

## DictaNum : système de dialogue incrémental pour la dictée de numéros.

Hatim Khouzaimi<sup>1,2</sup> Romain Laroche<sup>1</sup> Fabrice Lefèvre<sup>2</sup>

(1) Orange Labs, 38-40 Avenue du Général Leclerc, 92794 Issy-les-Moulineaux, France

(2) Laboratoire Informatique d'Avignon, 339 Chemin des Meinajaries, 84911 Avignon, France

hatim.khouzaimi@orange.com, romain.laroche@orange.com, fabrice.lefevre@univ-avignon.fr

**Résumé.** Les stratégies de dialogue incrémentales offrent une meilleure réactivité, une expérience utilisateur plus aboutie et une réduction du risque de désynchronisation. Cependant, les systèmes de dialogue incrémentaux sont basés sur une architecture logicielle dont l'implantation est longue, difficile et donc coûteuse. Pour faciliter cette évolution d'architecture, nous proposons de simuler un comportement incrémental en ajoutant une surcouche à un service de dialogue traditionnel existant. DictaNum est un démonstrateur de dialogue incrémental mettant en œuvre cette démarche. Sa tâche consiste à recueillir des numéros auprès des utilisateurs. Grâce à son fonctionnement incrémental, il autorise une correction rapide des erreurs au fil de la dictée.

**Abstract.** Incremental dialogue strategies are more reactive, offer a better user experience and reduce desynchronisation risks. However, incremental dialogue systems are based on architectures that are long, difficult and hence costly to implement. In order to make this architecture evolution easier, we suggest to simulate incremental behavior by adding a new layer to an existing traditional service. DictaNum is an incremental dialogue demonstrator that uses this approach. It collects numbers dictated by the user. Thanks to its incremental behavior, it makes it possible to rapidly correct errors on the fly.

**Mots-clés :** Systèmes de Dialogue, Traitement Incrémental, Architecture des Systèmes de Dialogue.

**Keywords:** Dialogue Systems, Incremental Processing, Dialogue Systems Architectures.

## 1 Introduction

Les systèmes de dialogue incrémentaux traitent les paroles de l'utilisateur au fur et à mesure qu'elles sont prononcées, sans attendre la fin de la requête comme c'est le cas des systèmes traditionnels. Les architectures incrémentales proposées dans la littérature (Dohsaka & Shimazu, 1997; Allen *et al.*, 2001; Schlangen & Skantze, 2011) nécessitent une construction intégrale du système. Afin d'éviter cela et les coûts que cela engendre, nous proposons de rajouter un module intermédiaire entre le client et le service appelé *Scheduler* (Khouzaimi *et al.*, 2014) (fonctionnement similaire à celui de l'Incremental Interaction Manager dans (Selfridge *et al.*, 2012) et du Micro-turn Interaction Manager dans (Hastie *et al.*, 2013)). Celui-ci permet de simuler un comportement incrémental vu du client sans devoir modifier le comportement interne du service (des modifications à l'échelle applicative sont cependant nécessaires). DictaNum est un service de collecte de numéros fonctionnant suivant le même principe. Ce système est inspiré de NUMBERS (Skantze & Schlangen, 2009) qui est basé sur une architecture complètement incrémentale.

## 2 Description du fonctionnement

Un service de dialogue recueillant des numéros est déployé sur un serveur distant. Celui-ci a été développé à l'aide de la suite logicielle Disserto d'Orange Labs. Le *Scheduler* est également déployé sur le même serveur (ce qui n'est pas obligatoire, les deux entités pouvant être déployées sur des machines différentes). Le client se présente sous la forme d'une page HTML communiquant en AJAX avec le *Scheduler*. Par ailleurs, celle-ci est dotée d'un module JavaScript utilisant la web API de Google pour effectuer les tâches de reconnaissance et de synthèse vocales.

La sortie incrémentale de la reconnaissance vocale est envoyée à intervalles réguliers au *Scheduler*. Chacun de ces intervalles est appelé *micro-tour* dont la durée doit être spécifiée sur l'interface du démonstrateur. Si pendant  $n$  micro-tours, la sortie du module de reconnaissance vocale reste inchangée, on dit qu'un *micro-silence* est détecté. De même, on dit qu'un *silence* est détecté après  $m > n$  micro-tours sans aucun changement de la phrase courante de l'utilisateur. Les paramètres  $n$  et  $m$  doivent également être spécifiés dans l'interface.

Le *Scheduler* envoie toutes les réponses intermédiaires au client. Cependant, seules certaines sont désignées pour être prononcées par le module de synthèse vocale (dans le cas de DictaNum, seules les réponses suivant un micro-silence ou un silence sont prononcées). Le *Scheduler* a aussi pour tâche de décider des moments de prise de parole par le système.

Une autre caractéristique importante du dialogue incrémental est la possibilité pour l'utilisateur d'interrompre le système pendant qu'il parle. Dans le cadre de notre architecture, cela est possible à certains endroits du dialogue définis au niveau du service. Celui-ci renvoie des messages en plusieurs blocks reconnaissables à l'aide d'un séparateur : *block1 [sep] block2 [sep] block3...* Une fois reçu par le client, celui-ci le prononce fragment par fragment, et en cas d'interruption de l'utilisateur, le couple {block durant lequel l'interruption a eu lieu, contenu de l'interruption} est renvoyé au service qui réagit en conséquence.

### 3 Usage du service

L'utilisateur est invité à dicter son numéro suite à un message vocal. Au cours de sa dictée, si un micro-silence est détecté après un fragment de numéro, le système prend la parole et répète ce dernier fragment. Ensuite, l'utilisateur peut continuer sa dictée normalement ou bien corriger le dernier fragment en commençant sa prochaine phrase par 'Non'. Par exemple :

- **Système** : Bonjour. Bienvenue dans le service DictaNum. Veuillez entrer votre numéro.
- **Utilisateur** : 01 45 (suivi d'un micro-silence)
- **Système** : 01 45
- **Utilisateur** : 28 (suivi d'un micro-silence)
- **Système** : 38 (erreur de reconnaissance vocale)
- **Utilisateur** : Non, 28 (suivi d'un micro-silence)
- **Système** : Désolé, 28
- **Utilisateur** : 79 88 (suivi d'un micro-silence)
- **Système** : 79 88
- **Utilisateur** : (silence)
- **Système** : Le numéro dicté est le : 01 45 28 79 88. Est-ce bien cela ?
- **Utilisateur** : Oui
- **Système** : Merci d'avoir utilisé DictaNum. A bientôt.

Le système comprend que la dictée est terminée quand il détecte un long silence. Par la suite, le numéro complet est récapitulé et l'utilisateur peut intervenir durant ce feedback pour effectuer une correction locale. Les corrections peuvent être faites par lots de deux chiffres par exemple :

- **Système** : Le numéro dicté est : 01 45 28 72
- **Utilisateur** : Non, 62
- **Système** : Désolé : 01 45 28 62 59. Est-ce bien cela ?
- **Utilisateur** : Oui.
- **Système** : Merci d'avoir utilisé DictaNum. A bientôt.

### 4 Conclusion et travaux futurs

DictaNum montre qu'en utilisant un module intermédiaire entre le client et le service (le *Scheduler*), il est possible de simuler un comportement incrémental à moindre coût. Cependant la version du *Scheduler* utilisée ici est élémentaire et nous projetons d'en faire un réel module de décision intégrant un fonctionnement plus élaboré (analyse du contexte de dialogue, détection de problèmes, correction d'erreurs...).

## Références

- ALLEN J., FERGUSON G. & STENT A. (2001). An architecture for more realistic conversational systems. In *6th international conference on Intelligent user interfaces*.
- DOHSAKA K. & SHIMAZU A. (1997). A system architecture for spoken utterance production in collaborative dialogue. In *IJCAI*.
- HASTIE H., AUFAURE M.-A., ALEXOPOULOS P. & AUTRES (2013). Demonstration of the parlance system : a data-driven incremental, spoken dialogue system for interactive search. In *Proceedings of the SIGDIAL 2013 Conference*.
- KHOUZAIMI H., LAROCHE R. & LEFÈVRE F. (2014). Vers une approche simplifiée pour introduire le caractère incrémental dans les systèmes de dialogue. In *Proceedings of the TALN 2014 Conference*.
- SCHLANGEN D. & SKANTZE G. (2011). A general, abstract model of incremental dialogue processing. *Dialogue and Discourse*, **2**, 83–111.
- SELFRIDGE E. O., ARIZMENDI I., HEEMAN P. A. & WILLIAMS J. D. (2012). Integrating incremental speech recognition and pomdp-based dialogue systems. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.
- SKANTZE G. & SCHLANGEN D. (2009). Incremental dialogue processing in a micro-domain. In *ACL*.