

CLASSiC

D6.1.3: Final Appointment Scheduling Data Collection for Task-Oriented Dialogue

Ghislain Putois, Romain Laroche, Philippe Bretier

Distribution: Consortium

CLASSiC

Computational Learning in Adaptive Systems for Spoken Conversation
216594 Deliverable 6.1.3

March 2010



Project funded by the European Community
under the Seventh Framework Programme for
Research and Technological Development



The deliverable identification sheet is to be found on the reverse of this page.

Project ref. no.	216594
Project acronym	CLASSiC
Project full title	Computational Learning in Adaptive Systems for Spoken Conversation
Instrument	STREP
Thematic Priority	Cognitive Systems, Interaction, and Robotics
Start date / duration	01 March 2008 / 36 Months
Security	Consortium
Contractual date of delivery	M24 = February 2010
Actual date of delivery	March 2010
Deliverable number	6.1.3
Deliverable title	D6.1.3: Final Appointment Scheduling Data Collection for Task-Oriented Dialogue
Type	Report
Status & version	final 1.1
Number of pages	32 (excluding front matter)
Contributing WP	6
WP/Task responsible	WP6, leader FT
Other contributors	Verena Rieser, Helen Hastie, Oliver Lemon, Steve Young, James Henderson, François Mairesse, Stéphane Rossignol
Author(s)	Ghislain Putois, Romain Laroche, Philippe Bretier
EC Project Officer	Philippe Gelin
Keywords	data collections, 1013, appointment scheduling

The partners in CLASSiC are:

Heriot-Watt University	HWU
University of Cambridge	UCAM
University of Geneva	GENE
Ecole Supérieure d'Electricité	SUPELEC
France Telecom/ Orange Labs	FT
University of Edinburgh HCRC	EDIN

For copies of reports, updates on project activities and other CLASSiC-related information, contact:

The CLASSiC Project Co-ordinator:
Dr. Oliver Lemon
School of Mathematical and Computer Sciences (MACS)
Heriot-Watt University
Edinburgh
EH14 4AS
United Kingdom
O.Lemon@hw.ac.uk
Phone +44 (131) 451 3782 - Fax +44 (0)131 451 3327

Copies of reports and other material can also be accessed via the project's administration homepage,
<http://www.classic-project.org>

©2010, The Individual Authors.

No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the copyright owner.

Contents

Executive Summary	1
I Introduction	2
1 System summary	2
2 The 1013+ service	2
3 Platforms	2
4 The appointment scheduling scenario	3
II The CLASSIC systems	5
5 The CLASSIC public variant (System 3)	5
6 The CLASSIC lab variant (System 4)	6
7 Evaluation, rewards, learning	10
III From Data Collection to Corpus Constitution	11
8 System 3	11
9 System 4	11
10 Annotation Scheme	11
IV Data	14
11 Industrial service	14
11.1 Specification	14
11.2 Preproduction	14
11.3 Pilot and Production	14
12 Lab service	14
V Specification Dialogues	16

A	University of Edinburgh’s example dialogues	16
A.1	Example EDIN1: Basic dialogue	16
A.2	Example EDIN2: Smoothly switching between alternate user goals	16
A.3	Example EDIN3: user provides precise time	17
A.4	Example EDIN4: user expresses a ranking of preferences and the DM/NLG	17
A.5	Example EDIN5: Alignment to temporal referring expressions, relative temporal references	17
B	France Télécom’s example dialogues	18
B.1	Example FT1: User changes his mind according to system availabilities	18
B.2	Example FT2: User changes his mind during dialogue	19
B.3	Example FT3: User constraints are disjunctives; System offers single slot	19
B.4	Example FT4: User constraints are disjunctives; System offers multiple slots	20
B.5	Example FT5: User constraints are inconsistent	21
C	University of Cambridge’s example dialogues	22
C.1	Example UCAM1	22
C.2	Example UCAM2	23
C.3	Example UCAM3	24
C.4	Example UCAM4	25
C.5	Example UCAM5	26
D	University of Geneva’s example dialogues	27
D.1	Example GENE1	27
D.2	Example GENE2	27
D.3	Example GENE3	28
D.4	Example GENE4	28
D.5	Example GENE5	29
E	Supelec Metz’s examples	29
E.1	Example SUPE1: Straightforward interaction	29
E.2	Example SUPE2: Straightforward goal	30
E.3	Example SUPE3: Goal with simple preference	30
E.4	Example SUPE4: Goal with simple preference + alternatives from the DM	31
E.5	Example SUPE5: Mistake in the initial goal	32

Executive summary

This document is the deliverable 6.1.3, due at month 24 of the CLASSiC project. It describes the data collection process implemented during the specification, the preproduction and the production of the appointment scheduling 1013+ commercial service. At the time of writing, approximately 400 dialogues per day are being collected. 100% of calls from French customers are routed to the CLASSiC system.

Part I

Introduction

1 System summary

This deliverable presents three appointment scheduling systems and their associated data collection process.

System	Short description	Deployment
Basic version	original 1013 version (system initiative only)	open to real customers
System 3	mixed initiative CLASSiC version	gradually being open to real customers
System 4	lab variant for experiments	tested by selected participants on a public phone line

2 The 1013+ service

The CLASSiC appointment scheduling research has been embedded inside the 1013+ service. The 1013+ service is dedicated to fixed phone line troubleshooting. It is composed of an interactive voice response system (IVR) and a pool of human operators. The IVR offers the following automation services: client identification, call reason recognition, phone line automated tests, and appointment scheduling. It aims to improve the user experience during the troubleshooting process, lowering the costs and freeing the human operator from tedious basic tasks.

3 Platforms

The 1013+ service is deployed in two data centres, which receive calls from the whole French territory. Each site contains its own voice platform servers and application servers. This process itself is composed of four main phases, each needing a formal agreement before going forward to the next phase:

- development
- preproduction
- pilot
- production

The development phase is the time when the service is designed according to a Functional Design Specification and illustrated by ideal use case scenarios. This phase ends with the unit and functional tests.

The preproduction phase sees the service deployed as a near-final version, but only testers can access the service. During this phase, every call to the service is monitored and completely analysed to try and detect anomalies or malfunctions not foreseen during the service specification. Such anomalies are then analysed to determine their functional and financial impacts on the service and on the return on investment. If the gain is not significant enough, some versions of the service can be cancelled.

The pilot phase is the first phase when the service is finally opened to real customers. At the beginning, only a very small subset of users can access the service, then the service is progressively generalised to the whole targeted subset of the population, while a complete monitoring and analysis of all the calls are done until the service is deemed stable and profitable enough.

Last, the production phase is the nominal phase when the system is fully operational and available to the entire French population.

4 The appointment scheduling scenario

The application domain of this document is the automatic appointment scheduling within the public France Télécom hotline. The 1013+ service is the France Télécom fixed-line troubleshooting hotline. When the customers call the hotline and when the reason for their call matches some given category, an appointment has to be scheduled between customers and a fixed-line repairman. The data collection described in this document concerns this appointment scheduling.

This section describes the basic appointment scheduling scenario. It is the dialogue strategy of the basic 1013+ version. Figure 1 provides an overview. The system first proposes a time schedule to the user. If the user agrees with this time schedule, the system accesses the back office to book the schedule. If the booking is okay, it confirms to the user that the booking is final and the dialogue ends. If the booking fails, then the user is transferred to an operator. If the user refuses the time schedule, then the system proposes another time schedule. According to the parameter n , the user can have n time schedules proposed to them, after that they are transferred to an operator. If the user does not say anything or if the ASR rejects the user utterance twice, then the user is transferred to an operator. An example dialogue is given below.

SYSTEM : D'après les tests effectués sur votre ligne, un rendez-vous avec un technicien est nécessaire à votre domicile. Le premier rendez-vous disponible est le suivant : le mercredi 28 octobre Le début d'intervention du technicien aura lieu entre 10 heure et 12 heure Est-ce que cela vous convient ? Merci de répondre par oui ou par non.

USER : oui

SYSTEM : Très bien. Nous enregistrons votre rendez-vous dans notre agenda. Merci de patienter quelques instants. Votre rendez-vous avec un technicien est confirmé pour le mercredi 28 octobre avec un début de l'intervention entre 10 heure

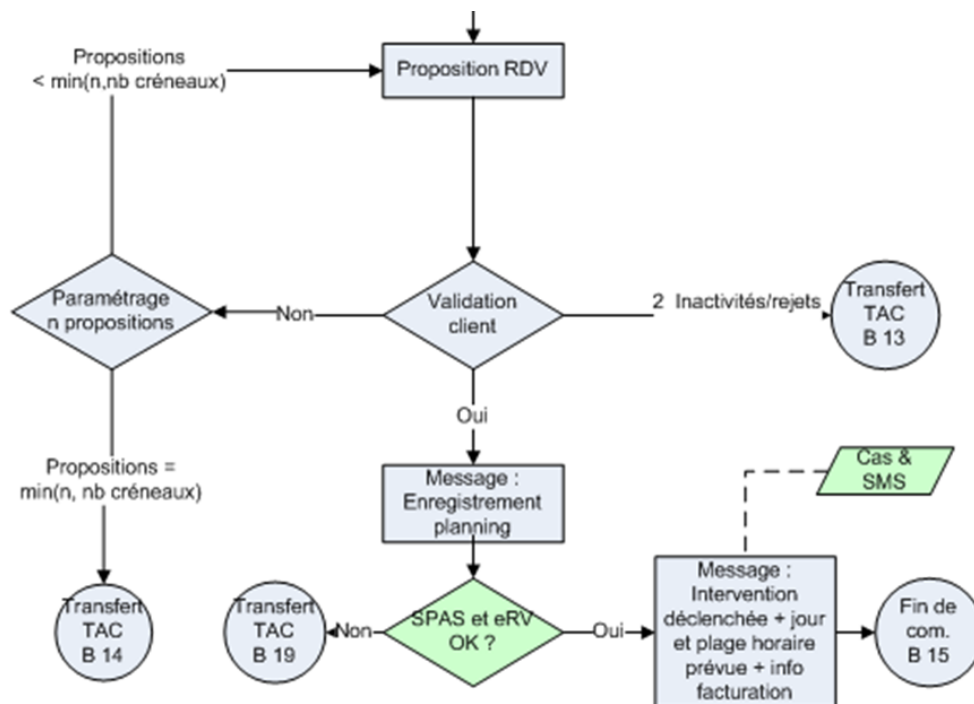


Figure 1: Basic version (original 1013+ service) of the appointment scheduling strategy.

et 12 heure Pour vous rappeler ce rendez-vous, un SMS vient de vous être envoyé. {Pause 1sec} Veuillez noter qu'en cas d'absence de votre part à ce rendez-vous, le déplacement du technicien peut vous être facturé 54,49 Euros TTC au minimum. {Pause 1sec} Pour modifier ou annuler votre rendez-vous, merci de rappeler le 1013 dans les plus brefs délais. {Pause 0,5sec} Merci de votre confiance. {Pause 3sec} {Pause 1sec} Au revoir ! {Pause 2sec}

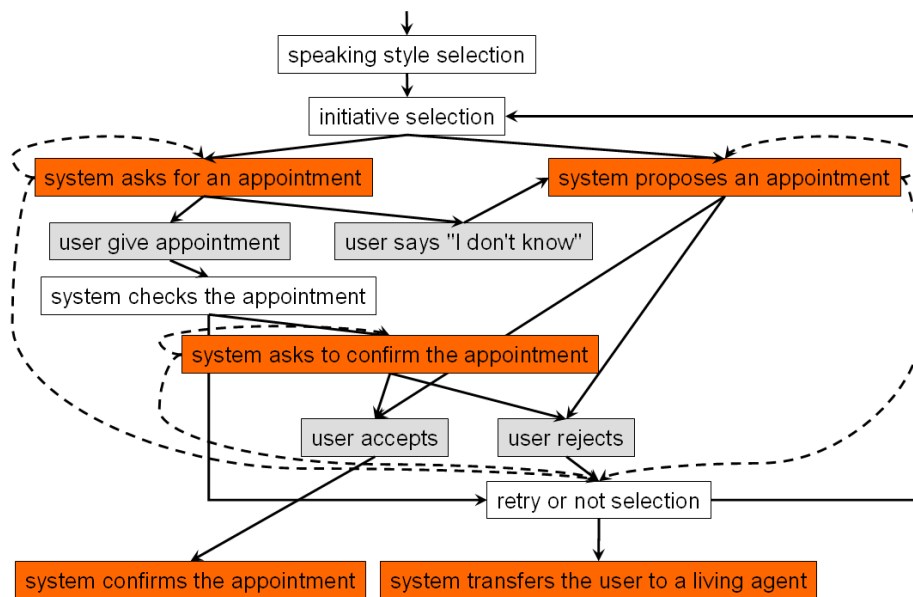


Figure 2: Dialogue strategy of System 3.

Part II

The CLASSIC systems

5 The CLASSIC public variant (System 3)

System 3 is an evolution of the basic 1013+ system introduced in Section 4 and is designed to support mixed initiative dialogue, while testing several prompt alternatives to lead to a better user understanding. Indeed, the system can ask the user to propose a date and time. The way the system chooses to retry or continue is also finer grained, and some prompt alternatives have been implemented. These prompt alternatives concern language generation from “Please tell me a week day when you would be at your home, from Monday to Saturday, in the morning or the afternoon. For instance: Tuesday morning.” to “When are you available?”. They also concern speaking style variants: personal (I), institutional (France Télécom / We), impersonal (passive mode). The full dialogue strategy is displayed in Figure 2

The system adapts online to users as determined by a reinforcement learning algorithm that chooses the best alternative at each decision point. This part is more widely explained in section 7. An example dialogue from System 3 is given below:

SYSTEM : D’après les tests effectués sur votre ligne, un rendez-vous avec un technicien est nécessaire à votre domicile. Merci de me dire un jour de la semaine

où vous seriez présents à votre domicile, du lundi au samedi, en précisant le matin ou l'après midi. Par exemple : mardi matin . C'est à vous !

USER : jeudi après-midi

SYSTEM : Très bien. Dans ce cas, je vous propose le rendez-vous suivant. aujourd'hui jeudi 29 octobre avec un début de l'intervention du technicien entre 14 heure et 16 heure Est-ce que cela vous convient ? Merci de répondre par oui ou par non.

USER : d'accord

SYSTEM : Très bien. Nous enregistrons votre rendez-vous dans notre agenda. Merci de patienter quelques instants. Votre rendez-vous avec un technicien est confirmé pour aujourd'hui jeudi 29 octobre avec un début de l'intervention entre 14 heure et 16 heure Pour vous rappeler ce rendez-vous, un SMS vient de vous être envoyé. {Pause 1sec} Veuillez noter qu'en cas d'absence de votre part à ce rendez-vous, le déplacement du technicien peut vous être facturé 54,49 Euros TTC au minimum. {Pause 1sec} Pour modifier ou annuler votre rendez-vous, merci de rappeler le 1013 dans les plus brefs délais. {Pause 0,5sec} Merci de votre confiance. {Pause 3sec} {Pause 1sec} Au revoir ! {Pause 2sec}

6 The CLASSIC lab variant (System 4)

System 4 is under development and will be the lab variant version of the 1013+ service. Its initial versions were build to explore how complex Appointment Scheduling can be by collecting a large variety of user utterances. Appendix V shows the complexity of real user dialogues. We do not address such a high level of complexity in System 4, but we have included open questions and Spoken Language Understanding of open user inputs, to test their relevance and efficiency for the appointment scheduling task against System 3.

The actual collected data are currently not far from those collected by System 3 because the extension is still preliminary. The following dialogue is an example of one collected using System 4:

SYSTEM : D'après les tests effectués sur votre ligne, un rendez-vous avec un technicien est nécessaire à votre domicile. Le premier rendez-vous disponible est le suivant : le mercredi 28 octobre Le début d'intervention du technicien aura lieu entre 10 heure et 12 heure Est-ce que cela vous convient ? Merci de répondre par oui ou par non.

USER : non

SYSTEM : Dans ce cas, merci de me dire un jour de la semaine où vous seriez présents à votre domicile, du lundi au vendredi, en précisant le matin ou l'après midi. Par exemple : mardi matin . C'est à vous !

USER : jeudi matin

SYSTEM : Très bien. Dans ce cas, je vous propose le rendez-vous suivant. aujourd'hui jeudi 29 octobre avec un début de l'intervention du technicien entre 10

heure et 12 heure Est-ce que cela vous convient ? Merci de répondre par oui ou par non.

USER : non

SYSTEM : Dans ce cas, je vous propose un autre rendez-vous. Seriez-vous disponible le mercredi 28 octobre avec un début de l'intervention du technicien entre 14 heure et 16 heure Oui ou non?

USER : non

SYSTEM : Dans ce cas, merci de me dire un jour de la semaine où vous seriez présents à votre domicile, du lundi au vendredi, en précisant le matin ou l'après midi. Par exemple : mardi matin . C'est à vous !

USER : XXX (noise)

SYSTEM : Je ne vous ai pas compris. Afin de permettre l'intervention d'un technicien sur votre ligne, merci de me dire un jour de la semaine où vous seriez présents à votre domicile, du lundi au vendredi, en précisant le matin ou l'après midi. Par exemple : mardi matin . C'est à vous !

USER : dimanche après-midi


SYSTEM : Dans ce cas, je vous propose le rendez-vous suivant Seriez-vous disponible aujourd'hui jeudi 29 octobre avec un début de l'intervention du technicien entre 14 heure et 16 heure Merci de répondre par oui ou par non.

USER : oui

SYSTEM : Très bien. Nous enregistrons votre rendez-vous dans notre agenda. Merci de patienter quelques instants. Votre rendez-vous avec un technicien est confirmé pour aujourd'hui jeudi 29 octobre avec un début de l'intervention entre 14 heure et 16 heure Pour vous rappeler ce rendez-vous, un SMS vient de vous être envoyé. {Pause 1sec} Veuillez noter qu'en cas d'absence de votre part à ce rendez-vous, le déplacement du technicien peut vous être facturé 54,49 Euros TTC au minimum. {Pause 1sec} Pour modifier ou annuler votre rendez-vous, merci de rappeler le 1013 dans les plus brefs délais. {Pause 0,5sec} Merci de votre confiance. {Pause 3sec} {Pause 1sec} Au revoir ! {Pause 2sec}

As the data collected with the initial version of System 4 turned out not to be diversified enough, we have decided to rebuild a completely new version of System 4.

The current version of System 4 is now a very modular system which aims at collecting the whole range of dialogue acts defined for appointment scheduling. As seen in Figure 3, it is composed of several chunks of dialogues which aim at having the user and the system agree on a specific appointment slot composed of an explicit day and an explicit hour:

- the system makes an explicit appointment proposition,
- the system asks for an appointment proposition,
- the system was given par  information, and asks for more details,
- the system reacts to user's inactivity,

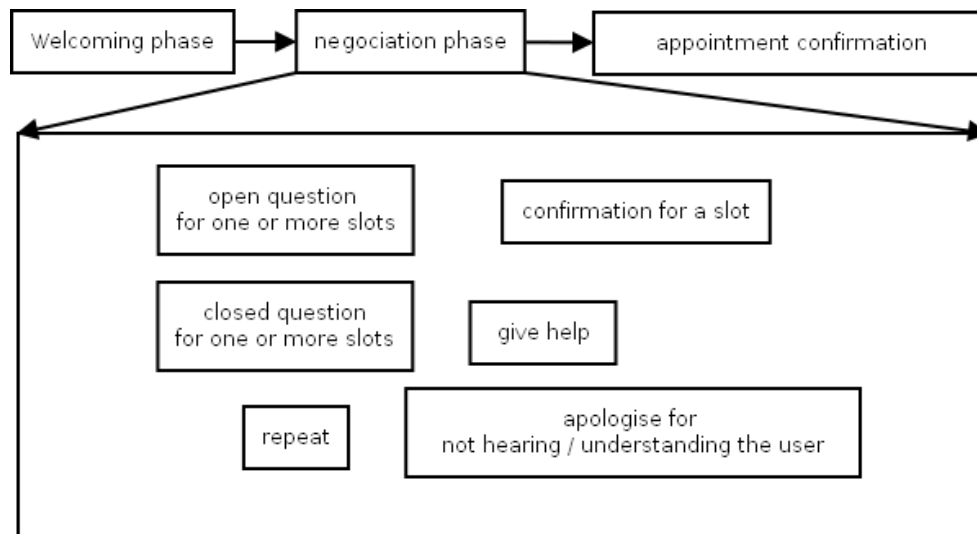


Figure 3: Dialogue strategy for System 4 data collection



- the system reacts to a lack of understanding of the user last utterance
- the system asks for a confirm act to its understood complete or partial appointment

To facilitate the data collection process and the upcoming evaluation, a website has been developed (see Figure 4). It presents the tester with a random use-case scenario to follow during the call. It also presents the tester with a questionnaire to collect subjective opinions about the dialogue. The questionnaire for now contains one assertion to which the user can agree according to a Likert scale, to ensure that she is satisfied with the appointment slot. This questionnaire will be further extended during the evaluation process. The webpage also asks the user for the dialogue ID, given at the end of the dialogue by the system, and their phone number to help compute the correlation between the system logs and the user questionnaire. Last, the users are asked to provide a pseudonym, to register the user to a lottery, used to foster more user participation. System logs can be correlated with user information.

Scénario
Consignes
 Votre connexion à internet est en panne, et le service technique a diagnostiqué qu'il y a besoin de l'intervention d'un technicien. Vous souhaitez donc prendre un rendez-vous compatible avec votre agenda dessiné ci-dessous.
Agenda n° 1

	Lundi 15.02.2010	Mardi 16.02.2010	Mercredi 17.02.2010	Jeudi 18.02.2010	Vendredi 19.02.2010	Samedi 20.02.2010	Dimanche 21.02.2010
8 ⁰⁰							
9 ⁰⁰							
10 ⁰⁰							
11 ⁰⁰							
12 ⁰⁰	Repas 12.00-13.00	Repas 12.00-13.00	Repas 12.00-13.00	Repas 12.00-13.00	Repas 12.00-13.00	Repas 12.00-13.00	Repas 12.00-13.00
13 ⁰⁰							
14 ⁰⁰							
15 ⁰⁰							
16 ⁰⁰							
17 ⁰⁰							
18 ⁰⁰							
19 ⁰⁰							

Evaluation
 Identifiant de dialogue (donné à la fin de l'appel) : Numéro d'agenda suivi :
 Le rendez-vous obtenu est-il satisfaisant ?
 pas de rendez-vous obtenu pas du tout plutôt non indifférent plutôt oui complètement
 pseudo : numéro de téléphone (pour corréler nos logs) :
 Si vous voulez ajouter des remarques ou des commentaires :

Figure 4: Companion website for System 4 data collection

7 Evaluation, rewards, learning

The Appointment Scheduling domain offers a strong task completion indicator, based on whether the system successfully schedules an appointment with the user. Indeed, the user has to confirm the appointment scheduling and this performative act serves as an online (almost) objective task completion indicator. With such a strong evaluation measure, one can cleanly study the impact of strategy variations on the overall system performance. The hybrid automata+learning based France Télécom Dialogue Management is used in System 3 and System 4 [1, 2, 3].

Their learned strategies are driven by the task completion.

Each alternative decision is evaluated with the rewards provided by the following empirical task key performance indicators: the system is given a large positive reward when the user accepts an appointment; it is given a small positive reward when the user is transferred to an operator. These rewards have been fixed in accordance with France Télécom business units.

The dialogue strategy is, therefore, learned to maximise this expected reward.

Part III

From Data Collection to Corpus Constitution

8 System 3

The 1013+ service generates exhaustive system logging, and audio files for each user interaction. Then, all the files are gathered on an operation central datasever on a daily basis. Our research team downloads the files to a dedicated analysis server, where they are parsed every morning to extract the calls containing appointment scheduling and generate a textual file containing the prompts and user interactions. These extracts are sent to a subcontractor for transcription and annotation in accordance to the agreed schema (see Section 10).

System 3 is now processing all the phonline troubleshooting calls from the entire French population. This represents around 400 dialogues per day.

9 System 4

A first version for System 4 was deployed in our lab, it was an slightly enhanced version of System 3.

We invited the France Telecom employees to call and test the service. Unfortunately, the data collected was too similar to the data of System 3 to enable a more complex dialogue system. We have therefore designed a new version of System 4 for data collection purpose, which has begun in March 2010.

10 Annotation Scheme

In this section, we summarise the proposed user utterance annotation that has been jointly constructed and agreed between the CLASSiC partners.. This is the basis of FT's annotation of user utterances, although extensions may be found to be necessary as the annotation effort proceeds. This annotation scheme allows for more compact representations than we expected to facilitate the SLU-DM interface (e.g. composite slot values). These can simply be expanded when necessary, and are included under the assumption that they simplify annotation. This annotation scheme allows for more expressive representations than we expect to allow in the SLU-DM interface (e.g. preferences). It is easier to map from a more expressive annotation to a less expressive one than vice versa.

DAs have been updated as also specified in Deliverable 5.1.2:

- `inform(a=x,b=y,...)` - provide info -

- `request(a,b=x,c=y)` - request info -
- `reqalts(a=x,b=y,...)` - request alternatives within (or without) some constraints -
- `reqmore(a=x,b=y)` - request more options -
- `confirm(a=x,b=y,...)` - request a confirmation -
- `affirm()` - yes -
- `negate()` - no -
- `hello()` - start -
- `silence()` - silence -
- `thankyou()` - "non-specific positive response from the user" -
- `ack()` - back-channel ok -
- `bye()` - bye -
- `hangup()` - hangs up -
- `repeat()` - request to repeat -
- `help()` - request for help -
- `restart()` - request to restart -
- `null()` - null act -

New DAs, or DAs with new meaning:

- `reject(a=x,b=y,...)` - wide-scope negation of all info together -
- `disprefer(a=x,b=y,...)` - preference in between `reject(...)` and `inform(...)` -

Slots and possible values:

- `time` = am,pm,0,1,...,12,13,...,23,dontcare
- `from_time` = am,pm,0,1,...,12,13,...,23,dontcare
- `to_time` = am,pm,0,1,...,12,13,...,23,dontcare
- `day` = tomorrow,dayAfterTomorrow,inThreeDays,Monday,...,Sunday,weekend,dontcare
- `from_day` = tomorrow,dayAfterTomorrow,inThreeDays,Monday,...,Sunday,weekend,dontcare

- to_day = tomorrow, dayAfterTomorrow, inThreeDays, Monday, ..., Sunday, weekend, dontcare
- week = this, next, nextnext, dontcare
- from_week = this, next, nextnext, dontcare
- to_week = this, next, nextnext, dontcare
- date = 1, ..., 31, dontcare
- from_date 1, ..., 31, dontcare
- to_date = 1, ..., 31, dontcare
- month = this, next, nextnext, jan, ..., dec, dontcare
- from_month = this, next, nextnext, jan, ..., dec, dontcare
- to_month = this, next, nextnext, jan, ..., dec, dontcare
- relative = before, after, beginning, middle, end
- reference = 1, ..., dontcare - enables ellipse resolution in the DM with a contextless annotation -
- booking = final - refers to the unique appointment booking. Used only by the system to inform that the schedule has been booked -

Lexical choice slots frame:

- slot = value[lexical choice]

Part IV

Data

11 Industrial service

Data collected during all the phases from specification to production are described in the following sections.

11.1 Specification

A corpus of 25 textual dialogues were handwritten by CLASSiC partners (5 each) in order to explore the complexity of the appointment scheduling task and to design an annotation scheme that covers most of the predicted user dialogue acts. The corpus can be found in Appendix V.

11.2 Preproduction

Around 50 dialogues have been collected during the industrial validation process. This phase has been mostly used to tune the data collection, transcription and annotation process. Examples can be found in Section 10.

11.3 Pilot and Production

Due to unforeseen circumstances, less data was collected during this phase than expected. Specifically, we had a high rate of defective phonelines in the winter of 2009/2010 due to bad weather conditions, which meant that all FT operational technicians were mobilised to repair the infrastructure. Therefore, no appointment scheduling was possible during this two month period. The users were instead served a message indicating that the network was undergoing major repairs.

With this issue resolved, around 40 dialogues are being collected each day by using System 3. At the date of 12th March, we have already collected 9923 dialogues. 966 of them contain NLG alternatives. As the appointment scheduling part of System 3 is performing well from a commercial perspective, we were given in March the whole French traffic, which contains 5 times more dialogues with NLG alternatives.

12 Lab service

Around 40 dialogues have been collected so far by the initial version of System 4. The new version of System 4 will begin collecting data commencing in March 2010.

References

- [1] R. Laroche, S. Young, O. Lemon, G. Putois, and P. Bretier. Requirements analysis and theory for statistical learning approaches in automaton-based dialogue management. Deliverable D1.1.1, CLASSIC Project, 2009.
- [2] R. Laroche, G. Putois, S. Young, J. Henderson, O. Lemon, V. Rieser, X. Liu, and P. Bretier. Final communication architecture and module interface definitions. Deliverable D5.1.2, CLASSIC Project, 2010.
- [3] R. Laroche, G. Putois, P. Bretier, and B. Bouchon-Meunier. Hybridisation of expertise and reinforcement learning in dialogue systems. In *Proceedings of the European Conference on Speech Communication and Technologies (Interspeech'09)*, 2009.

Part V

Specification Dialogues

A University of Edinburgh's example dialogues

A.1 Example EDIN1: Basic dialogue

S-en: When are you available for a visit from an engineer?

S-dm: request(day[When], time[When])

U-en: Monday afternoon.

U-dm: inform(day=Monday[Monday], time=pm[afternoon])

S-en: Sorry. There are no engineers available on Monday afternoon. How about Tuesday afternoon?

S-dm: reject(day=Monday[Monday], time=pm[afternoon]); inform(day=Tuesday[Tuesday], time=pm[afternoon])

U-en: OK, that's fine.

U-dm: affirm()

S-en: Great. I have booked that appointment for you.

S-dm: ack(); inform(booking=final[that appointment])

A.2 Example EDIN2: Smoothly switching between alternate user goals

S-en: What date would be most convenient for you?

S-dm: request(date[What date])

U-en: Either Wednesday afternoon or Friday morning.

U-dm: inform(day=Wednesday[Wednesday], time=pm[afternoon]); inform(day=Friday[Friday], time=am[morning])

S-en: Okay, the earliest appointment I have available is this Friday morning.

S-dm: ack(); inform(week=this[this], day=Friday[Friday], time=am[morning])

U-en: Friday this week is no good.

U-dm: reject(week=this[this week], day=Friday[Friday])

S-en: Okay, what about Wednesday next week.

S-dm: ack();inform(day=Wednesday[Wednesday], week=next[next week])

U-en: Yep, that's fine.

U-dm: affirm()

A.3 Example EDIN3: user provides precise time

S-en: The next available appointment is the afternoon of Tuesday the 17th?

S-dm: inform(day=Tuesday[Tuesday], date=17[17th], time=pm[afternoon])

U-en: That's no good, can you do Wednesday at 12 o'clock.

U-dm: negate(); inform(day=Wednesday[Wednesday], time=12[12 o'clock])

S-en: Unfortunately it is not possible to book exact times, appointments can only be scheduled for the morning between 8am and midday or the afternoon between 12 and 6pm. Would you like Wednesday morning or afternoon?

S-dm: help(); inform(day=Wednesday[Wednesday], time=am[morning]); inform(day=Wednesday[Wednesday]

A.4 Example EDIN4: user expresses a ranking of preferences and the DM/NLG

S-en: What date would be most convenient for you?

S-dm: request(date[What date])

U-en: It would be most convenient if you could come on Saturday. If not I might be able to get Wednesday or Thursday afternoon off work.

U-dm: inform(day=Saturday[Saturday]); disprefer(day=Wednesday[Wednesday], time=pm[afternoon]); disprefer(day=Thursday[Thursday], time=pm[afternoon])

S-en: The earliest Saturday appointment is on the morning of the 10th of May. Otherwise we can come this week on Wednesday afternoon.

S-dm: inform(day=Saturday[Saturday], date=10[10th], month=may[May], time=am[morning]); inform(day=Wednesday[Wednesday], week=this[this week], time=pm[afternoon])

U-en: The 10th is nearly two weeks away, that's no good. Okay it'll have to be Wednesday.

U-dm: reject(week=nextnext[two weeks away]); inform(day=Wednesday[Wednesday])

A.5 Example EDIN5: Alignment to temporal referring expressions, relative temporal references

S-en: When would you like to book an appointment?

S-dm: request(day[When], time[When])

U-en: Tomorrow morning.

U-dm: inform(day=tomorrow[tomorrow], time=am[morning])

S-en: Sorry, that is unavailable. How about Tomorrow evening?

S-dm: negate(); inform(day=tomorrow[tomorrow], time=pm[evening]);

U-en: No. I am busy tomorrow evening.

U-dm: negate(); reject(day=tomorrow[tomorrow], time=pm[evening])

S-en: Ok. How about the day after?

S-dm: ack(); inform(day=dayAfterTomorrow[the day after])
U-en: Great. Morning or evening?
U-dm: affirm(); request(time[Morning or evening])
S-en: Both slots are available. You could pick one that best suits you.
S-dm: inform(reference=dontcare[both slots], day=dayAfterTomorrow[both slots])
U-en: Great. Morning would be perfect.
U-dm: affirm(); inform(time=am[morning])
S-en: Ok. I am booking the Wednesday Morning slot for you.
S-dm: ack(); affirm(day=Wednesday[Wednesday], time=am[morning], booking=final)

B France Télécom's example dialogues

B.1 Example FT1: User changes his mind according to system availabilities

S-fr: Je peux vous proposer des rendez-vous du lundi matin au vendredi après-midi. Quand êtes-vous disponible ?
S-en: I could propose you appointment from Monday morning to Saturday afternoon. When are you available?
S-dm: inform(from_day=Monday[Monday], to_day=Friday[Friday]); inform(from_time=am[morning], to_time=pm[afternoon]); request(day[When], time[When])
U-fr: Comme je voyage beaucoup pour le travail, je ne suis disponible que les week-ends.
U-en: As I travel a lot for work, I'm available only during week-ends.
U-dm: inform(day=weekend[week-ends])
S-fr: Le prochain rdv disponible le samedi est en octobre, le 3 dans l'après-midi.
S-en: The next free slot on a Saturday is on October, the 3rd, during the afternoon.
S-dm: inform(day=Saturday[Saturday], date=3[3rd], month=october[October], time=pm[during the afternoon])
U-fr: Mmmhhh, J'ai besoin d'un rdv avant. Avez-vous des disponibilités lundi ?
U-en: Mmmhhh, I need an appointment before that. Do you have any availabilities on Monday?
U-dm: inform(relative=before[before]); inform(day=Monday[Monday])
S-fr: Très bien, donc rdv le lundi 21 septembre dans l'après-midi. Est-ce que vous confirmez ?
S-en: Ok, let me recap Monday the 21st of September on the afternoon. Do you confirm?
S-dm: ack(); confirm(day=Monday[Monday], date=21[21st], month=september[September], time=pm[afternoon])
U-fr: C'est parfait.
U-en: That's perfect.
U-dm: affirm()
S-fr: Très bien. J'enregistre le rendez-vous dans notre agenda.

S-en: Ok, I am recording your appointment in our agenda.

S-dm: ack(); inform(booking=final[your appointment])

B.2 Example FT2: User changes his mind during dialogue

S-fr: Je peux vous proposer des rendez-vous du lundi matin au vendredi après-midi. Quand êtes-vous disponible ?

S-en: I could propose you appointment from Monday morning to Friday afternoon. When are you available?

S-dm: inform(from_day=Monday[Monday], to_day=Friday[Friday]); inform(from_time=am[morning], to_time=pm[afternoon]); request(day[When], time[When])

U-fr: Le lundi matin.

U-en: On Monday morning.

U-dm: inform(day=Monday[Monday], time=am[morning])

S-fr: Très bien, donc rdv le lundi 21 septembre au matin. Est-ce que vous confirmez ?

S-en: Ok, let me recap Monday the 21st of September on the morning. Do you confirm?

S-dm: ack(); confirm(day=Monday[Monday], date=21,[21st] month=september[September], time=am[morning])

U-fr: Non non. Je n'avais pas réalisé que c'était le 21. Je ne suis pas disponible. Mmmh et le mercredi après-midi ?

U-en: No. No. I did not realise that it would be the 21st. I'm busy this day. Mmmh what about Wednesday afternoon?

U-dm: negate(); reject(date=21[21st]); inform(day=Wednesday[Wednesday], time=pm[afternoon])

S-fr: Très bien, donc rdv le mercredi 23 septembre dans l'après-midi. Est-ce que vous confirmez ?

S-en: Ok, let me recap Wednesday the 23rd of September on the afternoon. Do you confirm?

S-dm: ack(); confirm(day=Wednesday[Wednesday], date=23[23rd], month=september[September], time=pm[afternoon])

U-fr: C'est parfait.

U-en: That's perfect.

U-dm: affirm()

S-fr: Très bien. J'enregistre le rendez-vous dans notre agenda.

S-en: Ok, I am recording your appointment in our agenda.

S-dm: ack(); inform(booking=final[your appointment])

B.3 Example FT3: User constraints are disjunctives; System offers single slot

S-fr: Je peux vous proposer des rendez-vous du lundi matin au vendredi après-midi. Quand êtes-vous disponible ?

S-en: I could propose you an appointment from Monday morning to Friday afternoon. When are you available?

S-dm: inform(from_day=Monday[Monday], to_day=Friday[Friday]); inform(from_time=am[morning], to_time=pm[afternoon]); request(day[When], time[When])

U-fr: euh, en fin de semaine, disons jeudi ou vendredi.

U-en: mmh, at the end of the week, let's say Thursday or Friday.

U-dm: inform(week=this[at the end of the week], day=Thursday[Thursday]); inform(week=this[at the end of the week], day=Friday[Friday])

S-fr: Le premier rendez-vous disponible est le suivant: le jeudi 10 octobre. Le début d'intervention du technicien aura lieu entre 8 heures et 12 heures. Est-ce que cela vous convient ?

S-en: The first available appointment is the following: Thursday the 10th of October. The technician will come between 8am and 12pm. Does this suit you?

S-dm: confirm(day=Thursday[Thursday], date=10[10th], month=october[October], time=am[between 8am and 12pm])

U-fr: Non, c'est trop tôt

U-en: No, it's too early

U-dm: negate(); inform(relative=after[too early])

S-fr: Dans ce cas, je vous propose le rendez-vous suivant: vendredi 11 octobre de 12 heures à 16 heures. Est-ce que cela vous convient ?

S-en: In that case, I could propose you the following appointment: Friday th 11th from 12pm to 4pm. Does this suit you?

S-dm: confirm(day=Friday[Friday], date=11[11th], time=pm[from 12pm to 4pm])

U-fr: Oui

U-en: Yes

U-dm: affirm()

S-fr: Très bien. J'enregistre le rendez-vous dans notre agenda.

S-en: Ok, I am recording your appointment in our agenda.

S-dm: ack(); inform(booking=final[your appointment])

B.4 Example FT4: User constraints are disjunctives; System offers multiple slots

S-fr: Je peux vous proposer des rendez-vous du lundi matin au vendredi après-midi. Quand êtes-vous disponible ?

S-en: I could propose you appointment from Monday morning to Friday afternoon. When are you available?

S-dm: inform(from_day=Monday[Monday], to_day=Friday[Friday]); inform(from_time=am[morning], to_time=pm[afternoon]); request(day[When], time[When])

U-fr: Avez-vous des disponibilités vendredi ou lundi ?

U-en: Do you have any availabilities during either on Fridays or Mondays?

U-dm: inform(day=Friday[Friday]); inform(day=Monday[Monday])

S-fr: Le premier rdv correspondant est lundi 21 septembre dans l'après-midi, ou peut-être préférez-vous vendredi le 25 ?

S-en: Our first slot within these constraints is Monday, on September, the 21st during afternoon, or maybe you'd prefer on Friday, the 25th ?

S-dm: inform(day=Monday[Monday], month=september[September], date=21[21st], time=pm[afternoon]); inform(day=Friday[Friday], date=25[25th])

U-fr: Lundi c'est bien.

U-en: Monday is great.

U-dm: inform(day=Monday[Monday])

S-fr: Très bien, donc rdv le lundi 21 septembre dans l'après-midi. Est-ce que vous confirmez ?

S-en: Ok, let me recap Monday the 21st of September on the afternoon. Do you confirm?

S-dm: confirm(day=Monday[Monday], month=september[September], date=21[21st], time=pm[afternoon])

U-fr: C'est parfait.

U-en: That's perfect.

U-dm: affirm()

S-fr: Très bien. J'enregistre le rendez-vous dans notre agenda.

S-en: Ok, I am recording your appointment in our agenda.

S-dm: ack(); inform(booking=final[your appointment])

B.5 Example FT5: User constraints are inconsistent

S-fr: Je peux vous proposer des rendez-vous du lundi matin au vendredi après-midi. Quand êtes-vous disponible ?

S-en: I could propose you appointment from Monday morning to Saturday afternoon. When are you available?

S-dm: inform(from_day=Monday[Monday], to_day=Friday[Friday]); inform(from_time=am[morning], to_time=pm[afternoon]); request(day[When], time[When])

U-fr: Demain mercredi 31 ?

U-en: Tomorrow Wednesday the 31th?

U-dm: inform(day=tomorrow[tomorrow], day=Wednesday[Wednesday], date=31[31st])

S-fr: Voulez-vous dire demain mardi 29, le jour suivant c'est-à-dire le mercredi 30 ou le jour d'après c'est-à-dire le jeudi 31 ?

S-en: Do you mean Tomorrow Tuesday the 29th, the day after i.e. Wednesday the 30th or the day after again i.e. Thursday the 31st ?

S-dm: inform(day=tomorrow[tomorrow], day=Tuesday[Tuesday], date=29[29th]); inform(day=dayAfterTomorrow[the day after], day=Wednesday[Wednesday], date=30[30th]); inform(day=inThreeDays[the day after again], day=Thursday[Thursday], date=31[31st])

U-fr: Demain mardi 29 c'est bien.

U-en: Tomorrow Tuesday the 29th is great.

U-dm: inform(day=tomorrow[tomorrow], day=Tuesday[Tuesday], date=29[29th])

S-fr: Très bien, donc rdv le mardi 29 septembre dans l'après-midi. Est-ce que vous confirmez ?

S-en: Ok, let me recap Tuesday the 29th of September on the afternoon. Do you confirm?

S-dm: ack(); confirm(day=Tuesday[Tuesday], month=september[September], date=29[29th], time=pm[afternoon])

U-fr: C'est parfait.

U-en: That's perfect.

U-dm: affirm()

S-fr: Très bien. J'enregistre le rendez-vous dans notre agenda.

S-en: Ok, I am recording your appointment in our agenda.

S-dm: ack(); inform(booking=final[your appointment])

C University of Cambridge's example dialogues

C.1 Example UCAM1

S-en: Hi, welcome on the Orange's appointment scheduling dialogue system. When would you be available within the next two weeks?

S-dm: greet(); inform(from_week=this[within the next two weeks], to_week=nextnext[next two weeks]); request(day[When], time[When])

U-en: Thursday afternoon or Friday in the morning would be fine, but not on Monday

U-dm: inform(time=pm[afternoon], day=Thursday[Thursday]); inform(time=am[morning], day=Friday[Friday]); reject(day=Monday[Monday])

S-en: I am afraid we don't have any free slot on Friday morning. The closest free slot is on Wednesday the 16th during afternoon.

S-dm: reject(day=Friday[Friday], time=am[morning]); inform(date=16[16th], day=Wednesday[Wednesday], time=pm[afternoon])

U-en: No I can't make it on Wednesday afternoon. I can only make it in the morning.

U-dm: reject(day=Wednesday[Wednesday], time=pm[afternoon]); inform(time=am[morning])

S-en: Ok, there is another free slot on Tuesday the 22nd in the morning

S-dm: inform(date=22[22nd], day=Tuesday[Tuesday], time=am[morning])

U-en: Yeah ok but before 9 am if that's ok

U-dm: affirm(); reqmore(relative=before[before], time=9[9am])

S-en: I'm sorry, but we can only make appointments for the morning, between 9 and 12, or the afternoon, between 12 and 5

S-dm: reject(); inform(from_time=9[between 9], to_time=12[and 12]); inform(from_time=12[between 12], to_time=17[and 17])

U-en: oh ok well in the morning then

U-dm: ack(); inform(time=am[morning])

S-en: Can you confirm that you would like to book an appointment for Tuesday the 22nd in the morning?

S-dm: confirm(day=Tuesday[Tuesday], date=22[22nd], time=am[morning])

U-en: Yes

U-dm: affirm()

S-en: Your appointment has been booked for the Tuesday the 22nd, in the morning. You will receive a text confirming that. Feel free to contact us if you want to change anything. Please note that you will be charged a fee of 50 euros if you do not cancel your appointment at least 24h in advance. Can I help you with anything else?

S-dm: inform(booking=final[your appointment], date=22[22nd], day=Tuesday[Tuesday], time=am[morning]); reqmore()

U-en: No thanks

U-dm: negate()

S-en: Goodbye.

S-dm: bye()

C.2 Example UCAM2

S-en: Hi, welcome to the Orange's appointment scheduling dialogue system. When would you like to have the appointment?

S-dm: greet(); request(day[When], time[When])

U-en: Tomorrow morning if possible.

U-dm: inform(day=tomorrow[tomorrow], time=am[morning])

S-en: I am sorry but we are fully booked on Thursday 24th. The first available slot is on Tuesday 29th in the morning.

S-dm: reject(date=24[24th], day=Thursday[Thursday]); inform(day=Tuesday[Tuesday], date=29[29th], time=am[morning])

U-en: Can't do Tuesday, when is the next one?

U-dm: reject(day=Tuesday[Tuesday]); reqalts()

S-en: There is another free slot on Wednesday 2nd in the morning.

S-dm: inform(date=2[2nd], day=Wednesday[Wednesday], time=am[morning])

U-en: What about Friday?

U-dm: reqalts(day=Friday[Friday])

S-en: There is a slot available for Friday 3rd in the afternoon. There is a problem of dates in this example. As it does not matter, I did not change it

S-dm: inform(date=3[3rd], day=Friday[Friday], time=pm[afternoon])

U-en: I think I'll go for Wednesday.

U-dm: inform(day=Wednesday[Wednesday])

S-en: Can you confirm that you would like to book an appointment for Wednesday the 2nd in the morning?

S-dm: confirm(date=2[2nd], day=Wednesday[Wednesday], time=am[morning])

U-en: Is afternoon possible?

U-dm: reqalts(time=pm[afternoon])

S-en: Yes, there is a slot available on Wednesday the 2nd in the afternoon.

S-dm: affirm(); inform(day=Wednesday[Wednesday], date=2[2nd], time=pm[afternoon])

U-en: That's better.

U-dm: ack()

S-en: Can you confirm that you would like to book an appointment for Wednesday the 2nd of October in the afternoon?

S-dm: confirm(date=2[2nd], day=Wednesday[Wednesday], month=october[October], time=pm[afternoon])

U-en: Yes

U-dm: affirm()

S-en: Your appointment has been booked for the Wednesday the 2nd of October, in the afternoon. You will receive a text confirming that. Feel free to contact us if you want to change anything. Please note that you will be charged a fee of 50 euros if you do not cancel your appointment at least 24h in advance. Can I help you with anything else?

S-dm: inform(date=2[2nd], day=Wednesday[Wednesday], month=october[October], time=pm[afternoon]); reqmore()

U-en: No thanks

U-dm: negate(); thankyou()

S-en: Goodbye.

S-dm: bye()

C.3 Example UCAM3

S-en: Hi, welcome on the Orange's appointment scheduling dialogue system. The first available date is Wednesday October 7th in the afternoon. Does that suit you?

S-dm: greet(); inform(date=7[7th], month=october[October], day=Wednesday[Wednesday], time=pm[afternoon])

U-en: Wednesdays are not good for me. Thursday or Friday in the afternoon would be better.

U-dm: reject(day=Wednesday[Wednesday]); inform(day=Thursday[Thursday], time=pm[afternoon]); inform(day=Friday[Friday], time=pm[afternoon])

S-en: We have nothing available on Thursday pm or Friday pm. How does Thursday morning suit you?

S-dm: reject(day=Thursday[Thursday], time=pm[afternoon]); reject(day=Friday[Friday], time=pm[afternoon]); inform(day=Thursday[Thursday], time=am[morning])

U-en: That's no good. Perhaps later then, on Monday afternoon?

U-dm: negate(); inform(day=Monday[Monday], time=pm[afternoon])

S-en: Ok. We have a slot available on Monday October 12th between 2pm and 5pm. Do you want to book an appointment for this slot?

S-dm: ack(); inform(day=Monday[Monday], date=12[12th], month=october[October], time=pm[between 2pm and 5pm]); confirm()

U-en: Yes please.

U-dm: affirm()

S-en: Ok. A technician from Orange will call in on Monday October 12th between 2pm and 5pm. Please note that a fixed 50 euros fee will be charged on your account in case you are not at home at that time. Can I help you with anything else?

S-dm: inform(booking=final[A technician from Orange will call in], date=12[12th], day=Monday[Monday], month=october[October], time=pm[2pm and 5pm]); reqmore()

U-en: No thank you, goodbye.

U-dm: negate(); bye()

S-en: Goodbye and thank you for calling Orange's services.

S-dm: bye()

C.4 Example UCAM4

S-en: Hi, welcome to the Orange's appointment scheduling dialogue system. The first available date is Wednesday October 10 on the afternoon. Can you be at home at that time?

S-dm: greet(); inform(date=10[10], month=october[October], day=Wednesday[Wednesday], time=pm[afternoon])

U-en: uh No

U-dm: negate()

S-en: Ok. To find another possible slot you can either give directly a date or your most convenient day of the week. In either case mention morning or afternoon.

S-dm: request(date[date], day[day in the week], time[morning or afternoon])

U-en: Monday erm afternoon

U-dm: inform(day=Monday[Monday], time=pm[afternoon])

S-en: Monday October 15 in the afternoon does that suit you?

S-dm: confirm(date=15[15], day=Monday[Monday], month=october[October], time=pm[afternoon])

U-en: Perfect

U-dm: affirm()

S-en: Ok. A technician from Orange will call in on Monday October 15 between 2pm and 5pm. Please note that a 50 euros charge is due in case you are not present. Do you agree?

S-dm: inform(booking=final[A technician from Orange will call in], date=15[15], day=Monday[Monday], month=october[October], time=pm[between 2pm and 5pm]); confirm()

U-en: Yes

U-dm: affirm()

S-en: Ok. Goodbye. Thank you for calling Orange's services.

S-dm: bye(); thankyou()

C.5 Example UCAM5

S-en: Hi, welcome to the Orange's appointment scheduling dialogue system. The first available date is Wednesday October 10 on the afternoon. Can you be at home at that time?

S-dm: greet(); inform(date=10[10], month=october[October], day=Wednesday[Wednesday], time=pm[afternoon])

U-en: No

U-dm: negate()

S-en: Ok. To find another possible slot you can either give directly a date or your most convenient day of the week. In either case mention morning or afternoon.

S-dm: request(date[date], day[day in the week], time[morning or afternoon])

U-en: erm I check on the calendar erm Monday erm 15 will be good

U-dm: inform(day=Monday[Monday], date=15[15])

S-en: Will Monday October 15 morning be fine for you?

S-dm: confirm(date=15[15], day=Monday[Monday], month=october[October], time=am[morning])

U-en: uh no

U-dm: negate()

S-en: Will Monday October 15 afternoon be fine for you?

S-dm: confirm(date=15[15], day=Monday[Monday], month=october[October], time=pm[afternoon])

U-en: Yes better

U-dm: affirm()

S-en: Ok. A technician from Orange will meet you on Monday October 15 between 2pm and 5pm. Please note that a 50 euros charge is due in case you are not present. Do you agree?

S-dm: inform(booking=final[A technician from Orange will meet you], date=15[15], day=Monday[Monday], month=october[October], time=pm[between 2pm and 5pm]); confirm()

U-en: Not really pleasant but do I have the choice

U-dm: null()

S-en: Please answer yes or no

S-dm: confirm()

U-en: Ok. Yeees.

U-dm: affirm()

S-en: Ok. Goodbye. Thank you for calling Orange's services.

S-dm: bye(); thankyou()

D University of Geneva's example dialogues

D.1 Example GENE1

S-en: What date would be most convenient for you?

S-dm: greet(); request(date[what date])

U-en: Not on Tuesdays and not this Friday.

U-dm: reject(day=Tuesday[Tuesdays], week=dontcare[Tuesdays]); reject(day=Friday[Friday], week=this[this])

S-en: Wednesday morning?

S-dm: inform(day=Wednesday[Wednesday], time=am[morning])

U-en: In any case mornings are not possible.

U-dm: reject(time=am[mornings], day=dontcare[mornings])

S-en: Wednesday afternoon?

S-dm: inform(day=Wednesday[Wednesday], time=pm[afternoon])

U-en: Oh no not in the afternoon.

U-dm: reject(time=pm[afternoon])

S-en: Thursday afternoon?

S-dm: inform(day=Thursday[Thursday], time=pm[afternoon])

U-en: Yes, that's fine.

U-dm: affirm()

D.2 Example GENE2

S-en: What date would be most convenient for you?

S-dm: greet(); request(date[what date])

U-en: I don't like Mondays.

U-dm: reject(day=Monday[Mondays], week=dontcare[Mondays])

S-en: Do you prefer mornings or afternoons?

S-dm: request(time[mornings or afternoons])

U-en: Mornings are simply impossible. And I work Wednesdays.

U-dm: reject(time=am[mornings], day=dontcare[mornings]); reject(day=Wednesday[Wednesdays], week=dontcare[Wednesdays])

S-en: Okay, shall I book this Thursday afternoon for you?

S-dm: confirm(day=Thursday[Thursday], time=pm[afternoon])

U-en: No.

U-dm: negate()

S-en: The only available afternoon that is not a Wednesday or Thursday is on Monday. Shall I book this Monday afternoon for you?

S-dm: reject(time=pm[afternoon], day=Wednesday[Wednesday]); reject(time=pm[afternoon], day=Thursday[Thursday]); inform(day=Monday[Monday]); confirm(day=Monday[Monday], time=pm[afternoon], week=this[this]) override is not in the scheme.

U-en: I guess, yes.

U-dm: affirm()

D.3 Example GENE3

S-en: What date would be most convenient for you?

S-dm: greet(); request(date[what date])

U-en: I don't have much time this week, Wednesday next week would be possible.

U-dm: reject(week=this[this week]); inform(day=Wednesday[Wednesday], week=next[next week])

S-en: Sorry. That day is already taken.

S-dm: negate()

U-en: I am available Tuesday to Thursday.

U-dm: inform(from_day=Tuesday[Tuesday], to_day=Thursday[to Thursday])

S-en: How about Thursday afternoon?

S-dm: inform(day=Thursday[Thursday], time=pm[afternoon])

U-en: Yep, that's fine.

U-dm: affirm()

D.4 Example GENE4

S-en: What date would be most convenient for you?

S-dm: greet(); request(date[what date])

U-en: I usually have to work from Monday to Wednesday.

U-dm: reject(from_day=Monday[Monday], to_day=Wednesday[to Wednesday], week=dontcare[usually])

S-en: In that case we can propose Thursday in two weeks.

S-dm: inform(day=Thursday[Thursday], week=nextnext[in two weeks])

U-en: Two weeks without internet is impossible. I will take a day off.

U-dm: reject(relative=after[without], week=nextnext[Two weeks])

S-en: Is this Monday possible?

S-dm: confirm(day=Monday[Monday], week=this[this])

U-en: Yes, I will take a day off.

U-dm: affirm()

S-en: Okay, I book Monday in the afternoon for you.

S-dm: ack(); inform(booking=final[I book], day=Monday[Monday], time=pm[afternoon])

U-en: Great.

U-dm: affirm()

D.5 Example GENE5

S-en: What date would be most convenient for you?

S-dm: greet(); request(date[what date])

U-en: I am free all this week except Wednesday.

U-dm: inform(day=dontcare[all], week=this[this week]); reject(day=Wednesday[Wednesday], week=this[this week])

S-en: We have no available appointments for this week.

S-dm: reject(week=this[this week])

U-en: Next week only Friday works for me.

U-dm: reject(day=dontcare[only], week=next[next week]); inform(day=Friday[Friday], week=next[next week])

S-en: Okay, shall I book Friday in the afternoon for you?

S-dm: confirm(day=Friday[Friday], time=pm[afternoon])

U-en: Yep, that's fine.

U-dm: affirm()

E Supelec Metz's examples

E.1 Example SUPE1: Straightforward interaction

[CURRENT DAY: Tuesday 15th September]

[USR GOAL: all days free =_i week="dontmind" day="dontmind" time="dontmind" date="dontmind"]

S-en: Good morning. What can I do for you?

S-dm: greet()

U-en: I would like somebody to come.

U-dm: null()

S-en: When would you prefer us to come?

S-dm: request(day[when], time[when])

U-en: I am always free.

U-dm: inform(reference=dontcare[always free])

S-en: Could we come tomorrow at ten O'Clock?

S-dm: confirm(day=tomorrow[tomorrow], time=am[ten O'Clock])

U-en: Ok.

U-dm: affirm()

S-en: Bye.

S-dm: bye()

E.2 Example SUPE2: Straightforward goal

[CURRENT DAY: Tuesday 15th September]

[USR GOAL: next Wednesday =_i week="current" day="Wednesday" time="dontmind" date="dontmind"]

S-en: Good morning. What can I do for you?

S-dm: greet()

U-en: I would like somebody to come.

U-dm: null()

S-en: When would you prefer us to come?

S-dm: request(day[when], time[when])

U-en: Next Wednesday would be good.

U-dm: inform(day=Wednesday[Wednesday], week=next[next])

S-en: Could we come next Wednesday at ten O'Clock?

S-dm: confirm(day=Wednesday[Wednesday], week=next[next], time=am[ten O'Clock])

U-en: Ok.

U-dm: affirm()

S-en: Bye.

S-dm: bye()

E.3 Example SUPE3: Goal with simple preference

[CURRENT DAY: Tuesday 15th September]

[USR GOAL: every day but next Wednesday =_i week="dontmind" day="dontmind" time="dontmind" date="dontmind" + not(week="current", day="Wednesday")]

S-en: Good morning. What can I do for you?

S-dm: greet()

U-en: I would like somebody to come.

U-dm: null()

S-en: When would you prefer us to come?

S-dm: request(day[when], time[when])

U-en: I don't really mind.

U-dm: inform(reference=dontcare[always free])

S-en: Could we come next Wednesday at ten O'Clock?

S-dm: confirm(day=Wednesday[Wednesday], week=next[next], time=am[ten O'Clock])

U-en: Sorry, I can't.

U-dm: negate()

S-en: Could we come next Thursday at ten O'Clock?

S-dm: confirm(day=Thursday[Thursday], week=next[next], time=am[ten O'Clock])

U-en: Ok.

U-dm: affirm()

S-en: Bye.

S-dm: bye()

E.4 Example SUPE4: Goal with simple preference + alternatives from the DM

[CURRENT DAY: Tuesday 15th September]

[USR GOAL: every day, but next Wednesday, but every Monday =_ζ week="dontmind" day="dontmind" time="dontmind" date="dontmind" + not(week="current", day="Wednesday") + not(day="Monday")]

S-en: Good morning. What can I do for you?

S-dm: greet()

U-en: I would like somebody to come.

U-dm: null()

S-en: When would you prefer us to come?

S-dm: request(day[when], time[when])

U-en: I don't really mind.

U-dm: inform(reference=dontcare[always free])

S-en: Could we come tomorrow morning or Monday morning?

S-dm: inform(day=tomorrow[tomorrow], time=am[morning]); inform(day=Monday[Monday], time=am[morning])

U-en: No, I can't.

U-dm: negate()

S-en: Could we come next Monday at two O'Clock?

S-dm: confirm(day=Monday[Monday], week=next[next], time=pm[two O'Clock])

U-en: Sorry, I can't. Mondays are impossible.

U-dm: negate(); reject(day=Monday[Mondays], week=dontcare[Mondays])

S-en: Could we come next Thursday at ten O'Clock, or next Tuesday at two O'Clock?

S-dm: inform(day=Thursday[Thursday], time=am[ten O'Clock], week=next[next]); inform(day=Tuesday[Tuesday], time=pm[two O'Clock], week=next[next])

U-en: Next Tuesday at two O'Clock is ok!

U-dm: inform(day=Tuesday[Tuesday], week=next[next], time=pm[two O'Clock])

S-en: Is next Tuesday at two O'Clock ok?

S-dm: confirm(day=Tuesday[Tuesday], week=next[next], time=pm[two O'Clock])

U-en: Yes!

U-dm: affirm()

S-en: Bye.

S-dm: bye()

E.5 Example SUPE5: Mistake in the initial goal

[CURRENT DAY: Tuesday 15th September]

[USR GOAL: mistake in the goal =_i week="next" day="Monday" time="dontmind" date="22"
becomes: week="next" day="Tuesday" time="dontmind" date="22"]

S-en: Good morning. What can I do for you?

S-dm: greet()

U-en: I would like somebody to come.

U-dm: null()

S-en: When would you prefer us to come?

S-dm: request(day[when], time[when])

U-en: Monday the 22nd would be ok.

U-dm: inform(day=Monday[Monday], date=22[22nd])

S-en: Monday the 22nd doesn't exist.

S-dm: reject(day=Monday[Monday], date=22[22nd])

U-en: Sorry. I meant Tuesday the 22nd.

U-dm: inform(day=Tuesday[Tuesday], date=22[22nd])

S-en: Tuesday morning or Tuesday afternoon?

S-dm: request(time[morning or afternoon], day=Tuesday[Tuesday])

U-en: Afternoon, please.

U-dm: inform(time=pm[afternoon])

S-en: Bye.

S-dm: bye()