Ad-hoc evaluations along the lifecycle of industrial spoken dialogue systems: heading to harmonisation?

Marianne Laurent, Philippe Bretier, Carole Manquillet

Orange Labs
2 avenue Pierre Marzin, 22307 Lannion, France
marianne.laurent@orange-ftgroup.com, philippe.bretier@orange-ftgroup.com

Abstract

With a view to rationalise the evaluation process within the Orange Labs *spoken dialogue system* projects, a field audit has been realised among the various related professionals. The article presents the main conclusions of the study and draws work perspectives to enhance the evaluation process in such a complex organisation. We first present the typical spoken dialogue system project lifecycle and the involved communities of stakeholders. We then sketch a map of indicators used across the teams. It shows that each professional category designs its evaluation metrics according to a case-by-case strategy, each one targeting different goals and methodologies. And last, we identify weaknesses in the evaluation process is handled by the various teams. Among others, we mention: the dependency on the design and exploitation tools that may not be suitable for an adequate collection of relevant indicators, the need to refine some indicators' definition and analysis to obtain valuable information for system enhancement, the sharing issue that advocates for a common definition of indicators across the teams and, as a consequence, the need for shared applications that support and encourage such a rationalisation.

1. Introduction

The field of SDS has now been roused by the evaluation issue for a dozen years. Both industry and academic research groups have been looking for a standardized evaluation process, yet handling the challenge differently (Paek, 2007). On the one hand, researchers need benchmark solutions to evaluate and validate the results to be communicated to the community, either for overall systems or new components to be tested. On the other hand, industry strongly focuses on the projects' Return on Investment (ROI) which is a profit versus costs analysis. Yet, evaluation must also consider the application design relevance for the end-users. Consequently, in addition to the financial evaluation, a SDS is also evaluated through field tests involving both quantitative analysis of logged interactions with test users and qualitative ones based on the users perception.

Contrary to the academic field where, for a SDS to be developed, the entire project lifecycle is generally handled by the same group of researchers, in industry, each SDS development stage involves several professional fields, including engineers, ergonomics experts, marketers and decision makers. These groups are dedicated to different roles in the lifecycle. They consequently pursue different goals, have different evaluation needs and therefore select different sets of indicators. Altogether, this is about a hundred of them monitored alongside the major projects.

With a willing to industrialise the evaluation process in the SDS development process, we carried out an internal audit to inventory evaluation approaches of the various SDS-related professional groups. It aimed at providing ground material to work on the rationalisation of evaluation processes and its taking into account in a data-driven SDS development approach. An ergonomics expert processed interviews among the various communities involved in SDS lifecycles. She collected the interviewees' perception on their role, objective, involvement in the project lifecycle, interactions with other groups' members and their evaluation practice. This enabled to model the framework of intervention of the various teams and the indicators' context of use. It also helped identifying both best practices and opportunities for enhancement with a view to generalise them across the teams and to rationalise the processes.

The study provided two main results: The first deliverable is a map of the various teams' contributions to SDS projects. It covers their context of intervention, their interaction with other teams and their working methods. The sections 2. and 3. detail this prerequisite to understand the various possible uses and interpretations of indicators for SDS evaluation. The second one is a cartography of indicators involved during the SDSs conception and evaluation (see section 4.). Each indicator is detailed by its calculation method, the involved teams, the relative evaluated part of system (ex: application, service, caller experience) and the type of observed measure (ex: ROI, Service Level Agreements (SLA), supervision, Quality of Service (QoS)). The final objective is to improve the conception and evaluation processes by both (i) enriching the evaluation and monitoring tools and (ii) increasing the automation in evaluation, starting at the conception phase. The section 5. presents the issues for such a rationalisation and, the section 6., our consequent work in progress.

2. The typical SDS project lifecycle

Most of our SDS projects are structured around three stages (see Fig 1): a study period, a realisation and adjustment phase and the ongoing production phase, each of them potentially implying iterative loops. The transitions from a phase to another are identified by go/no-go milestones, more or less formally planned at the end of each phase by the decision makers.

The study period It starts with an analysis of the end-users' needs and practices and a solutions benchmark. This enables to define the project scope, distinguishing mandatory features and the "nice-to-haves", write the specifica-

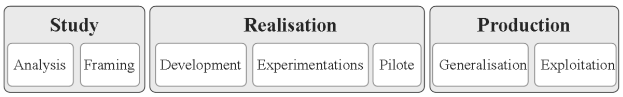

Figure 1: The typical project lifecycle.

tions and anticipate the project plan.

The realisation and adjustment phase This phase starts with the development of a prototype integrating the initial design, linguistic, usability and technical considerations. This beta-solution is then adjusted through iterative development loops, so as to gradually tailor the solution to the targeted live conditions. We may distinguish three steps of development and adjustment: the development itself, the experimentation and the pilot phases.

The *development phase* begins with the development of a first version of the service. Experts set up the initial automatic speech recognition (ASR) models, natural language understanding (NLU) models and callflow. Then, by testing the solution inside the development team, they gradually enrich the latter components. Nevertheless, these tests do involve neither naive users nor the final platform architecture. Consequently, the evaluation is limited the technical implementation accuracy.

Subsequently, the *experimentation phase* grants the opportunity to test the service on a restricted panel of recruited users. The collected corpus, gathering both interaction logs and users feedbacks, helps the experts in apprehending the users behaviour with the system. With this material, they adjust the various design parameters, including the ASR and NLU models, the syntax and the prompts terminology. This way, it feeds the iteration cycles until a satisfying level of performance is reached. At this stage, the evaluation focuses on the application enhancement and the alignment of observations with *a priori* projections. Operational constraints (ex: performance, traffic) are not addressed before the next stage.

Last, *the pilot phase* consists in testing the application under real conditions with a fragment of the real flow of traffic. It allows testing the complete technical architecture so as to detect and fix the last beta-solution's design issues. Hence, this constitutes a comprehensive validation of the service before its access is extended to all users.

The deployment phase It begins with granting access to the entire flow of traffic. Then follows the exploitation, which consists in the service hosting and maintenance. This phase permits to both observe the users behaviour in live conditions and collect their feedbacks. The service keeps constantly evaluated, monitored and supported according to a three-level escalation process. The first level involves the helpdesk for simple issues. Then, the service hosting providers cope with both end-users issues and issues detected by automatic monitoring tools. Finally, the service developers handle the escalated issues requiring a code revision or elaborated parameterisation. Hence, the deployment is not a cut-off point in the project since this ongoing phase reveals enhancement opportunities. The service is then tailored according to the users practices, changes in the service scope or the competitive environment.

Three approaches of evaluation In this lifecycle three evaluation strategies follow one another. First, in the development phase, the evaluation focuses on *spotting the design errors* to be repaired. Then, if experimentation and pilot phases' evaluations include error spotting too, they also aim at assessing both the performance variation between two iterative development loops and the objectives achievement for go/no go decision before service going live. Last, the deployment phase is monitored with very high level indicators addressed to various stakeholders into daily dashboards. In case of an indicator *going red*, they may look for more precise indicators to trace the causes.

3. Three groups of stakeholders

3.1. Varied points of view

Industrial SDS projects involve many *communities of practice*. Below, we propose the various SDS definitions as perceived by the main groups of stakeholders.

- The *contracting owners* see SDSs as hard/software platforms delivering a service the customers.
- The *"métier" people* consider SDS as an automate routing service satisfying the customer relationship management and the service policy.
- *Technical developers and hosting service providers* picture them as hard and software platforms.
- *Ergonomists* paint SDS as interactive systems implying a human-machine dialogue, a quality of service (QoS) and an individualised access to information according to the end-user needs and expectations.
- *Marketing people* mention a service based on specific technologies and sold to a business customer to qualify their incoming calls and to provide their end-users with the required pieces of information.
- *Business Managers* refer to a service enabling to optimise organisation of human operators teams by automating the repetitive or low-added value tasks, while adequately orienting the customer toward human skills or self-service applications.

Each of these communities reveals a specific point of view, follows specific activities and thus has different evaluation expectations. To facilitate their characterisation, we regroup them into three categories: the ordering parties, the *métier* and the project owners (see table 1).

Customers	Providers
- Ordering parties : decision makers	Project Owners groups:
- *Métier*: definition of the customer's specificities	- Marketing - Business manager - Technical experts - Ergonomics experts - Hosting providers

Table 1: Identified teams of stakeholders

3.2. The ordering parties

In SDS projects, the ordering parties aim at enhancing the customer service with automated vocal solutions that respect the QoS standards. First, they are in charge of the global project management. This includes planning, prioritisation, validation, go/no go decisions and reporting. Second, in line with "métiers" and marketing, they provide project owners with the technical and functional specifications. Hence, they have to manage the customers' expectations and the QoS standards as well as the strategic, legal and financial stakes. Third, in parallel to the system development, they organise the change management (ex: operators' training and the retail network development).
The ordering parties' evaluation mostly consists in measuring the service performance and quality according to predefined goals. It requires to assessing the respect of commitments, the efficiency of the NLU+ASR and the users satisfaction. Moreover, for project launch and financing decisions they use ROI indicators; for service going live and replacements, they evaluate the quality of the provided service according pre-agreed terms of reference.

3.3. "Métiers"

"*Métiers*" regroups the employees working in the business lines targeted by the SDS implementation. Their role is to deliver a service to the customer with respect to the customer relationship policy.
They are involved in the service functional definition and its *a posteriori* validation. Therefore, they verify its realisation, its economical and organisational relevancy and the performance of the customer service. To that purpose, they both define the monitoring metrics and tailor the service parameterisation so as to make the adjustments easier for non-technical stakeholders. Monitoring both the system performance and the service quality according to the end-users experiences, they gather requirements for future service evolutions.
A SDS may either replace a human operator (for after-sale customer support for example) or unburden operators by automatically routing the incoming calls to the accurate service. Hence, the relative teams need qualitative and quantitative analysis on the human-machine interaction to analyse the end-users experiences within the service. They need to check to what extent the SDS is supporting their activity. Their evaluation consequently covers: the user perception, the dialogue quality, the calls motives repartition, the user exit, indicators linked to the service.

3.4. Project owners

Three groups of project owners share the service design, development and enhancement activities.

The marketing people and business managers. Responsible of indentifying the users' needs, they write the commercial offers. Then, they make sure the deployed services answer both the customers and the business needs, for which they need a measure of the end-users' perceptions and the service performance.

The development teams. They deal with the proper development of the service along with three points of view. First, the *technical experts* develop the service, control its functioning and performance, monitor and correct the escalated anomalies. Second, the *vocal experts* are responsible for the dialogue specifications (ASR and NLU models and callflow design). Upstream, they provide the other groups of experts with an expertise on dialogue management; downstream, they evaluate the dialogue quality when the SDS is confronted to real users. And third, the *ergonomists* deploy a user-centric approach. So as to monitor the service perceived quality (callflow adequacy, usability, acceptability, reliability, etc.), they manage expert evaluation campaigns with both field tests and real users. They deploy observation, analysis and decision tools to identify and understand how to fix the problems caused by either system weaknesses or unexpected user behaviour. They are also consulted during the design phase.

The hosting services providers. Responsible of the service hosting and maintenance, they need tools to both monitor the good functioning of the live services they host and escalate the identified anomalies.
On that account, we notice that evaluation is threefold along the SDS projects lifecycle. It relies upon: (i) a measure of the service performance, (ii) a tool for observation, analysis and decision support and (iii) a monitoring of the hosting and exploitation services.

3.5. Cross interactions

Each evaluation is conditioned by the relative stakeholders' specific activities, goals, practices and system of values. Yet, these groups of actors, because of their complementary in the project value chain, need to interact with each other. We can therefore anticipate potential misunderstandings in the way evaluation reports may be interpreted across the teams. To clarify each group's point of view as regard evaluation tools, the next section describes the main goals and use of evaluation indicators across the teams.

4. Mapping Indicators

4.1. Definition and functions

We define an *indicator* as the qualification or quantification of a feature, measured so as to evaluate to what extent a given result is achieved. Such a variable is calculated from a range of parameters and positioned on a value scale that may describe a phenomenon and assess its relative change in the time. Our field study enabled to identify the majority of indicators handled across the teams. We detailed them according to the "5Ws":

What: What is the relative studied SDS?
Who: Which team is involved?
Where: Which level is targeted? (ex: dialogue, phase)
When: Which phase of the lifecycle is concerned?
How: What are the resources, the calculation, etc.?
Why: Which criteria is assessed? (ex: ROI, SLA, QoS)
How much: On what frequency is the indicator observed?

The next section describes the different uses of indicators and a map of indicators we sketched out of this corpus.

4.2. Different use by group of stakeholders

An indicator may be used and interpreted differently according to a given evaluator's *community of practice* and situation of evaluation. Indeed, just as the community influences its practice, interests and system of values, the situation conditions its evaluation goals. The examples presented in the tables 2 and 3 illustrate how an *a priori* unique indicator may lead to different interpretations according to its audience. Then, we present the three major approaches as regard the use of indicators.

Ordering parties	Monitor the service performance.
Mtier	Observe the QoS perceived by users.
Project owner	Analyse the system performance.

Table 2: Time to identify the call motive.

Ergonomics	Analyse the users behaviour and identify causes of dialogue failure.
SLU	Assess vocal recognition performance.
Technical	Analyse the platform performance.

Table 3: Ratio of unrecognised user utterances.

Ordering parties They check if the system is functioning in accordance to the predefined expectations. Their selection of indicators includes both: qualitative ones for the assessment of the compliance to the users expectations and performance ones to check the global flow of calls on the dedicated platforms. They mostly examine indicators linked to (i) their providers commitments (ex: unavailability ratio) and (ii) their customers SLAs' commitment.

Métier They need precise data on the user experience (ex: waiting time, resignation) and its options for task achievement (ex: path in the callflow, employed functionalities). This technical and usability analysis enables to anticipate the future design evolutions.

Project owners Monitoring indicators allows them to identify the problems linked to both design issues and unexpected users behaviour.

4.3. Five evaluation points of view

An indicator strongly depends on the interpretation and the use that will be made out of it. Quoted out of its context, an indicator is only data. It only becomes relevant information when considered in its context of interpretation. Therefore building a set of indicators supposed to fit the needs of every group of stakeholders is risky. Actually, even if it might gather common information for analysis, each group of actors will interpret them differently according to their very needs and local interests in the project.

Nonetheless, a single stakeholder may eventually take a range of several points of view into account, depending, for example, on its career path. This may help him to interact with other stakeholders, whether they are from different communities of practice or hierarchical levels.

These different uses may be defined into five points of interest for SDS evaluation. We listed them in the table 4, with the relative evaluation criteria and examples of possible indicators.

User experience
- User "reception": *ingoing & outgoing calls, duration* - Dialogue phase: *phase duration* - User leaving: *hangover inside a phase, transfer* - User path: *path into the callflow*
The service
- Call motive qualification: *call motives repartition* - Treatment of unrecognised call motive: *unrecognised motive, out of scope* - User request treatment: *treatment either automatic or by a human operator, volume by operator platform* - End-user/SDS interaction: *used functionalities, used operations and commands*
SDS-user interaction
- User experience: *perceived satisfaction, efficiency, conviviality, simplicity, etc. (voice, service, etc.)* - Generated dialogue: *mean duration, number of turns* - Reco/SLU performance: *Vocal recognition error ratio, number of out-of-vocabulary words* - Failed communications: *transfer by default, technical failure, dialogue failure* - Broadcast: *help messages, silences, misunderstanding*
Platform interactions
- Access to the service: *waiting time before the system takes the call, number of refused access* - Callflows: *number of calls, maximum number, duration* - Routing: *incoming routed calls, hangover while routing* - Transfer to human operator: *number of calls transferred*
Technical performance
- Monitoring of breakdown and incidents: *number of escalated incidents, incident gravity, unavailability* - Server supervision: *CPU load, disc resource* - Scalability test: *time to answer minimum-maximum-average, available RAM*

Table 4: Five evaluation levels for a taxonomy of indicators

We identify an evaluation panorama composed of five perspectives involving both, very high level indicators addressing the notion of service and customer relation and indicators describing the technical functioning of the system.

4.4. Factors for a categorization

We organised the corpus of indicators around three axes.

The indicator's level of interest within the service. On the one hand, a macroscopic level gathers indicators that monitor the functioning of the service. They alert the stakeholders when a problem is identified (ex: a strong decrease in the number of calls). Most of these metrics can be automatically collected from the interaction logs. Yet, however they may alert the experts, they cannot inform on the cause of the failure. The microscopic level, on the other hand, regroups indicators enabling more detailed analysis that may inform on the problem origin. For example, we use an indicator of optimum conversation that analyses the user path within the callflow so as to precisely locate the potential difficulties a user may face.

The metric's degree of specificity. We collapse it into three classes: a metric can be very general and applicable to any type of studied SDS, linked to the type of SDS (self-care or call routing) or specific to a service, i.e. the domain.

The collection/calculation method. First, some parameters can be extracted automatically from the logs. However, the selection of parameters and their aggregation into a relevant and consistent evaluation indicator needs to be defined. The automatically collected indicators are favoured across the teams, not necessarily because of their relevancy to the evaluation needs, but because of their low average cost and ease of collection. Second, parameters may need previous human handling, such as manual transcription and annotation. And third, various parameters cannot be automatically extracted from the logs with the design and monitoring tools implemented. Anticipation at the very first stages of future SDS developments might enable to automate their collection. As mentioned in 5.2., this leads to some severe shortages with respect to the evaluation needs. For example, when analysing problematic logged calls, our designers do not have precise information on the way a given communication may start and end. Actually, on the one hand, the calls' motives and sub-motives are barely recognised. On the other hand, they do not have precise information on how an interaction finishes (ex: caller hang-up, SDS failure, routing decision) and when precisely the caller may hang up (before, during or after the caller hears a specific message). Such details would be of a great help to identify potential callflow design issues, which could, consequently, help experts improve the global service quality.

5. Identified perspectives of enhancement

This section reports perspectives of enhancement we identified with a view to rationalise the evaluation process.

5.1. Oral dialogue indicators, the interpretation issue

A vocal human-machine dialogue is supported by both the identification of the callers' intentions and the consequent sequence of dialogue turns to handle the recognised intended task. We may therefore distinguish indicators linked to the vocal recognition and natural language performance from the ones monitoring the dialogue quality.
Handling indicators for ASR and NLU is a big issue. Identifying if the user intention has been correctly identified and the relative correctly processed and routed is a difficult task since this involves the end-user point of view. The study of such metrics requires the analysis of the recorded corpus of interaction. Third-party annotators compare the actual user request with the system's interpretation and consequent behaviour. But these evaluations are costly and time-consuming. Moreover, such a human analysis, *i.e.* third-party identification of user intentions, is prone to misinterpretation. An automatic calculation of these indicators is actually a major but open issue.

5.2. Dependency on deployed technologies and need for a refinement of indicators

The automatic collection of metrics strongly depends on the technical platform that supports the service. Actually, a platform may deliver interaction log files including, natively, a set of predefined indicators that depends on the upstream agreements for parameters collection. This may lead to a strong heterogeneity in disposable parameters for each SDS to be evaluated. However, as new projects are launched, a harmonisation of architecture and design tools is to be expected. This should encourage and make easier a normalised approach to evaluation. Anyway, this underlines the need to define the indicators for the future evaluations as early as possible in the SDS design process.
When evaluations, and relative needed parameters, have not been defined upstream, evaluations are built from the corpus of parameters available in logs. Nevertheless, as illustrated below with examples quoted from the audit, such *ad hoc* solutions may not fit to the exact evaluators needs, being therefore unworthy for the overall evaluation process.

End-user entry, path and exit. The indicators that monitor the user experience mainly focus on the calls beginning and analysts regrettably get only vague information on the user leaving the SDS. However, project owners, for example, need the number of users hanging up the phone at each node of the callflow. Therefore they need counters at every possible exit (at the beginning, the unfolding and the ending of each dialogue phase). Meanwhile, the audit revealed a need of refinement for the following indicators:

- *Transfer to a human operator*. The teams may observe the transfer ratio but they cannot identify from which exact callflow phase the user has been transferred.
- *Hang up within a dialogue phase*. This indicator only informs that a given user has left the SDS at a given moment. It does not specify if the exit is due to hanging up, transfer to another SDS or human operator or a platform failure. Thus, the evaluator lacks useful precisions to monitor the QoS, hanging up being a potential flag for an interaction design problem.
- *Number of calls dealt by the platform*. The number of calls "picked up" by the platform informs on the number of end-users arrived on the service. Yet, for a more accurate monitoring, it should be associated with the number of calls being re-routed from another service. This would enable to estimate the number of users quitting while being rerouted.

Such precisions would enable a better interpretation of the users exit. It may help distinguishing between the problematic calls due to design issues and the ones due to external motives. For example, a hang up ratio may be increased by the users dialling a wrong number or facing a personal event compelling him to end the call.

Caller history. The *métier* and development teams require a precise user history monitoring. For example, they may check if a customer having requested an automatic test of landline had actually beneficed from the service.
The observations are processed by extracting the list of calls per dialling number. A precise typology of customers can be extracted by the analysis of the recalls (1, 2, 3, etc. recalls in one day, one week, etc.). It permits to address operations by end-user profile, so as to understand the motive of their calling back for example.

Call motive. Analysts may need to study the precise repartition of calls' motives and sub-motives on a given SDS. Such information helps to optimise the customer relationship by enhancing the segmentation of the SDS and identifying the need for new branches in the service and relative human operators skills. As mentioned above, this requires to take the network of connected SDS services and the possible call transfers between them for a given call.

Reference value A major issue concerns the definition of threshold values to set alarms for *red-light indicators*. Generally, the value obtained in previous measures is used as reference. The evaluation is thus based on the estimation of the indicator variation.
Today's practice consists, while designing the service, in fixing thresholds that correspond to observed values of relative ratios for a good functioning of the service. For example: "in a regular functioning of the service, the habitual hang up ratio is X, the percentage of very short dialogs is Y, etc." In parallel, the vocal recognition experts work on reference curves to evaluate systems individually.

5.3. Homonymy and synonymy

As mentioned above, the audit revealed the absence of consistency within the definition of indicators, both among the different groups of stakeholders and across the SDSs. Each evaluator, inside its community of practice, tends to maintain its own spreadsheets to store, calculate and analyse a personal set of indicators. The existence of such bespoke solutions leads to problems in terms of:

- *Traceability*: Often based on informal relationships, an evaluator cannot guarantee the data origins and the initial collection conditions. Indicators may even have been pre-processed before they obtaining, with unknown calculation formulas.
- *Homonymy*: Under a same indicator description, different calculations may be found.
- *Synonymy*: For a similar calculation, an *a priori* same indicator can be found under several names.

The work on naming and calculation proposed by the Recommendation P.Sup24 (ITU-T, 2005) will be an excellent framework to initiate the internal harmonisation.

6. Perspectives for rationalisation

We consider two concrete measures to rationalise evaluation practices across teams. The first one is supported by the actual SDS development suite, developed internally and used for the dialogue design of every SDS developed by the Group. The second on relies on a work in progress that consists in a multi-points of views evaluation platform.

6.1. Evaluation coupled to the dialogue design

The Orange SDS development suite integrates, in parallel to the proper design functions, evaluation and user experience feedback features within the very same application. Actually, the generated interaction logs are uploaded in the application so as to project local Key Performance Indicators (KPI) into the original dialogue callflow. It provides a detailed feedback on user behaviour, node by node, in the callflow GUI. Such a feedback is of prime interest for developers to tailor the dialogue design to the users practices. This feature fulfils two requests listed in the previous section. First the counters positioned along the callflow generate extensive logs. It allows to obtaining refined indicators to precisely locate and address design issues. Second, the KPIs being defined *a priori* with the application supporting the design of all services, homonymy and synonymy phenomena are limited. Offering a unique consistent *vocabulary* across both teams and SDS projects, the design suite removes a source of misunderstanding among stakeholders, and therefore facilitates their cooperation.

6.2. Multi points of view evaluation platform

We are developing software platform that aims at supporting the various SDS projects stakeholders evaluation needs. It will support the easy creation, from a unique corpus of parameters, of evaluations adapted to their local needs. First, this advocates for a unique common database that gathers parameters retrieved from the interaction logs, the user questionnaires and the third-parties annotations. Second, the indicators used for evaluation are all defined, and shared, within the platform. Therefore such a tool guarantees that the indicators used across teams and projects are all equally defined, calculated and maintain in the same place. It assists the cooperation between teams and the cross comparisons between system performances.

7. Conclusion

We have observed a strong correlation between, on the one hand, the goals pursued by the various stakeholders and, on the other hand, their choice and interpretation of indicators. This analysis triggers the reflection on: (i) the lack of formalisation in the interactions among and across teams, which both prevents them from building on experience and leads to the loss of precious information; (ii) the need for a rationalisation in the definition and monitoring of indicators; (iii) the need to take evaluation needs as early as possible in the service design so as to dispose of appropriate indicators for analysis. Our consequent work in progress, instead of targeting an homogenisation of evaluation practices, focuses on building a framework that enables a rationalisation of practices, while respecting the cohabitation of various points of view.

8. Acknowledgements

We thank Diane Cros (IPSIS) for her contribution to the internal audit led within the Orange project teams.

9. References

ITU-T. 2005. Rec. P.Sup24, Parameters describing the interaction with spoken dialogue systems.

Tim Paek. 2007. Toward evaluation that leads to best practices: reconciling dialog evaluation in research and industry. In *Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, pages 40–47, New York. ACL, Rochester.