and user map (Appendix C). The user was not allowed to write anything done.

The system was given a copy of the system instructions (Appendix D) as well as copies of the system map (Appendix E). The system was also given blank paper and a pen, and was encouraged to use these to help them remember the plan, and answer the user's queries.

Once the user and system had read over the instructions, the coordinator had them practice on the warmup problem given in the user's instructions. For the **d93** dialogues, the warmup problem was not recorded, but was used to test the recording levels.

The participants then proceeded to do anywhere between two and seven more problems, depending on how many they could complete in the thirty minute session. All problems were arranged into three piles on the user's desk, with each pile corresponding to a different level of difficulty (see Appendix F). When ready to begin a dialogue, the coordinator would instruct the user to take a problem from the top of a certain pile. The first problem, after the warmup, was always from the easiest pile. For latter problems, the level of difficulty was chosen on the basis of how well the participants handled the previous problem and how much time remained.

After a problem was chosen, the user was given time to read the problem over (less than a minute). Once this was done, the user would signal by saying "ready". The coordinator would then set the DAT deck into record mode and push a button that would cause a green light to turn on at the user's and system's desk, which would signal the system to begin the conversation.

The coordinator would record the conversation until it was clear that the two participants had finished the dialogue. At that point, the user would hand the problem card to the coordinator, who would write the problem number (written on the back of the card) down on the recording log.

5.2 The subjects

The role of the system was played primarily by graduate students from the department of Computer Science and the department of Linguistics, except for CK, an undergraduate who was transcribing the TRAINS 91 dialogues using Waves. About half of the systems were involved in the TRAINS project. As for the users, we mostly used naive subjects who did the experiment as course credit for a cognitive science course. All participants were native speakers of North American English. Most dialogue partners were not known to each other, except in the case of dialogues **d92-1**, **d93-21**, and **d93-24**. The list of speakers is given in Appendix H.

5.3 Comments on Collected Dialogues

Participants sometimes made meta-comments about the task, or about the instructions. They also tended to make mistakes. Appendix G gives a list of the errors that were made, and Appendix I lists which errors occurred in each dialogue. These errors were discovered by reading throught the transcript of each dialogue. One particular problem was that the instructions were not very clear of what it meant for engines to be on the same track.

Of the 98 dialogues collected, 3 of them ended with the participants giving up, 23 of them ended with a plan that was not valid, and 14 more of them ended with a plan in which the execution time was miscalculated, but where the plan could execute within the required time constraints.

6 Segmenting a Dialogue

In order to work with a dialogue, we have segmented each dialogue into conveniently sized single-speaker segments that capture the sequential nature of the dialogue. These guidelines are taken from our technical note on dialogue transcription tools (Heeman and Allen, 1995), in which we also give worked examples, and describe tools that assist in this process. For the first pass, we made use of an automatic tool that segmented by silences. After the words were transcribed, we went back over the segmentation, and fixed up any problems by hand.

The guidelines rely on the fact that the two conversants are participating in a dialogue. So, the conversants tend to follow a turn-taking protocol and tend to break their speech into distinct units. The aim of the segmentation is the following.

A1: Each utterance file should be short enough so it is easy to analyze (not more than 12 seconds long), and so it does not include effects due to interactions from the other participant.

A2: Each utterance file should be long enough so that local phenomena are not split across utterance-file boundaries.

The first guideline should ensure that the sequence of single-speaker utterance files captures the sequential nature of the dialogue, thus allowing the flow and development of the dialogue to be preserved. In order words, the single-speaker utterance files should not contain or overlap a contribution by the other speaker. The second guideline ensures that the segments allow local phenomena to be easily studied, since they will be in a single file suitable for ToBI annotation. There can be conflicts between these two aims. If this happens, the first aim, (A1), should take priority.

In using the above guidelines for segmenting dialogues, we have operationalize what constitutes as a suitable place to break up a speaker's speech. We propose the following list of conditions, of which only one needs to hold. These are ordered by the appropriateness of the resulting break.

C1: A suitable break occurs in a speaker's speech whenever she stops and the other speaker starts (or continues), without the first trying to continue.

C2: A suitable break occurs whenever all of the following criteria hold.

1. There is an intonational phrase boundary.
2. There is a major syntactic category (i.e. NP or S) boundary.
3. There is a breath or pause.

C3: A suitable break occurs whenever the first two criteria of (C2) hold, and there is a break between the words (but shorter than a pause).

C4: A suitable break occurs whenever two of the three criteria of (C2) hold.

In segmenting a dialogue, utterance unit boundaries should be chosen so that (A1) and (A2) are satisfied using the strongest breaks possible.

7 Word Transcription

After the dialogues were segmented into utterance files, we used the Waves software to transcribe each utterance file, according to the conventions given in Appendix J.

For each dialogue, we first did a rough transcription, in which the words were transcribed without regard to the time-alignment of each word. After the rough transcriptions were finished, we used a program that scattered the word annotations evenly throughout the length of the utterance, and then hand-aligned each word using Waves. Halfway through this effort, we switched to using an automatic word aligner from Entropic Research Laboratories (Ent, 1994), which we fed the rough transcriptions. The result of the aligner was then hand-checked and corrected if necessary.

To test the accuracy of our transcriptions, we selected a set of utterances that spanned across the dialogue corpus. (This set of utterances consisted of 287 words.) We then extensively hand-checked the alignment, using spectrographs to increase the accuracy. We will refer to this alignment as the *reference* alignment. All silences, regardless if they were perceptable, were transcribed in the reference alignment. We then compared the reference alignment against our original hand-aligned transcription and the output of the aligner, ignoring discrepancies that fell within the silence markings of the reference annotations. Figure 2 shows the percentage of words whose end-points were aligned within a given amount of time of the reference end-point for the word. From this, we can see that 95% of the words were hand-transcribed to be within 50 ms of the reference end-point, and 83% of the automatically aligned words.

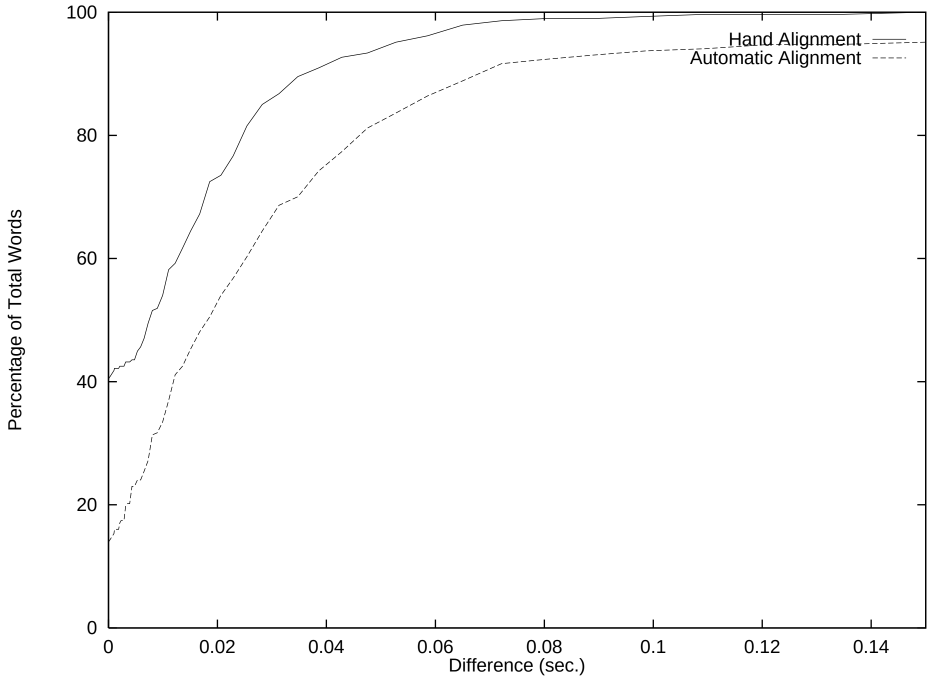


Figure 2: Difference in end points with respect to the reference alignments

Measuring the differences in end-points is not very informative. Some words, especially function words, can have a short duration, even less than 50 ms. Hence, in Figure 3, we measured the percentage of overlap. For each word, we took the overlap in the reference duration and the hand-transcribed duration of the word, and compared this to the average of the reference duration and the hand-aligned duration of the word. Again, differences that fell in the silence markings of the reference alignment were not counted. From the figure, we see that for the hand-alignment, 95% of the words overlap with the reference alignment by at least 77%; and for the automatic alignment, 78% of the words overlap with the reference alignment by at least 77%.

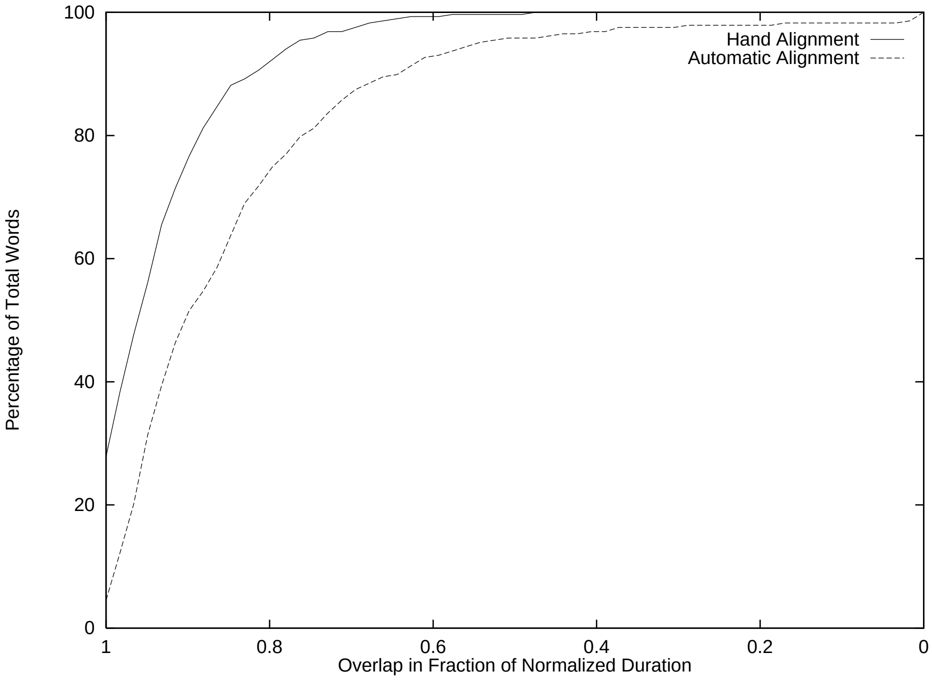


Figure 3: Overlap in word annotations with respect to reference alignment

8 Phoneme Transcription

We have also provided phoneme transcriptions. These transcriptions were all automatically obtained from the aligner, and are not hand-corrected. So, the accuracy of the phoneme alignment can be gauged by the accuracy of the automatic word alignment (from which they were obtained). As mentioned above, Figure 2 and Figure 3 give the results of how well the automatic word alignment did in comparison to the reference alignment. Since, 83% of the words have their end-points aligned within 50 ms of the reference alignment, the phoneme alignment should be about as accurate.

9 Using the Corpus

The dialogue collection is available on CD-ROM from the Linguistics Data Consortium. Each dialogue is in a separate directory under the directory **dialogs**. The name of the dialogue directory is in the form x-y.z, where x indicates which of the three collections, y denotes the system and user pair, and z denotes the dialogue number within the system and user's set of dialoges. The three collections are named **d92**, **d92a**, and **d93**.[1] Note that the **d92** dialogues consists of only one dialogue, which was named **d92-1**.

For each dialogue, we have included a text transcription, audio files, and time-aligned annotations. Table 1 gives a complete listing of the files associated with each dialogue. All

[1]The system and user pair indicators for the **d92** and **d92a** dialogs starts at 1, but for the **d93** dialogs, we made this number the same as the DAT tape number.

speaker$s$**.fea**	Sphere (NIST) Audio file for the speaker s. Speaker 0 is the system and speaker 1 is the user. Recorded at 16 kHz (except for **d92-1**, which was recorded at 8 kHz).
speaker$s$**.utts**	ESPS Annotation file that marks the utterance-level file boundaries for speaker s.
*dialog***.txt**	Formatted listing of the transcription of the dialogue, where *dialog* is the name of the dialogue. This is a standard ASCII file, and can be easily viewed.
speaker$s$**.words**	Annotation file that contains all of the hand-aligned words in the dialogue for speaker s.
speaker$s$**.display**	Annotation file that contains all of the words in the dialogue for speaker s, combined with marks that indicate the start and end of each utterance file (marked with **utt**i¡ and **utt**i¿. These files are provided so that the user can easily display the dialogue feature files along with the word and utterance segmentation (see Section 9).
speaker$s$**.awords**	Annotation files that contains the word transcriptions from an automatic aligner. These time alignments will be in agreement with the phonetic transcriptions.
speaker$s$**.phones**	Annotation files that contain the phonetic transcriptions produced by the automatic aligner (see Section 8).

Table 1: Dialogue Files

of these files are speaker specific, except the dialogue transcription. The audio files have been formatted using the NIST Sphere header structure, and are stored in compressed form. The annotation files are in the ESPS (Xlabel) format. Each annotation file has a header followed by a '#' symbol, followed by the annotations. Each annotation is on a separate line. The first column has the time-alignment (in seconds), the second a colour to display the symbol in (can be ignored), and the third has the actual symbol.

The simplest place to start using the corpus is with the text transcriptions of the dialogue (in ASCII). These are stored in the subdirectory **transcripts** as *dialog***.tex**, where *dialog* is the name of the dialogue (they are also in each dialogue directory).

To use the audio files, you will need to install the NIST Sphere toolset, which is included on the CD-ROM, in order to decompress the audio files. If you have access to Waves, the tool **displaydialog** can be run in the dialogue directories to display the audio files (after it decompresses it) and the annotations (**speaker**s**.display**). The Waves tools can display NIST Sphere files. Of course, any tool capable of handling NIST Sphere audio files can be used with the speaker audio files.

A third way of examining the dialogues is to segment them into single-speaker utterance files that preserve the sequential nature of the dialogue (see Section 6). A segmentation of the speaker files is included in each dialogue directory, and can be used for segmenting a dialogue into such a sequence of utterance files. In the **tools** subdirectory on the CD-ROM is a toolkit for managing such a segmentation. This toolkit makes use of Waves, and is described in a technical note (Heeman and Allen, 1995) included in the **docs** subdirectory. The program **expanddialog**, which makes use of the toolkit, can be used to create the utterance segmentation from the CD-ROM dialogue. This program is also in the **tools** subdirectory.

10 Acknowledgments

This dialogue collection was made possible by the help of many contributors. First, we thank Derek Gross and David Traum, authors of the report on the Trains 91 dialogues. We used their instructions, TRAINS map, and procedures as a starting point for this effort. We wish to the participants of James' seminar on dialogue issues, and in particular, Cheryl Beach, who revamped the problem scenarios and instructions. We thank Alice Kyburg who set up the 92 dialogue collection. We thank Brad Miller for overseeing the setup of the speech lab.

We thank the many participants, especially the graduate students who volunteered several hours of their time to play the role of the system.

We also thank Bin Li and Tsuneaki Kato for their help in formulating the heuristics for breaking up a dialogue. We also thank Bin, Greg Mitchell, Andrew Simchik, and Mia Stern for their help in annotating the dialogues.

References

Allen, James F., Lenhart K. Schubert, G. Ferguson, P. Heeman, C. H. Hwang, T. Kato, M. Light, N. G. Martin, B. W. Miller, M. Poesio, and D. R. Traum. 1995. The Trains project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*. To Appear. Also published as Trains TN 94-3 and TR 532, Computer Science Dept., U. Rochester, September 1994.

Anderson, Anne H., Miles Bader, Ellen Curman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry Thompson, and Regina Weinert. 1991. The HCRC map task corpus. *Language and Speech*, 34(4):351–366.

Entropic Research Laboratory, Inc., 1993. *WAVES+ Reference Manual*. Version 5.0.

Entropic Research Laboratory, Inc., 1994. *Aligner Reference Manual*. Version 1.3.

Gross, Derek, James Allen, and David Traum. 1993. The Trains 91 dialogues. Trains Technical Note 92-1, Department of Computer Science, University of Rochester, June.

Heeman, Peter and James Allen. 1994. Detecting and correcting speech repairs. In *Proceedings of the 32th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Las Cruces, New Mexico, June.

Heeman, Peter A. and James Allen. 1995. Dialogue transcription tools. Trains Technical Note 94-1, Department of Computer Science, University of Rochester, March. Revised.

MADCOW. 1992. Multi-site data collection for a spoken language corpus. In *Proceedings of the DARPA Workshop on Speech and Natural Language Processing*, pages 7–14, February.

Nakajima, Shin'ya and James F. Allen. 1993. A study on prosody and discourse structure in cooperative dialogues. Trains Technical Note 93-2, Department of Computer Science, University of Rochester, September.

Poesio, Massimo. 1992. Conversational events and discourse state change: A preliminary report. In *Proc. KR-92*, Boston, MA, October.

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech, and Jan Savrtvik. 1985. *A Comprehensive Grammar of the English Language*. London: Longman.

Traum, David R. and Elizabeth A. Hinkelman. 1992. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8(3).

A Consent Form

CONSENT FORM
Data Collection and Analysis
Principle Investigator: James Allen

We are interested in building a computer system that can help people in simple problem solving tasks using language. We are collecting some conversations, using a person to simulate the system, in order to better understand the task.

If you agree to participate in this study, you will be given problems to solve with the help of the "system". The "system" will have reasonably up-to-date information about the state of the world and you can ask it questions at any time. You, however, have the final say in what actions are to be included in your plan. The system will keep you informed of the situation when asked, provide you with information about the time required for manufacturing, loading, or shipping goods, and it will also try to help you, whenever possible, by anticipating problems that arise in your plan.

During the experiment, you will wear headphones and be speaking into a high quality microphone. The dialogs will be audio-recorded onto digital tapes. A picture of you will be taken to verify the microphone placement. During the experiment, you will not have visual contact with the system, nor will the experiment be video-taped.

The experiment carries no risk to you. The "system" role will not be competing against you, but is there to help in the problem solving endeavor. Also, the only benifits that you will receive is the satisfaction of solving the tasks. There is no monetary payment for participating.

By participating in this experiment, you agree to allow us to analyze and distribute the recorded conversations, but you will not be identified as the speaker in any way. Your participation is voluntary. At any time, you can withdraw with no change in circumstance.

____
Print Name

______________________________ __________
Signature Date

______________________________ __________
Principal Investigator Date

B User Instructions

The TRAINS World

We are interested in building a computer system that can help people in simple problem solving tasks. We are collecting some conversations, using a person to simulate the system, in order to better understand the task.

In this setting, you will playing the role of a manager who makes plans to manufacture and / or ship goods between cities as shown on the map on the desk. In your booth you wil see a clock, which gives the "current" time, just before midnight. On the desk is a stack of cars describing tasks that have come in from your boss before you came into work.

The "system" will have reasonably up-to-date information about the state of the world and you can ask it questions at any time. You, however, have the final say in what actions are to be included in your plan. The system will keep you informed of the situation when asked, provide you with information about the time required for manufacturing, loading, or shipping godds, and it will also try to help you, whenever possible, by anticipating problems that arise in your plan.

The map shows the layout consisting of several different cities connected by different rail lines. Some cities have warehouses for storage of goods. In this situation, the goods are either oranges or bananas. The rail lines are single track, so two trains cannot be on the same section of track at the same time.

In looking at the map, you will see the following:

Engines - labeled E1, E2, and E3
Boxcars for shipping - as indicated on the map
Tanker cars, at Corning, to be used for shipping OJ
Two warehouses - one for oranges at Corning, and one for bananas at Avon
An OJ Factory at Elmira - where a boxcar of oranges can be converted into a tanker of OJ

While it may look complicated, it really is quite simple once you get started. Remember that the "system" is available to help you at any time. The best way to learn is by trying a sample problem. On the following page, there is a warmup problem. Please turn the page now, and then go on to solve the problem.

Have fun!

Warmup Problem: You need to design a plan for shipping a boxcar of oranges to Bath by 8 AM today.

Remember that you may obtain further information and confim facts about the TRAINS world, including scheduling information, by asking the system. Also, you may use the layout map in any way that you wish to help you plan. The system does not know what particular problem you currently need to solve.

When you have finished the sample problem, please turn the page for further instructions.

INSTRUCTIONS

To begin the session:

First, we record background information on the tape to identify it.

1) The user first states his or her name, the date and time, sex, and mother tongue.

2) Then, the system identifies his or her name, sex, and mother tongue.

There will be a pause, and the co-ordinator will tell you when to start:

1) Say "Ready"

2) The tape is turned on

3) The system says "Hello, can I help you"

4) You may now proceed to solve the problem.

5) When you feel the session is complete, end the dialog by saying "I am now finished"

ADDITIONAL DIALOGS

On the table in front of you, you will see a set of cards that your boss left for you. Each card contains a different sample problem. Choose one of those cards now, and then go on to solve the problem in cooperation with the system. For each problem, ignore the fact that you have solved other problems previously. Treat each one as though it was the first one of the day. As a consequence, the world is in the same state (as shown on the map) at the start of each problem solving session, and the starting time is always just before midnight.

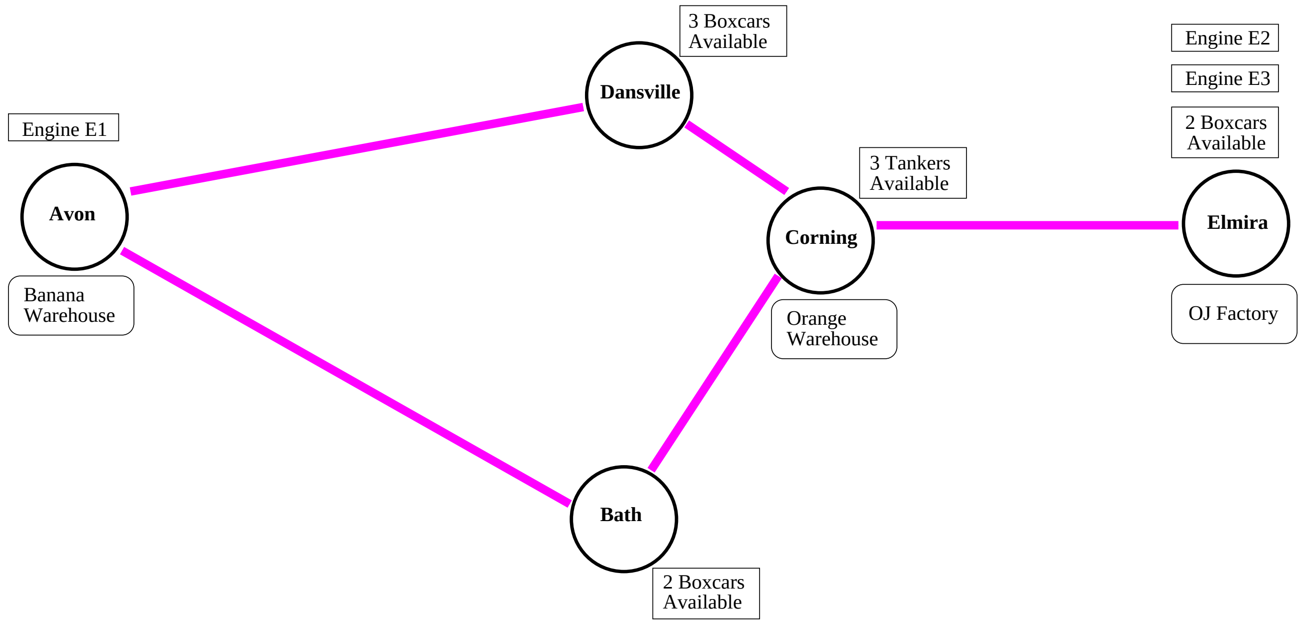
As before, use the following protocol for beginning and ending the dialog.

Wait for the co-ordinator to tell you when to start:

1) Say "Ready"
2) The tape is turned on
3) The system says "Hello, can I help you"
4) You may now proceed to solve the problem.
5) When you feel the session is complete, end the dialog by saying "I am now finished"

C User Map

TRAINS World Map



D System Instructions

Guidelines for the "system" in the TRAINS world

We are interested in building a computer system that can help people in simple problem solving tasks. We are collecting some conversations, using a person to simulate the "systemk", in order to better understand the task. You will be playing the role of the "system" in these dialogs.

You will be assisting a "human" user in making plans to manufacture and / or ship goods between cities. The user is provided with a layout map showing several different cities connected by different rail lines. The user is told that some cities have warehouses for storage of goods (either oranges or bananas). The rail lines are single track, so two trains cannot simultaneously occupy the same section of track. Notice that it takes time to load and unload cargo, and to make OJ; also, there are limits to how many cars an engine can pull. Refer to the master plan for information about loading times and load limits.

In looking at the map, you will see the following:

Engines - labeled E1, E2, and E3
Boxcars for shipping - as indicated on the map
Tanker cars, at Corning, to be used for shipping OJ
Two warehouses - one for oranges at Corning, and one for bananas at Avon
An OJ Factory at Elmira - where a boxcar of oranges can be converted into
a tanker of OJ

As the system, you should maintain reasonably up-to-date information about the state of the world at any time, and the user can ask the system questions. The human user, however, has the final say in what actions are to be included in the plan.

System functions

1. When requested, provide the user with information that was not directly given to them at the start of the problem (e.g., how long it takes to get from Avon to Bath).
2. When requested, clarify or explain facts about the TRAINS world (e.g., orange juice can only be transported in a tanker).
3. As the plan is being developed, remember / keep track of where the objects are.
4. When requested, provide the user with a summary of the plan at that point in the planning.
5. As the plan is being developed, compute whether the user's plan will work and monitor for potential problems.
6. If the system detects a problem with the plan, the user should be told right away (e.g. "There is a problem because the cargo won't get there in time").

To get familar with the domain, try to answer the following question:

How long will take for engine E3 to pick up a boxcar at Elmira, go to Corning, load the boxcar with oranges and couple on a tanker car, take the load back to Elmira, make OJ and load it into the tanker car, uncouple the boxcar and then take the OJ to Avon?

Interacting with the user

The user may obtain further information and confirm facts about the TRAINS world, including scheduling information, by asking the system. Also, the user may use the layout map to help them plan.

In general, let the user try to solve the problems with the plan. In performing system functions, try to be as helpful as possible without being "too smart". In other words, avoid volunteering alternative solutions to the problem unless it seems necessary. Simple refinements to the plan for efficiency reasons can be offered (e.g., recommending using the same train to carry 2 different cargoes).

The system should not be "too limited" either. Try to be as natural as possible in your interactions with the user. The system can make some simple inferences, and the user should not need to be overly or artificially specific. In particular, you don't need to oversimplify your language, just use regular conversational language. When the user's intentions are not clear, the system can ask for clarifications.

The TRAINS world problems

The first problem the user will work on is a relatively easy, warmup problem. After the warmup problem, the user will be give a set of cards. Each card will contain a different problem. The user will choose which problem they want to work on. There are three sets of cards; each set contains a collection of either easy, medium, or relatively difficult problems. At the first session, users will be given the easy problem set; at subsequent sessions, they will be given increasing difficult problem sets than at the beginning of the session. The user has been told that the system will not initially know what problem the user currently needs to solve.

Be sure to quickly write down what goal the user has when you find out. It is surprisingly difficult to remember what the goal is without writing it down.

INSTRUCTIONS FOR COLLECTING DIALOGS

To begin the session:

First, we record background information on the tape to identify it.

1) The user first states his or her name, the date and time, sex, and mother tongue.

2) Then, the system identifies his or her name, sex, and mother tongue.

There will be a pause, and the user selects a problem card.

To solve a problem:

1) The user says "Ready"

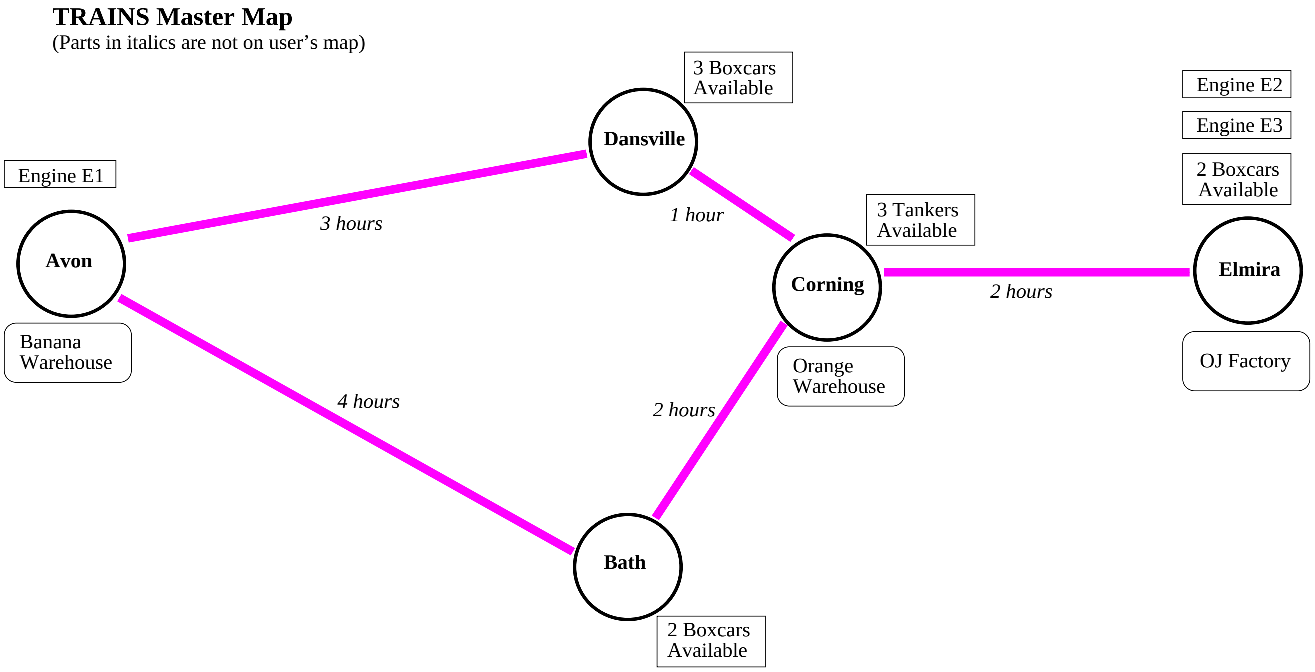
2) The tape is turned on

3) The system says "Hello, can I help you"

4) The dialog proceeds as long as it needs to

5) When the user feels the session is complete, the dialog is completed by the user saying "I am now finished"

E System Map



Timing Information:
It takes 1 hour to load or unload any amount of cargo on a train
It takes no time to couple or decouple cars
Manufacturing OJ: One boxcar oranges converts into one tanker load. Any amount can be made in one hour.

Capacity of Engines
An Engine can pull at most three loaded boxcars or tanker cars, and any number of unloaded cars.

F Problem Scenerios

The following are the problem scenarios that we used. They are classified into three levels of difficulty, 1, 2, and 3. Below, for each problem, we also give an analysis of the issues that need to be addressed to find a solution to the problem. Times shown in brackets, e.g. (5 a.m.), are the time by which the previous action will have been completed.

Problem W

You need to design a plan for shipping a boxcar of oranges to Bath by 8 AM today.

Analysis of Problem
This is the warmup problem. It is straight forward, and has a number of solutions.

Solution
Use engine E2 (or E3) from Elmira, couple a boxcar to it, and head to Corning (2 a.m.), where the boxcar can be loaded with oranges (3 a.m.). Then head to Bath, arriving there at 5 a.m., three hours ahead of schedule.

Other solutions will work. E2 could pick up a boxcar in Dansville (3 a.m.), head to Corning (4 a.m.), load up oranges (5 a.m.), and then head to Bath (7 a.m.). Or use engine E1 and pick up a boxcar in Dansville (3 a.m.) and continue as above.

Problem 1-A

Transport a tanker of OJ to Avon by 3 PM. It is now midnight.

Analysis of Problem
This is a sequencing problem: the user must ensure that the oranges are brought to the factory, orange juice is manufactured, and then picked up in a tanker car. Due to the time constraints, the user must bring a tanker to the factory at the same time as he brings the boxcar of oranges. Other than that, there are relatively few restrictions on choice of engine, boxcars, and choice of route.

Solution
One solution is to use engine E2 (or E3), and take a boxcar from Elmira to Corning, arriving there at 2 a.m. Load oranges into the boxcar (3 a.m.) and couple a tanker. Then head back to Elmira (5 a.m.), and make orange juice (6 a.m.). Then go to Corning (8 a.m.), Dansville, (9 a.m.), and then arrive at Avon by 12 p.m.

A second solution is to use E1 from Avon. Go to Dansville (3 a.m.), and pick up a boxcar. Then continue on to Corning (4 a.m.), and load up the boxcar with oranges (5 a.m.) and hook up a tanker. Then travel to Elmira (7 a.m.) and make orange juice (8 a.m.). Then go to Corning (10 a.m.), Dansville (11 a.m.), and then to Avon, arriving there at 2 p.m.

If E2 (or E3) is used, a boxcar at Dansville can be used instead of the one at Elmira, or the train can go through Bath, instead of Dansville, on its way to Avon.

Problem 1-B

Transport 2 boxcars of bananas to Corning by 11 AM. It is now midnight.

Analysis of Problem

This is a straight forward problem, but with a tight time constraint. Hence there are few choices for engine, boxcar and route that will result in a plan that finishes on time. In fact, all valid plans must travel via Dansville to get from the banana warehouse in Avon to Corning.

Solution

If engine E1 (from Avon) is used, it must pick up two boxcars in Dansville (3 a.m.), so that it can return to Avon by 6 a.m. The bananas should then be loaded (7 a.m.) and then transported to Dansville (10 a.m.) and then to Corning, by 11 a.m. Going to Corning via Bath will result in the task being completed two hours late.

If E2 or E3 (from Elmira) is used, then it must travel to Corning (2 a.m.), to Dansville (3 a.m.), and then to Avon, arriving there by 6 a.m. Boxcars can be picked up along the way either in Elmira or in Dansville. The rest of the plan is the same as for engine E1.

Problem 1-C

Transport 4 boxcars of oranges to Bath by 2 PM.

Analysis of Problem

This problem involves load-limit constraints. The user must get four boxcars of oranges to Bath, but an engine can only pull 3 full boxcars. So, he must either make two trips or use two trains. Also the user must get boxcars from more than one location, since no one spot has 4 boxcars. So, the user must be careful not to schedule two trains onto the same track.

Solution

One straightforward solution is to use E2 (or E3) to deliver the two boxcars from Elmira to Corning arriving there at 2 a.m. Load up the boxcars with oranges (3 a.m.), and then take them to Bath, ariving there at 5 a.m. At the same time, take E1 to Dansville (3 a.m.), pick up two boxcars and take them to Corning, arriving there at 4 a.m., which will be after E2 has already left. Load up the boxcars with oranges and arrive at Bath by 7 a.m.

Due to the abundance of time, many other solutions will work, including using E3 to pick up the boxcars in Dansville; or using E2 to pick up all of the boxcars first, then load oranges, and then make two trips to Bath to deliver the oranges.

Problem 1-D

Ship 3 boxcars of bananas and 3 tankers of OJ to Bath.

Analysis of Problem
This problem involves sequencing the actions to achieve two tasks. The first task is to make three tankers of oranges, for which the user first needs to get three boxcars of oranges, and convert them into three tankers worth of OJ. The second task is to ship three boxcars of bananas. These two tasks can either be done sequentially or in parallel.

Solution
The fastest way to accomplish this task is to tackle the two problems separately. Both subtasks need three boxcars each. The orange task can be done by taking engine E2 (or E3) with the two boxcars at Elmira to Corning (2 a.m.), and then to Bath (4 a.m.), where a third boxcar can be picked up. Then head back to Corning (6 a.m.), where the three boxcars can be loaded with oranges (7 a.m.) and the three tanker cars can be added. Then head to Elmira (9 a.m.), make the orange juice (10 a.m.), and then head to Corning (12 a.m.) and then to Bath by 2 p.m. This finishes the orange juice subtask. As for the bananas, take E1 from Avon to Dansville (3 a.m.), hook up the three boxcars, and return to Avon (6 a.m.). There, load up the three boxcars with bananas (7 a.m.), and then head to Bath, arriving there at 11 a.m.

Since there is no time constraint, many other solutions are possible. One other solution worth mentioning is using one engine to do the entire task. Use engine E2 from Avon, and head to Dansville (3 a.m.), hook up the three boxcars and head to Corning (4 a.m.), and couple up the three tankers and fill the boxcars with oranges (5 a.m.). Then head to Elmira (7 a.m.), make orange juice (8 a.m.), thus emptying the three boxcars. Then head to Corning (10 a.m.), and then to Bath (12 p.m.), and drop off the three tankers of orange juice. Then head to Avon (4 p.m.), and load up the boxcars with bananas (5 p.m.), and then head back to Bath, finishing the task by 9 p.m.

Problem 1-E

Engine E1 leaves Avon every night at midnight, picks up boxcars in Dansville, and then continually runs back and forth between Avon and Dansville until 9 AM, carrying 3 boxcars of bananas to Dansville on every trip. Your task is to transport 2 boxcars of oranges to Avon by 9 AM. It is now 12 midnight.

Analysis of Problem
This problem involves resource conflicts; the most direct route between Corning and Avon, by way of Dansville, is occupied, and so the alternate route through Bath must be used.

Unfortunately, the scenario presented in the problem uses boxcars that are not part of the TRAINS domain given on the user's or the system's map. In fact, in order to make an integer number of trips, E1 must drop off the boxcars in Avon, rather than unload them and take them back to Dansville. This problem proved difficult for the conversants. Of the four users given this problem, two ignored the constraint, one misread it, and only one pair was able to solve the problem.

Solution
Use Engine E2 (or E3), and take 2 boxcars from Elmira and go to Corning (2 a.m.), and

load them up with oranges (3 a.m.). Then go to Bath (5 a.m.), and then to Avon, arriving there at 9 a.m.

Problem 2-A

There are 5 boxcars of oranges waiting for you at Corning. You are to make 1 tanker of OJ, and then deliver the 1 tanker of OJ plus 4 boxcars of oranges to Bath as soon as possible.

Analysis of Problem
This problem involves dealing with load-limit constraints: an engine can only pull three loaded boxcars. So, the user must break the problem into two parts (or more) and execute them simultaneously. Many solutions are possible. The one below does the task with three engines, thus finishing the entire task by 9 a.m., with the four boxcars of oranges arriving there at 6 a.m.

Solution
Take engine E2 from Elmira to Corning (2 a.m.) and pick up one of the boxcars of oranges and a tanker and take them to Elmira (4 a.m.). Make a tanker's worth of orange juice (5 a.m.), then take it to Corning (7 a.m.), and then to Bath, arriving there at 9 a.m. While this is happening, take engine E3 and depart from Elmira at 1 a.m. (since it cannot depart at the same time as E2), and arrive in Corning at 3 a.m. In Corning, hook up three of the boxcars and take them to Bath, arriving there at 5 a.m. For the last boxcar, use engine E1 from Avon, and go to Dansville (3 a.m.), Corning (4 a.m.), and hook up the last boxcar. Then take it to Bath, arriving there at 6 a.m.

Problem 2-B

Transport 1 tanker of OJ to Avon and a boxcar of bananas to Corning by 3 PM.

Analysis of Problem
This is the test problem from the 91 dialogues (Gross, Allen, and Traum, 1993). This problem involves time constraints, and sequencing of actions, and can only be solved by using two trains and doing each task separately, but watching for harmful interactions.

Solution
Take engine E1 to Dansville (3 a.m.), pick up a boxcar, and return to Avon (6 a.m.). Load the boxcar with bananas (7 a.m.), then go to Dansville (10 a.m.) and then Corning, arriving there at Corning at 11 a.m. While this is happening, take engine E2 (or E3) with one boxcar from Elmira and go to Corning (2 a.m.). Load the boxcar with oranges (3 a.m.) and then pick up a tanker, and return to Elmira by 5 a.m. At Elmira, make orange juice (6 a.m.) and then go to Corning (8 a.m.), then to Dansville (9 a.m.), and then wait there for an hour for E1 to clear the track. Then at 10 a.m., head to Avon, arriving there at 1 p.m.

Problem 2-C

Ship 3 boxcars of bananas to Dansville, and 2 boxcars of oranges to Avon. Both shipments must arrive at their destinations by 9 AM. It is now 12 midnight.

Analysis of Problem
This problem is not solvable, and was pulled from the stack of problems before it was ever used.

Solution
Getting 3 boxcars of bananas to Dansville will take at least 10 hours. So, this problem is not solvable.

Problem 2-D

Ship 1 boxcar of bananas, 1 boxcar of oranges, and 1 tanker of OJ to Bath. All 3 commodities must be in Bath by noon. It is now 12 midnight.

Analysis of Problem
The difficulty with this problem is that there are three separate tasks that must be accomplish. Due to the lenient time constraints, many solutions are possible.

Solution
For the bananas, take engine E1 from Avon to Dansville (3 a.m.), pick up a boxcar, and then return to Avon (6 a.m.). Load the boxcar with bananas (7 a.m.), and head to Bath, arriving there at 11 a.m. For the oranges and orange juice, take engine E2 (or E3) with two boxcars from Elmira to Corning (2 a.m.), load up the boxcars with oranges (3 a.m.) and hook up a tanker. Then return to Elmira (5 a.m.), and make one tanker's worth of orange juice (6 a.m.). Then go to Corning (8 a.m.), and then Bath, arriving in Bath at 10 a.m.

Problem 2-E

Determine the maximum number of boxcars of oranges that you could get to Bath by 7 AM tomorrow morning. It is now 12 midnight.

Analysis of Problem
The difficulty with this problem is that the user must ensure that there is not a better plan that can bring even more boxcars of oranges to Bath, and hence must consider alternate solutions.

Solution
Take engine E2 with the two boxcars at Elmira, and go to Corning (2 a.m.), where they can be loaded with oranges (3 a.m.). Then head to Bath, arriving there at 5 a.m. At the same time, take E1 from Avon, go to Dansville (3 a.m.), and hook up 3 boxcars. Then head to Corning (4 a.m.), and load up the boxcars with oranges (5 a.m.). Then go to Bath, arriving there at 7 a.m. The two boxcars in Bath cannot be filled and brought back to Bath by 7 a.m., so the maximum is 5 boxcars.

Problem 2-F

Ship 3 boxcars of bananas to Bath and 2 tankers of OJ to Dansville. All deliveries must be made before noon.

Analysis of Problem
This problem has two easy subtasks, but the user must travel all over the map.

Solution
For the bananas, use engine E1, and go to Dansville (3 a.m.), and pick up the three boxcars there. Then return to Avon (6 a.m.) and load up the boxcars with bananas (7 a.m.) and then go to Bath, arriving there at 11 a.m. For the orange juice, use engine E2 (or E3). Take the two boxcars from Elmira and head to Corning (2 a.m.). Load up the two boxcars with oranges and hook up the two tankers (3 a.m.). Then, go back to Elmira (5 a.m.) and make orange juice (6 a.m.). Then go to Corning (8 a.m.), and then to Dansville, arriving there at 9 a.m.

Problem 2-G

Ship 2 boxcars of bananas and 2 boxcars of oranges to Dansville. All cargo must arrive by 10 AM.

Analysis of Problem
This problem should prove relatively easy as long as two engines are used: one for the transporting the bananas and one for transporting the oranges.

Solution
Take Engine E1 from Avon to Dansville (3 a.m.), pick up 2 boxcars, and return to Avon (6 a.m.). At Avon, load up the two boxcars with bananas (7 a.m.), and go back to Dansville (10 a.m.). While this is happening, take engine E2 (or E3), with two boxcars to Corning (2 a.m.). Load up the two boxcars with oranges (3 a.m.), and go to Dansville, arriving there at 4 a.m.

Problem 3-A

Pick up 2 boxcars of oranges, make into 2 tankers of OJ and ship OJ from Elmira to Avon, arriving by 1 PM at the latest. It is now 12 midnight.

Analysis of Problem
This problem involves sequencing and a difficult time constraint. The user must schedule the empty tankers to be taken to the orange juice factory at the same time as the loaded boxcars are taken there. The solution below is the only solution that will work in time.

Solution
Take engine E2 (or E3) from Elmira with two boxcars and go to Corning (2 a.m.). Load up the two boxcars with oranges (3 a.m.) and hook up two tankers. Then go back to Elmira (5 a.m.) and make the oranges into orange juice (6 a.m.). Then ship the orange juice to Elmira, via Corning (8 a.m.) and Dansville (9 a.m.), arriving in Elmira at noon.

Problem 3-B

Plan a round trip from Avon to Elmira. On the way out, take 3 boxcars of bananas, and deliver 1 each to Bath, Corning, and Elmira (you do not have to unload the boxcars, just drop them off). Bring 2 tankers of OJ back to Avon (you'll have to get oranges and make the juice). You must arrive back in Avon within 24 hours.

Analysis of Problem

Relatively complex subgoal planning is involved. The user must realize that while delivering the bananas, two empty boxcars must be picked up for the transporting the oranges to the OJ factory.

In order to qualify as making a round trip, this problem must be solved with a single engine doing everything. Inappropriate solutions made use of one engine to transport the bananas and a second to handle the orange juice, which does not constitute making a round trip. One problematic aspect of the solution is that it requires returning to Avon twice, which might be a bit confusing.

Solution

Take engine E1 to Dansville (3 a.m.) and pick up the three boxcars there. Then head back to Avon (6 a.m.) and load up the three boxcars with bananas (7 a.m.), and then go to Bath (11 a.m.). At Bath, drop off the boxcar of bananas, and hook up the two empty boxcars. Then go to Corning (1 p.m.), where the boxcar of bananas should be dropped off, two empty tankers hooked up, and the two empty boxcars loaded with oranges (2 p.m.). At this point, E1 will have one boxcar of bananas, two boxcars of oranges, and two empty tankers. E1 should now head to Elmira (4 p.m.), drop off the boxcar of bananas there, and make the oranges into orange juice (5 p.m.). Then, with the two full tankers of orange juice, go to Corning (7 p.m.), to Dansville (8 p.m.), and then to Avon, arriving there at 11 p.m.

Problem 3-C

Transport 2 tankers of OJ to Avon and 3 boxcars of bananas to Elmira. The bananas must arrive in Elmira by 9 PM. You can only use Engine E2 to solve this problem because the other engines will be undergoing routine maintenance.

Analysis of Problem

The difficulty with this problem is that the user must solve both subtasks with only a single engine. Due to the constraint on the number of loaded boxcars that an engine can pull, the shipment must be transported in two trips. There are a number of possible solutions, since the time constraint is lenient. Below, we give two solutions, one that completes the banana task as quick as possible, and a second that completes the overall plan as quick as possible (and delivers the bananas by 9 p.m.).

Solution
To complete the banana task as quickly as possible, use engine E2 from Elmira. Hook up the two boxcars there, and go to Corning (2 a.m.). Drop off the two boxcars in Corning and load them with oranges, while engine E2 gets three boxcars of bananas. Take E2 to Dansville (3 a.m.), and hook up the three empty boxcars. Then head to Avon (6 a.m.), and load the three boxcars with bananas (7 a.m.). Then head back to Dansville (10 a.m.), then to Corning (11 a.m.), and then to Elmira (1 p.m.), where the bananas should be dropped off, eight hours ahead of schedule. Then go back to Corning (3 p.m.), and pick up two empty tankers and the two boxcars loaded with oranges. Then, head back to Elmira and make orange juice, finishing this by 6 p.m. Then take the two tankers of orange juice to Corning (8 p.m.), Dansville (9 p.m.), and then to Avon, arriving there at 12 a.m., the following day.

To complete the overall task as quickly as possible, make the orange juice first, and bring it to Avon when you go to pick up the bananas. Use Engine E2 with two boxcars, and go to Corning (2 a.m.). Load the boxcars with oranges (3 a.m.) and couple the two tankers to the train. Then go to Elmira (5 a.m.), and turn the two boxcars of oranges into two tankers' worth of orange juice (6 a.m.). Then head for Avon, via Corning (8 a.m.) and Dansville (9 a.m.), stopping in Dansville to couple a third boxcar. You should arrive in Avon by 12 p.m. In Avon, drop off the two tankers of orange juice and load the three empty boxcars with bananas, finishing this by 1 p.m. Then head for Elmira, by way of Dansville (4 p.m.) and Corning (5 p.m.), arriving in Elmira at 7 p.m., two hours before the deadline.

Problem 3-D

It is now 12 midnight. Transport 2 boxcars of oranges and 1 tanker of OJ to Avon. Due to heavy rail traffic, the timing of some of your stops has been predetermined as follows.

1. *The oranges are due to be processed at the factory in Elmira at 7 AM sharp. As soon as the OJ is made, the train must leave Elmira.*
2. *The train can't arrive at Avon before 3 PM because there is no room.*

Analysis of Problem
This problem involves working around time constraints. There are many different solutions possible due to the lenient time constraints. Note that the wording of the problem seems to imply that a single train be used; but this is not explicitly stated.

Solution
Take engine E2 (or E3) from Elmira with one boxcar and go to Corning (2:00 a.m.). Load the boxcar with oranges (3 a.m.) and hook up a tanker. Go back to Elmira (5:00 a.m.) and wait two hours (7 a.m.). Then make the orange juice (8 a.m.), and hook up a second boxcar. E2 should now consist of two empty boxcars and a tanker full of orange juice. Next, go to Corning (10 a.m.), and load the two boxcars with oranges (11:00 a.m.), and then head to Avon, by way of Dansville (12:00 p.m.), arriving there by 3:00 p.m.

Problem 3-E

Get 7 boxcars of oranges to Elmira by 9 AM. It is now midnight.

Analysis of Problem
The solution to this problem involves simultaneous use of engines on different tracks. The solution that we give below makes use of all three engines. Note that our solution has engine E3 following E2, one hour begind, on the same section of track.

Solution
Take engine E1 from Avon to Dansville (3 a.m.), hook up the three boxcars and go to Corning (4 a.m.). Load the boxcars with oranges (5 a.m.), and go to Elmira, arriving there at 7 a.m. Take engine E2 and go to Corning (2 a.m.), and then to Bath (4 a.m.), and pick up the two boxcars there. Go back to Corning (6 a.m.) and load up the boxcars (7 a.m.), and go to Elmira, arriving there at 9 a.m. Take engine E3 with the two boxcars, depart at 1 a.m. and go to Corning (3 a.m.), and load up the boxcars with oranges (4 a.m.), and return to Elmira (6 a.m.).

Problem 3-F

Ship 2 boxcars of bananas and 1 boxcar of oranges to Dansville by noon using only one engine. It is now midnight, and the shipment must be unloaded by 1 PM.

Analysis of Problem
The user must do the orange task first, in order to be in a position to accomplish the banana task on time.

Solution
Take engine E2 (or E3) with a boxcar from Elmira and go to Corning (2 a.m.), and load up the oranges (3 a.m.), and go to Dansville, arriving there at 4 a.m. Drop off the boxcar of oranges, and have it unloaded, while E2 picks up two empty boxcars and goes to Avon (7 a.m.). At Avon, load the two boxcars with bananas (8 a.m.), then go to Dansville (11 a.m.), and unload the boxcar, finishing by noon.

G Error Codes

This appendix gives a list of all of the errors that were made in the dialogues. We have divided the errors into two groups, those that we have attributed to the system, and those that we have attributed to the user. System errors include giving the wrong information (or wrong interpretation) of the domain rules, and computing the wrong amount of time it takes to complete the various tasks. Even if the error resulted from a user suggestion or comment, it is still attributed to the system, since she is responsible for correcting such mistakes. User errors, on the other hand, include problems with interpreting the task that was given; for instance, if part of the task is forgetten, or if the task is incorrectly understood. Even in cases where the system made the incorrect interpretation, or suggested a plan that didn't meet the user's task, the error is still attributed to the user.

G.1 System Errors

The system errors that were observed in the dialogues have been listed below. These errors have been categorized into three groups. The first involves resource conflicts. The second involves violations of the properties of objects. The third involves mistakes in computing the duration of actions. These errors are prefixed by **R**, **P** and **T**, respectively.

R1: *Resource conflict.* Some piece of rolling stock (i.e., an engine, boxcar, or tanker) was scheduled to be used for two different tasks at the same time.

R2: *Train Crash.* Two trains are scheduled on the same piece of track running towards each other. Since the lines are single tracks, they will collide.

P1: *Orange juice can not be transported in boxcars.* Both the system and user instructions mention that tanker cars are to be used for shipping orange juice.

P2: *Tankers can not move by themselves.* The system must have thought that a tanker car was not a piece of rolling stock, but was a tanker truck.

P3: *Orange juice is not available at the orange juice factory.* Orange juice must be made. The user must bring a boxcar of oranges to the factory to make a tanker's worth of orange juice.

P4: *It is not the case that an engine can pull at most three boxcars or tankers.* This should only be a restriction on full boxcars or tankers. The system map says "Any engine can pull at most three loaded boxcars or tankers cars, and any number of unloaded cars."

P5: *An action is missing.* The plan is missing some action that is necessary.

T1: *It takes one hours to unload oranges, make OJ, and load the OJ into tankers, not three.* The system instructions are not clear enough about this. The system map reads "One boxcar oranges converts into one tanker load. Any amount can be made in one hour." The intended interpretation of "Any amount can be made in one hour" is that it includes loading and unloading.

T2: *It takes an hour to load or unload any amount of cargo on a train, not an hour for each boxcar or tanker.* The system instructions specify this clearly.

T3: System confused the time it takes to get from one city an adjacent one.

T4: System forget to include the time it takes to complete some task.

T5: Some error occurred in computing the amount of time required. However, from the dialogues, it is not possible to pinpoint the exact error that was made.

One extra "error" that we noted involved scheduling trains going in the same direction on a rail line connecting two cities. The system instructions are not clear whether this is to be allowed. "The rail lines are single tracks, so two trains cannot simultaneously occupy the same section of track." Due to the frequency of occurence, we did not mark cases where the system scheduled the trains one hour apart. But we did mark occurences of the following, but did not view them as makeing the proposed plan invalid.

R3: *Two engines can not travel together or immediately following each other.* There should be at least an hour delay between trains goig in the same direction on the same track.

G.2 User Errors

The errors that the user made are given below. Some of the more common errors have been classified, while all others are grouped into **U**.

U1: User misunderstands what is meant by a "round trip".

U2: When asked to deliver a boxcar of some good, the user unloads the boxcar at the destination instead of leaving it loaded. We did not count such misinterpretations as making the plan invalid.

U3: The task called for the boxcar to be unloaded, which the user forget to do. We did not count such problems as making the plan invalid, as long as there was enough time left over to do this.

U4: User mixed up a time constraint, or forgot one.

U5: User misunderstood the task, or didn't finish it, or ignored part of it.

H List of Speakers

Below, we give the list of speakers. The first column gives a two character speaker code, used in Appendix I. The second column gives the sex of the speakers. The third column indicates whether the speaker was an undergraduate (**U**), a graduate student in computer science (**GC**), a graduate student in linguistics (**GL**), or a faculty (or staff) member (**F**). The fourth column indicates whether the speaker was involved with the Trains project. See additional comments in Section 5.2.

Speaker	Sex	Type	Trains
AM	M	U	
BA	M	U	
BM	M	F	Y
CB	F	U	
CK	F	U	Y
CR	M	U	
DT	M	GC	Y
EB	F	U	
EL	F	U	
EP	M	U	
GD	F	GC	
JA	M	F	Y
JC	F	U	
JH	F	GL	Y
JL	F	U	
JT	F	U	
KK	M	U	
KL	M	U	
LP	F	U	
MF	F	U	
ML	M	GC	
NB	F	U	
NH	M	U	
NM	M	F	Y
PH	M	GC	Y
PL	M	U	
PP	F	GC	
PS	M	U	
PV	M	GC	
RD	F	U	
RI	F	U	
SB	F	U	
SS	F	U	
TG	F	U	

I List of Dialogues

Below we give a table with information about each dialogue. The order of the list reflects the order in which the dialogs were collected. The first column gives the name of the dialog. The second and third give the codes for the user and system, which can be used to look up more specific information about them in the table in Appendix H. The fourth column gives the problem number, which can be used to access the problem descriptions in Appendix F. The fifth column gives the length of the dialogues in minutes and seconds.

The sixth column states whether a valid plan, which completes the specified task within the specified constraints, was found: "Y" indicates one was found, "N" indicates that the proposed plan is not valid, and "G" indicates that the system and user gave up and did not complete a plan. If the system miscalculated the time required for the plan, but it does in fact meet the time constraints, then the plan is marked as being valid.

The seventh column indicates the errors that the system or user made that were not later resolved and that impacted the planning process[2] The error codes are explained in Appendix G. If the same type of error was made more than once, its error code will be reported once for each occurence. Only errors that are part of the final plan were included. Errors that resulted in the initial plan being abandoned are reported in the comments column.

Finally, the eighth column indicates any other comments.

[2]Not all mistakes are viewed as errors. Only the following types of mistakes are included: those caused the time to be computed wrong; those that forced the user to reject the current plan; and those that caused the final proposed plan to be wrong.

Dialog	S/U	Prob.	Len.	Valid	Errors	Comments
d92-1	PH NM	2-E	3'09"	Y		Recorded at 8 kHz.
d92a-1.1	RD JA	W	1'24"	Y		System did not compute running time of plan.
d92a-1.2		2-E	4'23"	Y	R3	User referred to problem number, **2-E**.
d92a-1.3		2-A	4'58"	N	R3,U5,U5	User did not use the 5 boxcars mentioned in problem (**U5**) and delivered an extra boxcar (**U5**).
d92a-1.4		3-E	4'13"	Y	R3	User initially said she only needed six boxcars of oranges rather then seven, but ended up shipping seven (as required by the problem) anyways.
d92a-2.1	JC DT	2-D	10'32"	Y	T5	
d92a-2.2		2-B	2'49"	Y	T5	
d92a-3.1	LP DT	W	1'57"	Y		
d92a-3.2		2-G	7'10"	Y		System initially made error (**T2**), but then corrected himself later in the dialogue.
d92a-4.1	CR DT	W	1'23"	Y		
d92a-4.2		2-B	6'35"	Y	T5	
d92a-4.3		2-A	53"	G		System did not understand initial scenario, causing it to give up.
d92a-4.4		2-E	4'50"	Y		
d92a-5.1	EP DT	W	2'00"	Y		
d92a-5.2		2-D	13'09"	Y	T1	Error **T1** also occurred in initial plan, causing it to be abandoned.

Dialog	S/U	Prob.	Len.	Valid	Errors	Comments
d93-8.1	BA NM	1-A	1'16"	N	P3	
d93-8.2		2-G	2'30"	Y		
d93-8.3		3-A	7'54"	G		System made error **T1** (as well as **P4**, **T5**, and **R3**) in the initial plan, forcing the participants to give up.
d93-8.4		2-B	2'15"	N	T5,R2	The error in computing the time is consistent with error **T1**.
d93-9.1	NH PH	2-E	8'01"	Y		
d93-9.2		1-C	3'19"	Y		
d93-9.3		1-B	3'34"	Y		
d93-10.1	RI PH	1-C	1'46"	Y	R3	
d93-10.2		2-E	1'14"	Y	R3	
d93-10.3		3-A	2'03"	Y	T3	System confused the time from Dansville to Avon as one hour. However, the plan will still work.
d93-10.4		3-B	5'14"	N	R3,U1,R1	
d93-10.5		1-B	55"	Y		
d93-11.1	CB PH	1-D	6'02"	N	R1,T5	Resource conflict involving one of the boxcars from Dansville.
d93-11.2		1-E	5'32"	Y		
d93-11.3		2-D	4'42"	Y		
d93-12.1	SB PH	1-A	4'04"	Y		
d93-12.2		1-B	1'26"	Y		
d93-12.3		2-F	3'55"	Y		
d93-12.4		3-C	6'29"	Y		
d93-13.1	EL CK	1-C	6'25"	Y	R3	
d93-13.2		2-G	4'54"	Y		System initially thought Avon to Dansville was one hour (**T3**), but later reaslized the mistake.
d93-13.3		3-E	8'23"	Y	R3	
d93-14.1	NB CK	1-E	2'29"	N	U5	User ignored constraint of E1's scheduled route (**U5**).
d93-14.2		2-B	6'08"	Y		
d93-14.3		3-F	6'55"	Y	U3	Unclear if oranges got unloaded (**U3**).
d93-15.1	SS CK	1-D	2'14"	N	P3	System did not compute running time of plan.
d93-15.2		2-E	4'08"	Y		
d93-15.3		3-B	2'52"	N	T5	
d93-15.4		3-A	3'52"	N	T5,P5	System made error **T5** (as well as **P2**) in initial plan, causing it to be abandoned.Error **T5** occurred in initial plan, causing it to be abandoned.
d93-15.5		2-D	2'39"	Y	T5,R3	System miscalculated time by 1 hour (**T5**). This might have been done to avoid having two trains run at the same time (**U3**).

Dialog	S/U	Prob.	Len.	Valid	Errors	Comments
d93-16.1	AM CK	1-A	2'57"	Y		
d93-16.2		2-F	4'40"	Y		Difficult to determine if **R3** occurred in making the OJ.
d93-16.3		3-C	4'06"	Y	T5	
d93-16.4		3-E	6'17"	Y	R3	
d93-17.1	MF ML	1-B	1'58"	Y	T5,U4	User mixed up starting and ending times.
d93-17.2		2-G	4'11"	Y		
d93-17.3		3-F	2'34"	Y	U3	
d93-17.4		3-A	3'22"	Y		
d93-18.1	TG ML	1-C	3'04"	Y		
d93-18.2		2-B	5'40"	N	P3	
d93-18.3		3-B	4'16"	Y		
d93-18.4		3-D	8'34"	Y	R3	Although not explicitly disallowed, the user used two trains to accomplish the task.
d93-19.1	PS ML	1-E	1'20"	N	U5	User ignored constraint of E1's scheduled route (**U5**).
d93-19.2		2-E	2'41"	Y		
d93-19.3		3-C	3'47"	Y	T4	System forgot to include the time it takes to make OJ (**T4**). Also, unclear if tankers were included in the final plan.
d93-19.4		3-E	3'40"	Y	R3	
d93-19.5		3-F	4'20"	Y	U3	Unclear if oranges were unloaded **U3**.
d93-20.1	JT GD	1-D	1'41"	N	P2,P3	
d93-20.2		2-D	10'43"	Y	R3	
d93-20.3		2-F	1'56"	N	U5	User delivered OJ to Bath instead of to Dansville (**U5**).
d93-20.4		2-G	1'34"	Y		
d93-21.1	PV PP	1-B	42"	N	U5	User transported goods to the wrong city.
d93-21.2		2-B	4'27"	N	R2	
d93-21.3		2-E	2'55"	Y	R3	
d93-21.4		3-A	1'06"	N	P1	
d93-21.5		3-B	3'21"	Y		
d93-21.6		3-D	2'42"	Y		Some confusion about what time oranges were to be processed into OJ **U4**.
d93-22.1	EB JH	1-A	1'32"	N	P3,P2	
d93-22.2		2-D	7'57"	N	P3	
d93-22.3		3-F	6'12"	Y	T2,U3	
d93-23.1	PL JH	1-C	3'00"	Y	R3,U2	
d93-23.2		2-F	6'23"	N	R3,P3,U5	User delived OJ to Bath instead of to Dansville (**U5**).
d93-23.3		3-E	5'24"	Y	R3	Unclear if system remembered all three of the trains that the user scheduled.
d93-23.4		3-C	4'10"	N	U2,P3	Initial user misunderstanding about the time requirements.

Dialog	S/U	Prob.	Len.	Valid	Errors	Comments
d93-24.1	JL CK	1-E	3'31"	N	U5	User was unable to understand the constraints of E1's scheduled stops, and so ignored them (**U5**)
d93-24.2		2-G	3'31"	Y	U2	
d93-24.3		2-B	3'34"	Y		
d93-24.4		3-A	1'16"	Y		
d93-24.5		3-B	3'10"	Y		
d93-24.6		3-D	4'10"	Y		
d93-24.7		2-E	2'22"	Y		
d93-25.1	KK CK	1-D	3'19"	Y		Unclear if third tanker of OJ was properly made (**P5**).
d93-25.2		2-D	3'13"	Y	T5,R3	System miscalculated time by 1 hour (**T5**). This might have been done to avoid having two trains run at the same time (**U3**).
d93-25.3		2-F	1'16"	Y		System did not compute running time of plan.
d93-25.4		3-F	1'43"	Y	U3	
d93-25.5		3-E	7'36"	Y	T5,R3	Error **T5** (as well as **R3**) occurred in initial plan, causing it to be abandoned.
d93-25.6		3-C	2'27"	Y		
d93-25.7		3-D	4'32"	N	R1,U4,R3	Final plan involved filling up a boxcar twice (**R1**) and not waiting until 7 a.m. to process the oranges (**U4**).
d93-26.1	KL BM	1-D	6'13"	Y	U2,T5,T5	
d93-26.2		1-A	3'43"	Y	T1	Error **T1** and **T5** occurred in initial plan, causing it to be abandoned.
d93-26.3		2-E	2'31"	Y	R3	
d93-26.4		3-B	4'40"	N	U1,T1	Error **T1** occurred in initial plan, causing it to be abandoned.
d93-26.5		3-A	3'11"	G		Error **T1** occurred in initial plan, causing the system and user to give up.

J Word Annotations

The words spoken are entered in ESPS annotation files that end in **.words**, which are associated with each utterance file. Each word is indicated at the ending point of the word in the speech signal using its orthographic spelling. This should be the point that best separates the current word from the next so that when listening to each word individualy, there is a minimum of sound from the neighboring words. This will usually be at a trough in the amplitude of the speech signal.

J.1 Silences and Breaths

In order to capture word durations, we also mark other phenomena in the word layer. This includes breaths within the utterance, and noticeable silences, which disrupt the flow of speech. As with words, these are labelled at the end of the breath or silence, with a <**brth**> or <**sil**>, respectively. Only silences that are perceptible should be transcribed.

There is typically a silence or a breath at the beginning of an utterance file. These also need to be marked so that the duration of the first word can be computed.

J.2 Disfluencies and Partial Words

When possible, words should be transcribed using their standard spelling; do not try to capture the way a word was spoken by modifying its normal spelling. Disfluencies that are not identifiable words should be transcribed as they sound, with a dash, '-', at the end of the word. These include words that are so misspoken that the intended word is not recognizable. In cases when the word is recognizable but cut off, it should be labelled by spelling as much of the word as can be heard, followed by a dash. For example "oran-" would label the word "orange" that was interrupted before the "g" sound.

If the word that was cut off midstream, but the word that the speaker intended to say is obvious, then the rest of the word should be included in parentheses. For instance, in the above, it would be transcribed as "oran(ge)-".

Sometimes when a speaker is trying to interrupt the other person, she will start to say a word and then give up. The resulting sound should be transcribed as a partial word.

When multiple sounds are emitting in a row, they should be transcribed as a single partial word, unless there is a perceptible silence in between the two sounds.

J.3 Capital Letters and Proper Names

Capital letters should only be used to begin proper names, such as "Corning" and for the pronoun "I". Engine names, such as "E2" are problematic, since they could be viewed as a proper name. However, we will transcribe them as two words "E" and "two'.

A partial word should only be capitalized if its completion is clear, and that completion is a proper name. In this case, the completion will be put in parentheses. So, for the city "Corning", "Corn-" is not exceptable, but either "Corn(ing)-" or "corn-" is (the second case is where it is not clear to the transcriber that the speaker was going to say "Corning".

J.4 Mispronounced Words

Mispronounced words should in general be spelt correctly, unless the misspelling is needed in order to understand the subsequent dialogue. For instance, if the speaker says “Dansvull”, and later corrects herself by saying “Dansville”, then the first occurance should be spelt in a way that reflects the mispronunciation.

J.5 Numbers

Numbers should be spelled out in full, as they are said. For instance, if a speaker says “104” as “one oh four”, this is what should be transcribed, as three words.

J.6 Times

Each component of a time, should be spell out in full. The words “a.m.” and “p.m.” should be spelled as such. For instance “four p.m.” should be annotated as two words.

J.7 Common Words in Spoken Dialogues

There are a number of words that are common in spoken dialogues that do not occur in written forms. This section discusses how such words should be transcribed. Where possible, we use the spelling from Quirk et al. (1985).

Filled Pauses

Filled pauses are very common in natural dialogue. There seem to be two types, ones that sound like “uh” and ones that sound like “um”. The endings of these words are often prolonged, thus tempting transcribers to label it is “ummm”. Rather, these words should be classified as either “um” or “uh”, and transcribed as such. We also include “er”, which is more common in British accents. Note that the filled-pauses should never be transcribed as partial words.

um	Filled pause.
uh	Filled pause.
er	Filled pause. More common in British English.

Acknowledgments

The following is a list of commonly occurring acknowledgments, and how they should be spelt.

okay	Agreement. Speakers will often produce variants of this, such as “kay”, “mkay”, “umkay”. All of these variants should be spelt as “okay”
uh-huh	Agreement.

uh-hm	Agreement.
mm-hm	Agreement.
uh-uh	Disagreement.
mm	Agreement, stalling for time
huh	Request for clarification. Puzzlement.
hm	Stalling for time.
nah	Informal version of "no".
nope	Informal version of "no".
a-ha	Interjection denoting surprise, as in "aha! I found it", rather than "uh-huh" as an acknowledgement
ha	Interjection, similar to "aha".
oh	Surprise.
ooh	As in "ooh, that's gross."
yeah	Informal version of "yes"
yep	Informal version of "yes"

Contractions

Contractions, that are common, should be written as one word. The following is a list of common contraction endings. Note that there can often be an ambiguity as to whether the speaker was saying the words as one or as two individual words, especially since words are often blurred together. If in doubt, annotate the word pair as two separate words, spelling out the second in full.

'll	for "will"
've	for "have"
n't	for "not"
're	for "are"
's	for "is"

All other contractions are left to the transcriber's discretion as to whether they should be transcribed as one word or two.

Word Pairs

There are some word pairs that are so altered (in pronounciation) that they seem to be one lexical item. Such pairs can be transcribed as single words. Below, we give some common word pairs.

lemme	for "let me"
wanna	for "want to"
gonna	for "going to"
gotta	for "going to"

J.8 Other Phenomena

Laughter, tongue clicking, and other phenomenon can be transcribed in the word tier if it does not overlap with the speaker's speech. Such phenomena should be enclosed in angle brackets. Common transcriptions are given below.

<**click**>	Obvious tongue-clicking
<**clicks**>	Tongue-clickings. Usually made to fill a silence.
<**sigh**>	
<**clear-throat**>	Throat clearing, and coughs.
<**laughter**>	
<**mumble**>	Incomprehensible word or words. Perhaps not even intended for the other conversant to hear.
<**noise**>	Noises that the speaker unintentionally makes, as well as other noises.
<**filled-pause**>	Speaker-specific sounds that the speaker makes to fill a silences.