



GT Auditabilité

Réunion de lancement
13/09/2017 – Paris
Compte–Rendu

Historique du document

Date	Version	Rédacteur	Évolution
14/09/2017	v1.0.0	Olivier Boux /Phast	version initiale
22/09/2017	V1.1.0	Olivier Boux /Phast	ajouts /validation par le GT

Sommaire

1	Présence	1
2	Contexte	1
3	Objectif de la réunion.....	1
4	Présentation	1
5	Discussion	2
6	Relevé de décision	5

1 Présence

Présents

(MB)	Michel BLONDEL	Phast-Services
(OB)	Olivier BOUX	Phast-Services
(JCC)	Jean-Christophe CAUVIN	Inter'op Santé
(OG)	Olivier GASCUENA	Computer Eineering
(FG)	Franck GENER	Phast-Services
(IG)	Isabelle GIBAUD	Phast-Services
(JP)	Joseph PACHOD	Berger-Levrault
(SP)	Séverine PARIS	Computer Eineering

Absent

Badreddine ZEGHICHE Socianova

2 Contexte

Pour répondre au besoin d'auditabilité des interfaces PN13 des logiciels, Phast a réalisé une API de contrôle de conformité des messages de prescription.

Dans le cadre de la communauté SIPh, Phast constitue un groupe de travail « auditabilité PN13 » pour préciser les usages de cette API et définir un jeu de test qui pourra être invoqué par un laboratoire accrédité ISO 17025.

3 Objectif de la réunion

Lancement d'un groupe de travail dont le but est de

- Préciser les usages de l'API
- Définir un jeu de tests à intégrer dans Gazelle

4 Présentation

Voir le document support de cette réunion (Auditabilité PN13 - GT R_00 (2017-09-13) - présentation)

Isabelle GIBAUD fait une démonstration des usages et fonctions de Gazelle sur l'instance d'Interop'Santé et les contrôles des profils IHE :

- Outil d'automatisation des tests d'interface utilisé pour les connectathons IHE en Amérique, Europe et Asie
 - o Plateforme utilisable à travers une IHM (ESVClient) ou un webservice
 - o Supporte tous les profils IHE internationaux et français (dont PAM 2.6 mais pas 2.8)
 - o Supporte HPrIM'XML, toujours la dernière version



- Permet l'enregistrement des systèmes pour que leur soient soumis des tests par des systèmes tiers au travers de Gazelle
- Gazelle Test Management : jeux de test proposés pour un connectathon
- Interop'Santé a acquis une instance de Gazelle, accessible à partir de son site web à tous ses adhérents

Michel BLONDEL fait une démonstration de l'API développée par Phast :

- Prend un message de prescription PN13 en entrée
 - Analyse l'ensemble du message
 - Conformité syntaxique (xsd) + conformité sémantique (code bien présent dans le jeu de valeurs attendues)
- Retourne le résultat des contrôles dans un document XML selon schéma attendu par Gazelle
 - Les non conformités ne sont pas toutes codées
 - Beaucoup apparaissent sous forme d'une chaîne de caractères
 - Il y a un marqueur global de conformité / non-conformité. Pour le détail des non conformités, il faut analyser toute la réponse pour y identifier les éléments déclarés non conforme
- Intégration Gazelle en cours : pas de démonstration possible en l'état

5 Discussion

Gazelle

Les contrôles PN13 seront-ils hébergés par la plateforme Interop'Santé ?
C'est une discussion à mener entre Phast et Interop'Santé.

Formation à l'utilisation de Gazelle ?

Interop'santé aborde l'usage réel de Gazelle lors de ses formations concernant IHE. La prochaine formation est programmée le 13 octobre prochain ([Les outils au service du test IHE](#)).

Gazelle ne traite pas des problèmes d'intégration entre les applications, notamment celui du non-alignement de concepts censés être partagés de part et d'autre des messages. Par exemple, les horaires associés aux concepts *matin*, *midi* et *soir*, entre les logiciels de prescription et celui de la pharmacie.

Certification des logiciels

PAM : La certification des LAD devrait imposer une conformité PAM (le référentiel n'est pas publié). La HAS utilisera-t-elle Gazelle et comment ?

Il est probable que la HAS reprendra un sous-ensemble des tests actuellement disponibles pour définir son jeu de test que Kereval, laboratoire de certification homologué, utilisera pour auditer les

interfaces PAM des LAD candidats. Aucune autre information n'est disponible (outils qui seront mis à disposition des éditeurs en amont de la certif, modalités d'usage, coûts, financement, ...)

PN13 : La certification des LAD se limiterait (le référentiel n'est pas publié) à l'exigence que le LAD supporte au moins une communication structurée de prescription et que dans cette communication, le médicament prescrit y est identifié par un code UCD si c'est une spécialité ou un code Médicabase si c'est un médicament virtuel. PN13 entrerait donc dans le champ des standards utilisables mais ne sera pas exigé. PN13 supportera l'usage d'un code Medicabase quand cette base sera effectivement publiée. Toute question complémentaire concernant les contenus et les modalités de diffusion de Medicabase est à poser directement à ses propriétaires et auteurs, qui n'ont à ce jour fait aucune communication publique à destination des parties prenantes industrielles concernées.

API

Est-il possible de ne contrôler que la conformité syntaxique ? Ou telle ou telle partie du message (la posologie par exemple)

Actuellement, le contrôle porte sur l'intégralité du message et de ses jeux de valeurs attendues. Une analyse partielle ne peut pas être demandée. Il est possible d'ignorer les non conformités sémantiques lors de l'exploitation du rapport.

JC Cauvin insiste pour Gazelle présente 2 points d'entrée : 1 point pour tester uniquement la structure du message (via xsd ou autre si besoin de définir des règles de contrôle) et 1 point pour l'API qui teste aussi les jeux de valeurs.

Ne supporter que la dernière version du schéma est très contraignant. Ne faudrait-il pas supporter plusieurs versions ?

La compatibilité ascendantes des versions de schéma fait que les évolutions de structure y figurent comme des alternatives à la structure antérieure. Les restrictions propres à telle ou telle version ne figurent à ce jour que sous forme textuelle dans les spécifications fonctionnelles. Elles ne sont donc pas machinables. Comme toute combinaison de ces alternatives est légitime du point de vue du schéma, elle n'est, à ce jour, pas détectable comme une non-conformité.

Dans le monde réel, il existe des messages dont la structure est une hybridation de plusieurs versions, voire comporte des balises propriétaires surnuméraires. Comment savoir à quel niveau de version est conforme le message contrôlé ?

Si les balises surnuméraires entrent dans le cadre d'une spécialisation acceptable par le schéma, elles sont conformes (spécialisation d'une balise de type ANY) et non détectées à ce jour. Sinon, elles apparaissent comme une non-conformité syntaxique avec localisation explicite.

Quant à l'hybridation de versions, voire les limites énoncées à propos du support de plusieurs versions : S'il s'agit de combinaisons d'alternatives figurant dans le schéma, elles ne sont à ce jour pas détectables.

Une déclaration de conformité d'un message de format ancien, voire qu'aucun autre éditeur n'intègre, n'est-elle pas contre-productive ?

C'est effectivement une limite. Néanmoins, il faut rappeler que la conformité à telle ou telle version du standard ne garantit qu'un *certain* niveau d'interopérabilité, niveau qui progresse d'une version à



l'autre. Par exemple, ce n'est qu'à, partir de la v3.3 de septembre 2014 que l'interopérabilité d'une posologie de fréquence autre que « tous les jours » ou d'horaire de prise type « matin, midi et soir » est garantie. Antérieurement, il était nécessaire de s'entendre localement sur les formats et systèmes de codage à utiliser car le standard n'en fournissait pas.

Il est rappelé que la [Charte BP6](#) (LESISS) permet d'exiger la dernière version.

L'usage de schematron est évoqué pour porter ces règles actuellement textuelles. Il n'est pas retenu car c'est une technique ancienne qui s'est avérée très lourde à mettre en œuvre et extrêmement difficile à maintenir. Elle est de ce fait considérée comme périmée.

L'API ne teste que la conformité syntaxique (limitée, cf. supra) et sémantique (limitée aux jeux de valeurs attendus). Comment tester que le message transmis est sémantiquement conforme au cas d'usage soumis ?

C'est l'association cas d'usage – message XML PN13 correspondant (avec, éventuellement ses variantes acceptables) qui permet d'effectuer ce test. L'API n'a pas le niveau de finesse qui consiste à dire : « c'est bien ce type d'information que j'attends ou que je n'attends pas dans ce contexte utilisateur et c'est bien cette valeur pour cette information que j'attends ». Seul un utilisateur-testeur peut réaliser ce rapprochement entre cas d'usage et valeurs attendues dans tels et tels éléments de message.

Quel est, finalement, le niveau de finesse et de codage des non-conformités rapportées par l'API ? C'est bien le rôle des éditeurs du GT d'évaluer la valeur d'usage de ces informations pour définir les améliorations nécessaires en tenant compte de la complémentarité apportée par l'intégration dans Gazelle.

Les progrès du service rendu viendront d'une évolution selon deux axes : 1. l'augmentation du périmètre fonctionnel de l'API et 2. de l'enrichissement des tests configurés dans Gazelle.

Jeu de test

Le périmètre fonctionnel du jeu de test est largement débattu : complexité des posologies (« posologies séquencées »), séquences de médicaments, séquences des messages (une modification avant une création), etc.

Il est convenu de définir comme premier palier les prescriptions initiales (pas de modification, suspension, arrêt ou annulation) aux premiers niveaux de complexité présentés dans la diapo 6.

L'objectif est d'intégrer ces premiers cas d'usage dans Gazelle afin de disposer d'une base robuste à partir de laquelle le jeu de test pourra s'enrichir progressivement de cas plus complexes.

Les éditeurs disposent d'une imposante bibliothèque de messages. Celle-ci est constituée de fichiers XL décrivant le cas d'usage et les fichiers XML des messages associés. Les contributeurs en extrairont des cas qu'ils transmettront sous cette forme au GT (fichiers XML + fichiers XML) comme contribution à la constitution du jeu de test.

Les éditeurs demandent aussi la création d'un espace de discussion ouvert à la communauté SIPh pour échanger sur la mise en place du standard, tel que c'est fait sur les groupes PAM, labo, radio ... d'Interop'Santé.

Communication

Communiqué de presse : cible éditeurs, mais aussi MOA. Ce sont deux communications différentes.

Fonctionnement du GT

Les propositions (cf. présentation, diapo 5) sont acceptées.

L'espace de travail sera constitué d'un groupe discussion et d'un espace de stockage partagé réservés exclusivement aux membres du groupe.

Les éditeurs fournissent une matière première. OB la met en forme pour constitution du jeu de test. Concertation / validation en réunion webex. Les contributeurs assume la propriété intellectuelle des ressources qu'ils mettent à disposition à travers le groupe de discussion. Ces ressources provenant de leur expérience de validation d'interface avec d'autres éditeurs, il pourrait être estimé qu'elles ne doivent pas être partagées en l'état. Dans ce cas, il est décidé de les adresser directement à OB qui en extraira les éléments utiles pour la constitution du jeu de test et seuls ces éléments utiles seront partagés après accord du contributeur.

Les dates planifiées pourront être révisées si besoin

6 Relevé de décision

Membres du GT : Les présents à la réunion de lancement

Calendrier : Confirmation de la proposition (cf. présentation dia 5)

- Prochaine réunion : Ve 22/09 11h-12h30 webex, réservé aux membres du GT

Espace de travail

- OB crée un groupe de discussion et un espace documentaire partagé
- Réservé aux membres du GT

Périmètre fonctionnel : définition du premier palier à atteindre

- Prescription élémentaire
- Niveaux de complexité selon axes présentés en séance (cf. présentation, dia 6)

Jeu de test

- Contributeurs : SP, OG, JP
- Format : Fichier XL « computer » et son XML associé



- OB analyse les contributions et en extrait les cas d'usage répondant au périmètre défini pour les soumettre au GT

API

- Importance majeure de son intégration à Gazelle
- Pour usage direct de l'API, se mettre en relation directe avec MB
- Préciser le problème du support de différentes versions du standard