

**Reprendre le contrôle d'un legacy Mainframe de 10M de lignes de code
avec l'IA & l'Expertise d'une équipe SI.**

Compte rendu rédigé par ANDSI & Pierre Delort

En bref...

Philippe MARTINET, DSI du Groupe Hachette Livre et membre de l'association, explique comment il reprend le contrôle d'un système *legacy* né avant le minitel en combinant IA (Intelligence Artificielle) et expertise humaine. Ingénieur formé à l'ESIEA et à l'ESCP, il a dirigé auparavant les systèmes d'information de plusieurs groupes, dont Eutelsat et ADP.

L'Association Nationale des Directeurs des Systèmes d'Information organise des débats et en diffuse des comptes rendus, les idées restant de la seule responsabilité de leurs auteurs. Elle peut également diffuser les commentaires que suscitent ces documents.

Hachette Livre

Philippe MARTINET indique que le groupe Hachette Livre rassemble plus de 200 marques d'édition dans plus de 70 pays dans le monde qui évoluent dans des segments éditoriaux variés : littérature, éducation, jeunesse, pratique, tourisme, art de vivre, pour instruire, cultiver et divertir. Le Groupe se diversifie également dans les jeux et les collections.

La chaîne de valeur du livre implique plusieurs acteurs : l'auteur crée le livre, puis interviennent l'éditeur, l'imprimeur, le diffuseur, le distributeur et les points de vente, avant d'atteindre le lecteur.

La facturation, le transport, la gestion des retours font de la distribution un maillon stratégique de la chaîne de valeur du livre. En plus de son métier d'éditeur, Hachette Livre propose ses services de diffusion et de distribution à des éditeurs partenaires et assure le réassort des libraires en moins de 48h.

Contexte informatique

Le système d'information créé il y a près de 50 ans repose en bonne partie sur un Mainframe IBM. Il est robuste mais de nombreuses évolutions successives l'ont rendu difficile à faire évoluer avec un enjeu particulier à conserver les expertises au fil des années. L'impossibilité de changer le legacy en mode « big-bang » a conduit à rechercher une autre approche : Rendre le SI du mainframe sécable afin de pouvoir remplacer étape par étape l'ancien système spécifique par des solutions du marché. Un projet de cartographie de l'obsolescence a été lancé par nos experts informatiques pour accroître la maîtrise de l'environnement technique, le système réalisant chaque nuit un aiguillage entre des dizaines de milliers de programmes selon les traitements réalisés.

Le découpage du mainframe par tranches, tout en maîtrisant les jointures essentielles pour détourer le système, nécessite une équipe technique ayant bien compris les enjeux techniques et fonctionnels et utilisant une approche empirique.

L'équipe a tenté de modéliser les structures, puis effectué des recherches manuelles dans la documentation pour obtenir une vision claire du système legacy et identifier les points de jointure entre les programmes.

La démarche de rétro-ingénierie était itérative pour reconstruire une vision claire des périmètres applicatifs à partir du code existant :

- Extraction du code source du périmètre étudié ;
- Identification du type de traitement (batch ou TP) ;
- Reconstitution de la chaîne d'ordonnancement (batch) ;
- Modélisation des dépendances (JCL, fichiers, sous-programmes) ;
- Qualification fonctionnelle des composants pour reconstituer les processus métiers.

Il s'agissait de cartographier, qualifier et outiller les traitements de manière reproductible.

Ces analyses ont mis en évidence le caractère tentaculaire du système, demandant plus de dix ans à nos équipes pour arriver à bout de cette cartographie : beaucoup trop long et trop cher, aux dépens des besoins d'évolutions fonctionnelles rapides de nos métiers. Le besoin d'automatisation de nos procédés au moyen d'un LLM (Large Language Model) est paru une évidence, les initiatives autour de l'IA étant dans l'air du temps. La question de la fiabilité et de la complétude de cette approche s'est posée, notamment si ce modèle LLM risquait d'avoir des hallucinations, ou s'il trouverait toutes les informations pertinentes en maximisant et en systématisant les processus de cartographie. L'outil devait également être ergonomique pour faciliter l'intégration des nouveaux collaborateurs qui deviendront la prochaine génération d'experts du SI.

HAMI – L'IA comme levier d'accélération des études de rétro-ingénierie mainframe

L'outil que nous avons développé, Hachette Assistant for Mainframe Information (HAMI), est parvenu à lire le code et à générer des documents. En revanche, la vérification de la qualité de l'information générée restait nécessaire.

Nous avons fait le pari de l'intelligence artificielle : HAMI utilise une API LLM, Gemini pour :

- **Lire et comprendre** automatiquement le code Mainframe ;
- **Générer une documentation** claire et homogène ;
- **Visualiser** les chaînes de traitement ;
- **Discuter** avec le code en langage naturel.

Notre outil présente les fonctionnalités suivantes :

Ordonnanceur

Le parsing des codes source JCL et des exports de notre ordonnanceur permet une extraction directe des liens d'ordonnancement présents dans les scripts JCL.

Explorateur de composants

Permet une navigation et la documentation des objets Mainframe. Il s'agit de centraliser l'information technique et fonctionnelle autour d'un composant mainframe, pour en faciliter la compréhension, la documentation et la transmission :

- Recherche ciblée par Job, fichier ou programme ;
- Génération automatique de fiches descriptives (modifiable) ;
- Schéma graphique de la structure d'appel :
 - Composant central relié à ses dépendances (input/output) ;
 - Visualisation intuitive des interactions ;
- Affichage du code source .

Comparateur de jobs

Cette fonctionnalité permet d'identifier rapidement les liens fonctionnels ou techniques entre plusieurs jobs batch, pour comprendre les zones d'interdépendance, anticiper les effets de bord d'un détournement, ou isoler des blocs fonctionnels cohérents. Cela en fait un outil d'aide à l'analyse d'impact rapide, visuel et modulaire

Assistant Chatbot,

pour une interaction naturelle avec le code

Requêteur

Pour interroger dynamiquement la base de données HAMI à l'aide de requêtes en langage naturel, pour extraire des listes ciblées selon des critères complexes.

- Recherche en langage naturel (Prompt to SQL) :
 - L'utilisateur saisit sa requête en français (ex. : « *Donne-moi tous les jobs qui appellent le programme X* ») ;
 - Le module la traduit automatiquement en SQL ;
- Aide à la formulation → Une fiche de documentation décrit le modèle de données pour guider les utilisateurs.
- Résultats personnalisables & exportables.

Nous avons développé un outil opérationnel, du *run* au *build*

En run (exploitation, diagnostic d'incidents), il permet de :

- D'identifier rapidement les dépendances techniques entre jobs, programmes, fichiers (via l'Explorateur) ;
- D'analyser un incident en suivant une chaîne batch visuellement (Ordonnanceur) ;
- D'analyser les similitudes et écarts entre deux jobs pour comprendre leurs dépendances croisées, les échanges de fichiers ou leur articulation dans l'ordonnancement (Comparateur) ;
- D'accélérer la compréhension d'un fonctionnement applicatif sans lecture manuelle du code (Explorateur / Chat bot) ;
- De faciliter la collaboration entre équipes (études, support, infra) avec un référentiel partagé.

En build (projet), il permet de :

- Cartographier un périmètre applicatif pour cadrer un chantier de refonte ;
- Qualifier automatiquement les assets à reprendre, migrer ou isoler ;
- Détecter les écarts entre l'existant et une version projetée ;
- Générer des éléments de spécification technique à partir du code analysé ;
- Définir des zones fonctionnelles pertinentes pour des travaux de découpage ou de modernisation.

Nous avons fait de l'IA un partenaire quotidien pour les équipes IT, augmentant nos experts sans qu'ils en perdent le contrôle, l'IA restant au service de l'homme.

Débat

Int : La Génération Augmentée de Récupération (RAG) a-t-elle été envisagée ?

PM : Cette fonctionnalité est déjà intégrée. La base vectorielle intègre tous les liens entre les programmes (ordonnanceurs, JCL, routines dans les programmes, partage de fichiers, bases de données...) et a été structurée de façon à répondre à toutes les questions.

Int : Comment avez-vous amélioré les résultats de l'outil ?

PM : La méthodologie employée consistait tout d'abord à identifier un besoin à automatiser qui était communiqué à l'IA. Ensuite, une comparaison a été établie entre l'expertise humaine et les résultats de l'IA. Un module exploratoire complétait le dispositif.

Notre équipe a validé le travail de l'IA en réorientant le prompt lorsqu'une erreur ou une imprécision était détectée. Le requêteur *chat bot* a ainsi gagné progressivement en performance. Nos experts ont recommandé de ne pas pousser l'automatisation trop loin afin de conserver un travail intellectuel d'appropriation des informations collectées pour garder la maîtrise humaine du processus. Le niveau de confiance envers l'outil a considérablement augmenté.

Int : L'IA génère-t-elle le nouveau code source ?

PM : A ce stade, il ne s'agit pas de génération de code, mais de cartographier totalement le mainframe et de rendre sa description technique et fonctionnelle intelligible, juste et complète.

Int : Avez-vous réussi à intégrer une fonctionnalité d'évaluation automatique des risques cyber ?

PM : Non, pas pour l'instant, mais c'est une remarque pertinente. Cette fonctionnalité sera ajoutée dans les prochaines versions du système. L'analyse des risques cyber constitue effectivement un axe de développement important.

Int : Comment la trajectoire du POC a-t-elle été gérée avec l'équipe lors de la phase d'amélioration de votre outil ?

PM : Le pilotage hebdomadaire a poussé le groupe de travail à tester un maximum de cas techniques plutôt que d'aller en profondeur dans les quelques périmètres fonctionnels choisis. J'ai fait le choix de développer les fonctionnalités du produit basé sur l'IA plutôt que d'apporter des réponses immédiates à des cas métier. Le but de ce POC était de prouver que nous pouvions reprendre la main sur le patrimoine du Mainframe et de permettre le détournement un à un des applicatifs. C'était parfois frustrant car l'envie était grande d'analyser un lot de traitements fonctionnels complets pour immédiatement apporter de la valeur au métier. Cela aurait pu être une victoire à la Pyrrhus car toute notre capacité à créer un outil puissant aurait été détournée pour un résultat immédiat et fragile sous la pression du métier qui aurait pris trop tôt goût à un outil pas assez éprouvé et polyvalent.

Int : À quel moment l'équipe de la DSI confronte-t-elle ses résultats avec ceux des Métiers ?

PM : Au stade de POC, il était prioritaire de tester notre outil dans toutes les situations techniques identifiées (i.e. plus d'une dizaine de langages de programmation, des multiples moyens d'appeler les programmes et partager les données) pour avoir confiance dans notre capacité et notre nouvelle vélocité à apporter de la valeur aux métiers. Nous sommes maintenant prêts à lancer un projet pilote de refonte d'une partie du code avec une équipe intégrant le métier avec la DSI.

Présentation de l'orateur

Philippe MARTINET, ingénieur ESIEA & MBA ESCP, a dirigé, avant celui d'Hachette Livres, les SI de plusieurs groupes, dont Eutelsat & ADP.