

CLASSiC

D5.3.2: Final France Telecom industrial platform adapted to CLASSiC architecture (“System 3”)

Ghislain Putois, Philippe Bretier, Romain Laroche

Distribution: Consortium

CLASSiC

Computational Learning in Adaptive Systems for Spoken Conversation
216594 Deliverable 5.3.2

May 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	M27 = May 2010
Actual date of delivery	May 2010
Deliverable number	5.3.2
Deliverable title	D5.3.2: Final France Telecom industrial platform adapted to CLASSiC architecture ("System 3")
Type	Prototype
Status & version	final 1.0
Number of pages	5 (excluding front matter)
Contributing WP	5
WP/Task responsible	WP5, leader France Telecom
Other contributors	Oliver Lemon, Verena Rieser
Author(s)	Ghislain Putois, Philippe Bretier, Romain Laroche
EC Project Officer	Philippe Gelin
Keywords	Platform, System, user feedbacks, logging, design alternatives, Dialogue Management

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>

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
1 Introduction	2
2 Platform evolutions	2
2.1 Learning Manager	2
2.2 System-based decisions : dialogue management alternatives	2
2.3 Logging	3
2.4 Use and Reward feedback	3
3 Software	4

Executive Summary

This document is a short report to accompany the Prototype Deliverable 5.3.2, due at month 27 of the CLASSIC project. It describes the adaptations made to the industrial platform to conform to the specified CLASSIC architecture. The result is known as System 3. The main evolutions made are the introduction of system-based decisions in the dialogue automata design, an exhaustive logging, and the ability to handle use feedback and rewards.

1 Introduction

This document describes the changes made to the industrial platform to conform to the CLASSiC architecture as described in deliverable D5.1.2. The resulting system is known as System 3. The CLASSiC project intends to learn better dialogue behaviours over time. System 3 is based on the existing automata-based industrial solution: it enables an easy design of dialogue alternatives, and offers the capabilities to study their impact at runtime by archiving platform logs and aggregating them back, thus allowing to use them as feedback for learning mechanisms. The result of logging and feedback can be visualised in the Dialogue Design Studio. In the following sections, a brief overview of the evolutions made to the platform will be given.

2 Platform evolutions

The industrial platform has been evolved in three main directions: introducing system-based decision taking, completing system logging to be exhaustive to enable learning decision processes and adding use and reward feedback management. These three directions are used in synergy inside a new module, the Learning Manager.

2.1 Learning Manager

The Learning Manager module has been introduced in the latest evolution of the System 3 to handle the context information, and store all the previous dialogue performances. It is given a description of all the dialogue alternatives, and gathers the feedbacks of all past dialogues to act as an oracle to help the Dialogue Manager module selects the best dialogue alternative for each new dialogue.

2.2 System-based decisions : dialogue management alternatives

The Dialogue Design Studio has first been extended to allow the design of dialogue alternatives. Then, the runtime platform has been extended to allow the system to choose at runtime one path between the pre-designed alternatives.

Figure 1 illustrates this mechanism. First, the global variable *nomPDC* is filled with the name of the current point of choice. The Learning Manager component is then called. It computes the list of possible alternatives available for this point of choice according to the current context, and stores its proposed alternative in the form of an integer in the global variable *choixPDC*. Then, the switch case branches according to this value and plays the appropriate prompt. For instance, when *choixPDC* equals 0, the prompt “Merci de me dire un jour de la semaine où vous seriez présents à votre domicile du lundi au vendredi.” (Please tell me a day during the week when you would be at home, between Monday and Friday) is played. In this case, every branch is merged back into the main dialogue flow and the dialogue continues by logging the last events.

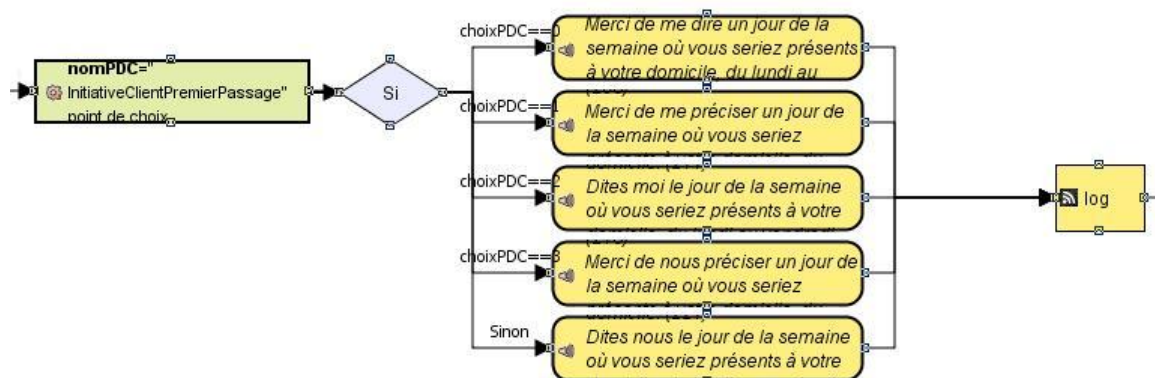


Figure 1: Point of choice mechanism

2.3 Logging

During the course of the project, the platform has undergone a major evolution to handle an exhaustive logging of the server activities throughout the dialogue session. In particular, it logs all current automata states, global variable states, used prompt alternatives, timing information, user utterances, recognition information, and back-end communications.

A dedicated system has been created to aggregate and manage all the logs: the Dialogue Database. A screenshot of this tool is presented in figure 2. The figure shows an excerpt of a captured dialogue session. One can see the details on the prompt played “D’après les tests effectués sur votre ligne, un rendez-vous avec un technicien est nécessaire à votre domicile” (According to the tests done on your landline, you need an appointment with an operator at your place.), as well as the associated recorded file (/1013son-G4/101.wav), so that one can listen to the prosody and speaking style used. One can also see the associated timestamps in the left column.

2.4 Use and Reward feedback

The system logs stored in the Dialogue Database can also be imported back straight in the Dialogue Design Studio, where they will be used to compute dialogue indicators: which are the most frequent dialogue alternatives used, when the customer hung-up, how many times a system prompt was cut before it ends by the client (barge-in). All these indicators are very important feedbacks for the designer, which can now easily measure the effects of her design choices. In figure 3, one can see small percentages under each branching point indicating the rate of traffic which was routed through it. In this session, we have focused on a recognition and NLU step : in 94% of the cases, the recogniser have heard the user ; the NLU has interpreted the recognised words as Repeat in 6% of the cases, as Yes in 43%, as No in 18%, and as Not Understood in 31%. Note that this was a debug session, and the “success” rate for the final service are really higher. In the far right of the figure, one can also see the total percentages each exit represents. Those percentages are relative to the beginning of this dialogue turn, and are smaller than the interpretation numbers, because some users have just hung up before the en of the dialogue turn.

Info	dialogObjects	value
file	/1013son-G4/101.wav	
TTS	D'après les tests effectués sur votre ligne, un rendez-vous avec un technicien est nécessaire à votre domicile.	
params	timeout	4s
	bargain	False
file	/1013son-G4/112.wav	
TTS	Le premier rendez-vous disponible est le suivant :	
params	timeout	4s
	bargain	False
TTS	le	
params	timeout	4s
	bargain	False
TTS	vendredi	
params	timeout	4s
	bargain	False
TTS	18	
params	timeout	4s
	bargain	False
TTS	decembre	
params	timeout	4s
	bargain	False
file	/1013son-G4/113.wav	
TTS	Le début d'intervention du technicien aura lieu entre	
params	timeout	4s
	bargain	False
TTS	10 heure	
params	timeout	4s
	bargain	False
TTS	30	
params	timeout	4s
	bargain	False
file	/1013son-G4/104.wav	
TTS	et	
params	timeout	4s
	bargain	False
TTS	12 heure	
params	timeout	4s
	bargain	False
TTS	30	
params	timeout	4s
	bargain	False
file	/1013son-G4/114.wav	
TTS	Est-ce que cela vous convient ?	
params	timeout	4s
	bargain	False
file	/1013son-G4/115.wav	

Figure 2: Dialogue logging database view

Other information is also available and given in a “Statistiques” (Statistics) tab on the bottom of the figure. Here, we can visualise the mean duration of the prompt, if there was hang-ups during the prompt, and how often the barge-in mechanism was used (14%), and how long after the beginning of the prompt it was used (0.34s).

The platform has also been extended to allow the logging of designed service specific information. The designer can include in the Dialogue Design Studio a set of global variables operations, which are useful to implement service specific rewards. Figure 4 illustrates an implementation of such a service-dependent reward. It represents the processing of a recognition error when the system asks the user to confirm a timeslot. An API of the Learning Manager is called to record a negative reward in the action block “punitionErreurReco”. Then, according to the content of the global variable *estInactivite*, a prompt is played to inform the user that the system either has not heard or understood her. The actions are then logged.

3 Software

The dialogue platform is based on the industrial Disserto Suite. The Disserto Suite is based on JAVA for the design tools, and J2EE for the runtime part. The CLASSiC project has led to the extension of the following components: the Dialogue Manager runtime, and the Dialogue Design Studio. It has also led to the development of new components: Dialogue Database, and Dialogue Design Studio Log Importer. The sources are not made publicly available, as they are part of the France Telecom commercial solution.

System 3 has opened the opportunity to introduce reinforcement learning mechanisms in the proof-of-concept Learning Manager module, which should be extended and industrialised outside of the CLASSiC project.

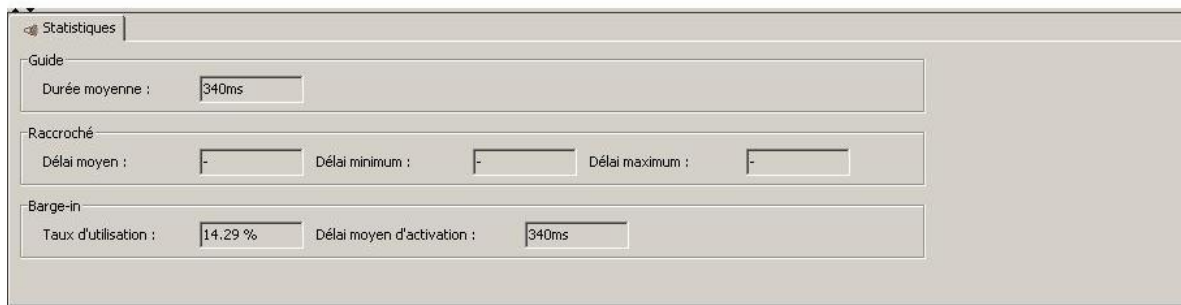
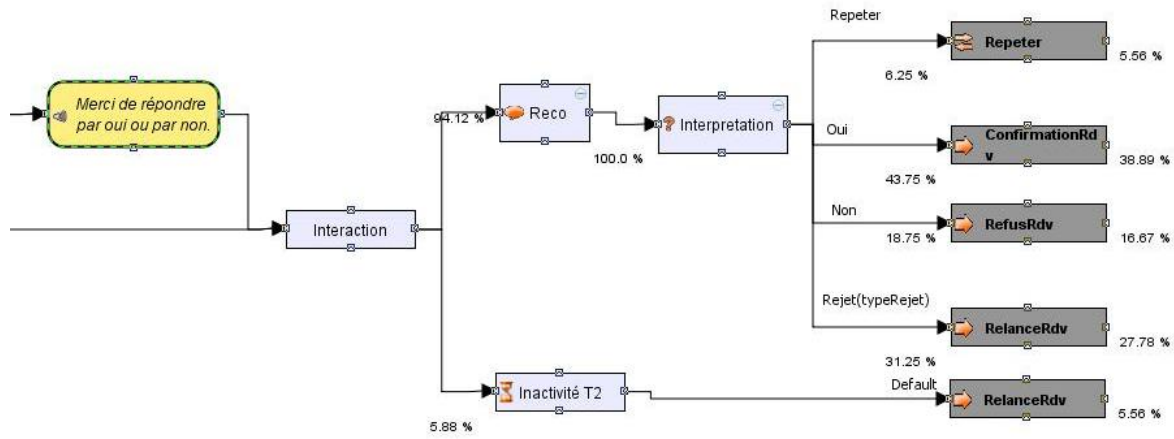


Figure 3: Feedback in the Dialogue Design Studio

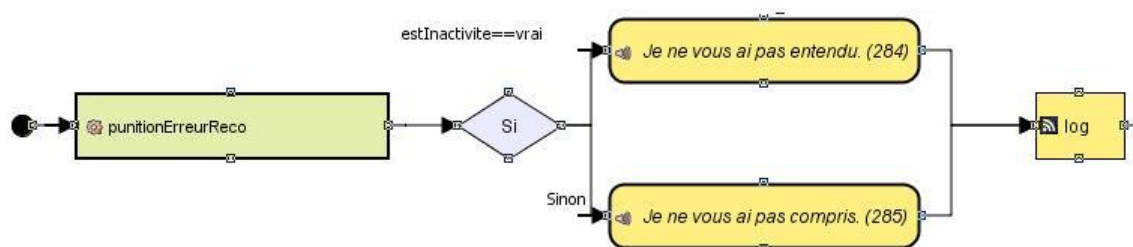


Figure 4: Service Reward Design