

PROBLEMATIQUE

Configurateur Extranet

SOMMAIRE

1. Itération n° 1 : Maintien en RAM de l'ensemble des données 3
2. Itération n° 2 : Développements sur la technologie IBM i sur le *Device Handler* 4
3. Itération n° 3 : Synchronisation par files d'attentes de données (Data Queues) 5

1. ITERATION N° 1 : MAINTIEN EN RAM DE L'ENSEMBLE DES DONNEES

Après avoir analysé l'existant, et évalué le principe sur plusieurs cas types, il est apparu que cette solution n'était pas la plus pertinente.

1. La structure très complexe des données à maintenir en mémoire, plus que leur volume, présentait un réel défi à retranscrire pour une persistance en RAM sur un serveur Web ou un poste individuel.
2. Il aurait aussi fallu maintenir en mémoire tous les curseurs qui permettent de parcourir les nombreuses tables de la base de données qui décrivent une configuration de manière statique (variables, enchaînement des questions, listes de valeurs, matrices de contrôle, etc.), et dynamique (piles d'exécution, valeurs intermédiaires des réponses, etc.). De manière générale, il aurait fallu émuler ou reproduire d'une manière ou d'une autre une partie des tâches du moteur de base de données du serveur.
3. Le flot d'exécution des programmes qui s'enchaînent lors du déroulement d'un questionnaire de configuration, et qui correspond finalement à l'exécution "juste à temps" d'instructions dans une pseudo-langage propre à ERP, n'est pas interruptible hors des interactions avec l'utilisateur, c'est à dire, schématiquement, dans 2 cas : la présentation de la liste des questions et des réponses déjà apportées et la demande d'une réponse à une question. Autrement dit, les opérations d'entrée-sortie.

Ces constats ont permis de se forger la conviction que pour solutionner de manière satisfaisante tous les points d'achoppement, il fallait soit repenser complètement toute l'architecture de notre moteur de configuration, soit déporter de nombreux traitements sur le poste client ou le serveur Web, ce qui revenait finalement à implémenter sous une autre forme la quasi-totalité de ce qui fonctionne déjà sur le server IBM i, et donc à "réinventer la roue" d'une certaine manière.

En se rappelant que l'objectif de base est de réutiliser au mieux l'existant, il a été décidé d'explorer une autre voie.

Les deux premiers points trouvent leur solution, dans l'état actuel des travaux, dans le principe de démarrer une session de travail sur le serveur : un vrai "job système". Un processus identifié, dont le cycle de vie est géré par l'OS disposant d'un contexte persistant et des mêmes ressources et possibilités qu'une session interactive 5250. Sauf, malheureusement, l'accès aux entrées-sortie par flot de données 5250. Et ceci est donc devenu un nouveau point technique à résoudre.

Néanmoins à ce stade de nos recherches, il a été établi que le point d'entrée du flux serait le lancement d'un questionnaire de configuration dans le cadre suivant :

- Un utilisateur ERP identifié
- Un environnement complet (liste de bibliothèques, données contextuelles, espace de stockage temporaire) – géré et maintenu par le travail sur le serveur
- Un contexte fonctionnel connu (création de ligne de commande ou de devis, pour un article donné et une quantité commandée connue)

2. ITERATION N° 2 : DEVELOPPEMENTS SUR LA TECHNOLOGIE IBM I SUR LE *DEVICE HANDLER*

Les travaux précédents ont montré qu'il fallait se concentrer sur les opérations d'entrée-sortie, le point de contact entre l'utilisateur et le déroulement du questionnaire. Le problème à résoudre étant l'impossibilité d'utiliser le flot 5250 pour communiquer avec l'utilisateur.

Une des options qui s'offrait à nous est la notion de *Device Handler*, une technologie développée par IBM qui permet d'ouvrir les programmes serveurs développés en RPG à d'autres ressources que les écrans 5250 ou les tables de bases de données

(https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzasd/openacchandlers.htm)

Dans le principe, un "Handler" expose une API standard avec des opérations simples (OPEN, READ, WRITE, etc.) et le programme qui l'utilise n'a que faire de la nature réelle de ce que le Handler représente, et continue d'effectuer les opérations classiques de gestion : lecture, écriture, mise à jour. Le programme qui gère le Handler, par contre, est dédié à une ressource bien spécifique (fichier Excel, fichier XML structuré, etc.).

Une idée a été de créer un Handler pour les écrans 5250 identifiés comme faisant partie du scénario de déroulement d'un questionnaire de configuration, et, dans le cas d'une exécution par une session Web, se servir ce canal pour communiquer avec la session du navigateur internet du poste client.

Après avoir effectué des recherches, nous avons découvert que seulement un très petit nombre d'éditeurs tiers proposaient cette technologie (et aucun de manière adaptée à nos besoins, et toujours intégrée à leur propre solution) et que nombre de fonctions n'étaient pas, ou très peu documentées par IBM. Nous avons donc tenté de définir notre propre Handler à partir des informations et des exemples existants. Nous avons pu mettre au point certains prototypes fonctionnels, mais nous avons aussi établi qu'il était loin d'être évident de concevoir un Handler générique qui aurait été adapté et connectable à l'ensemble de nos programmes, car il fallait coder "en dur" la prise en compte de chaque élément de donnée à gérer.

Ce résultat en demi-teinte nous a fait opter pour une nouvelle solution pour gérer les interactions client-serveur en se passant de l'interface 5250.

3. ITERATION N° 3 : SYNCHRONISATION PAR FILES D'ATTENTES DE DONNEES (DATA QUEUES)

La dernière évolution utilise désormais des files d'attentes de données, qui sont utilisées pour synchroniser les échanges entre les sessions Web et les travaux sur le serveur.

A chaque besoin d'entrée-sortie dans le déroulement d'un questionnaire de configuration, on a désormais un lien bi-canal asynchrone qui permet de gérer les interactions client vers serveur (saisie de réponse à une question) et serveur vers client (liste des questions, message d'erreur).

Chaque partie est en attente sur un canal prédéfini, et réagit instantanément à l'arrivée de données en provenance de son interlocuteur. Ceci est rendu possible par l'utilisation de Files d'Attentes de Données (Data Queues), un type d'objet système existant sur les serveurs IBM i (http://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_73/apis/vt1e.htm).

Entre deux interactions, la session Web et le travail sur le serveur sont déconnectés l'un de l'autre. Chacun a son propre cycle de vie. De ce fait, il a été nécessaire d'élaborer un mécanisme d'identification pour faire le lien entre une session Web et un travail serveur, afin que l'on garantisse que chaque transaction trouve sa bonne cible : la session Web pour un envoi de données du serveur, le travail sur le serveur (qui assure donc la persistance de tout son contexte d'exécution) pour une réponse à une question de la part de l'utilisateur.

Ce mécanisme d'identification permet aussi de gérer les travaux serveur qui sont inactifs depuis trop longtemps, et peuvent donc être arrêtés pour libérer des ressources systèmes.

Ce principe d'échange au moyen de files d'attentes de données, pour l'heure, été intégré spécifiquement dans les programmes serveur concernés par le configurateur.

Il n'est pas encore décidé de le généraliser à tous les programmes de l'ERP, mais cette possibilité est gardée à l'esprit lors des développements : il serait intéressant d'étendre le mécanisme à d'autres fonctionnalités de l'ERP, voire à son intégralité.