

Quatre idées fortes de Praxeme pour SOA

Objectif Praxeme est une méthodologie d'entreprise. Elle contient, notamment, les procédés nécessaires à la conception d'une architecture de services.

Cet article résume les principaux messages de Praxeme pour réussir les projets SOA.

- Contenu**
- SOA : un style d'architecture de SI
 - SOA de surface *versus* SOA de refonte
 - SOA et urbanisation de SI
 - SOA : la méthode

Rédacteur Dominique VAUQUIER

Version 2.0, le 31 mars 2015 (première version : le 7 avril 2007)

SOA : un style d'architecture de SI

Le service, atome du système informatique

La technologie nous offre des dispositifs de structuration et de communication qui permettent de donner du sens à la métaphore du service, appliquée aux systèmes informatiques. Car c'est de cela qu'il s'agit : une métaphore. Au sens qu'il prend dans le sigle SOA, *service oriented architecture*, le service est une unité de composition d'un système informatique. Si nous respectons cette métaphore, la notion de service désigne, d'ailleurs, l'unité de traitement la plus petite, qui peut être sollicitée. C'est donc une opération plutôt qu'un ensemble d'opérations. J'ai besoin d'une information, je déclenche une action sur le système, je demande une transformation, etc. : ceci s'obtient en activant un service dans le système, plus ou moins directement. Dans cette vision, les données sont masquées, protégées par les services qui garantissent l'intégrité du système.

La structuration du système

De tels services, le système informatique en comporte des milliers. Se pose alors la question de leur regroupement, de leur structuration optimale. Sur ce point, interviennent les préoccupations de réutilisation, de non redondance, de faible couplage et de maîtrise du volume, de la complexité et des coûts. Bref, nous abordons l'architecture du système informatique.

L'architecture technique

Il faut s'entendre sur le terme « architecture ». *L'architecture technique* fournit les conditions de faisabilité d'une architecture de services. Elle peut recourir à des nouvelles solutions mais peut, également, comporter des technologies plus classiques : l'important restant la métaphore du service. Les choix de structuration du système ne sont pas du ressort de l'architecture technique. Ils mobilisent une autre discipline : *l'architecture logique*.

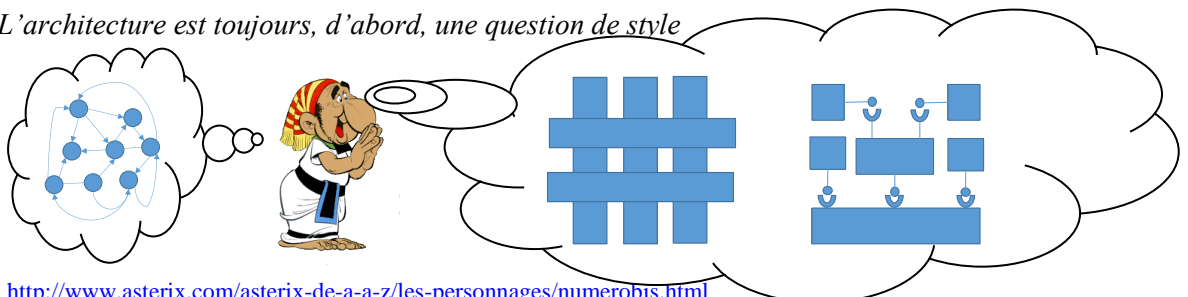
L'architecture logique

Une fois établies les conditions de faisabilité technique, SOA est une affaire d'architecture logique. La question est : « comment structurer au mieux le système informatique ? ». Le procédé consiste à appliquer quelques préceptes et contraintes topologiques, à s'imposer quelques règles formelles, afin d'organiser la substance du système. L'aspect logique est un aspect du « système entreprise », intermédiaire entre les aspects « amont » (le métier, l'organisation) et l'aspect informatique. Le modèle logique décrit le système informatique en des termes relativement indépendant des technologies. De là découlent deux avantages : d'abord, c'est un modèle plus facilement communicable ; ensuite, puisque protégé des évolutions techniques, c'est un modèle qui jouit de la stabilité nécessaire à une entreprise de long terme.

Le style SOA

L'architecture logique a connu, dans l'histoire de l'informatique, plusieurs époques et plusieurs styles. Le dernier en date était l'architecture fonctionnelle : la structuration du système à base de fonctions ou de fonctionnalités. Le critère de décomposition dominant dans cette approche est le domaine fonctionnel. Dans le style SOA, le terme élémentaire est le service et la question du critère de décomposition trouve des réponses plus sophistiquées.

Figure SLB-15_1. L'architecture est toujours, d'abord, une question de style



© Gosciny & Uderzo, <http://www.asterix.com/asterix-de-a-a-z/les-personnages/numerobis.html>

SOA de surface *versus* SOA de refonte

« SOA de surface » Dans les publications, l'appellation SOA recouvre deux pratiques fort différentes, tant dans leur intention que dans leurs retombées sur le système d'information. La réalisation de quelques services plaqués sur un système existant permet d'en faciliter l'accès et de l'ouvrir sur d'autres systèmes. Elle constitue un apport important mais ne modifie pas fondamentalement le système lui-même. Nous qualifions cette pratique de « SOA de surface ». De nombreux projets SOA entrent dans cette catégorie.

« SOA de refonte » SOA qualifie, aussi, une approche plus radicale du système, approche qui donne son sens plein au terme « architecture ». Nous parlons, alors, de « SOA de refonte ». Cette appellation caractérise une pensée architecturale qui vise à restructurer le système, perçu comme un tout, à bannir toute redondance et à pousser à son maximum la réutilisation. Dans cette vision, le système n'est plus un ensemble d'applications, ensemble forcément redondant, mais un mécano de services dans lequel un même comportement ou une même règle ne sont programmés qu'une fois.

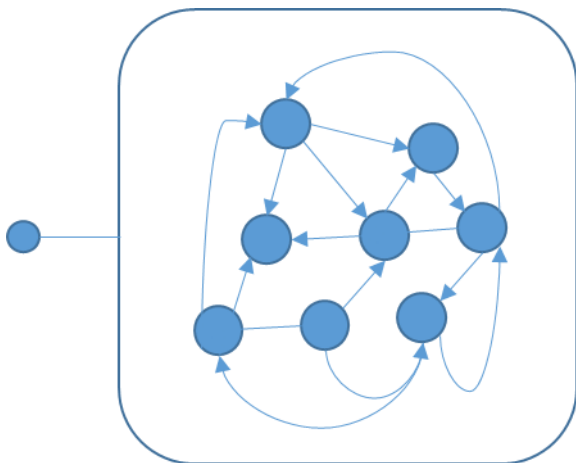
Une réforme progressive du SI

Cette reconstruction du système ne se fait pas, heureusement, en une fois. Il nous faut bien constater notre incapacité collective à mener de tels projets. L'avantage avec SOA réside dans la possibilité de refondre le système à partir d'une portion, et progressivement. Les projets de réalisation des services restent ainsi à une échelle contrôlable. Évidemment, pour que la réforme du système se poursuive sans dévier de sa trajectoire, les DSI doivent respecter plusieurs conditions parmi lesquelles la préexistence d'une cible rigoureusement élaborée, l'homogénéité de la méthode et quelques dispositions organisationnelles. Il est important de prendre la mesure de l'impact sur l'organisation. En effet, l'approche SOA subordonnant l'objectif immédiat à l'intérêt général, la DSI entre dans une nouvelle dynamique, avec un rapport nouveau entre les projets et les activités transverses.

Figure SLB-15_2. La qualité du système selon le degré de la SOA

SOA de surface

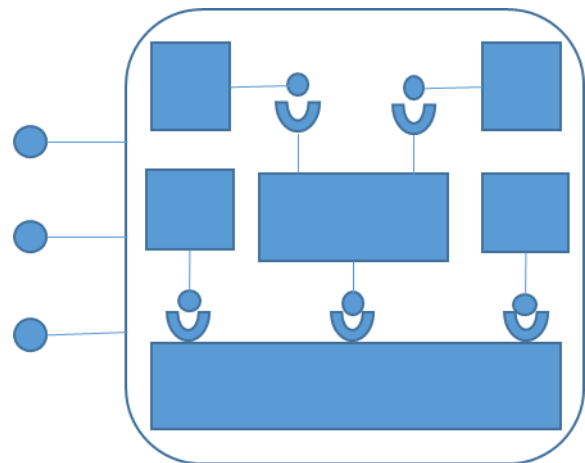
Le système fournit quelques services mais sa structure interne n'a pas évolué.



La réutilisabilité et la qualité interne n'ont pas progressé.

SOA de refonte

Le système fournit d'autant plus de services (interopérabilité) qu'il a été correctement structuré.



Le système est lui-même un assemblage de services, sans redondance et avec un couplage maîtrisé.

SOA et urbanisation de SI

Les différences

Il nous faut, maintenant, évoquer le rapport entre SOA et l'urbanisation de SI.

Ce rapport n'est pas, de prime abord, spontané, quoi que prétendent les bricolages marketing. En effet, les tenants de l'une et l'autre approches se font, de la substance du système informatique, une idée totalement différente et difficilement conciliable.

La perception du SI par l'urbanisation classique

Le discours orthodoxe de l'urbanisation de SI se caractérise par :

- le primat accordé à l'activité (exclusivité accordée à la modélisation des processus pour décrire le métier, prévalence des domaines fonctionnels) ;
- l'unité de l'application comme composant de base du système informatique.

La perception du SI dans le courant SOA

À l'opposé, l'architecture de services – du moins, dans la méthode Praxeme – fait place à des constituants qui se déduisent d'un modèle sémantique, en amont des processus. Elle introduit des « domaines d'objets », siège des services hautement réutilisables. Les objets « métier » ne sont pas secondaires par rapport aux processus et activités : ils sont antérieurs et vont influencer des choix très structurants dans l'architecture logique. Par ailleurs, cette philosophie du système informatique relativise l'application comme unité : sans disparaître, l'application se vide de sa substance et devient un lieu d'assemblage de services partageables. Ces services sont conçus dans une intention de réutilisation et dans la perspective d'une réorganisation complète du système en tant que tel. La perspective dominante n'est pas celle des projets : elle suppose un point de vue supérieur, porté par l'organisation de la DSI.

Les convergences

Malgré ces différences fondamentales¹, SOA et urbanisation de SI peuvent converger et se renforcer mutuellement. Pour cela, il convient de revisiter le discours et la pratique de l'urbanisation et d'adopter un nouveau discours, intégré à une méthodologie d'ensemble. La convergence entre SOA et urbanisation de SI apparaît si on revient à leur finalité. Il y a, en effet, communauté d'intention : les deux approches visent à faire évoluer le système vers une structure optimale. Leur positionnement est ou doit être le même : portée globale – l'échelle du système, la considération de l'entreprise – et longue durée. Cette posture est naturelle et, on peut dire, constitutive de l'urbanisation de SI, depuis sa première expression par Jacques Sassoon. Elle l'est moins côté SOA, cette approche ayant une origine essentiellement technicienne.

Les divergences

Si on compare les représentations que produisent, d'un côté, les urbanistes classiques, de l'autre, les architectes logiques, on constate que les cibles d'urbanisation, POS et autres PLU, dans leur grande majorité, se bornent à poser un découpage du système. Ces représentations sont *statiques* et exclusivement *fonctionnelles*². L'adjonction de « référentiels » et de « gisements de données », loin d'arranger l'affaire, montre que cette approche n'adopte pas le principe d'encapsulation. Selon ce principe, la donnée n'est pas vue de l'extérieur : on n'y accède que par... un service. Ce même principe d'encapsulation se tapit au cœur de la notion de service. Dans le style SOA, le modèle logique est tel qu'un plan des services surplombe et masque un plan des données.

¹ J'ai bien conscience d'avoir forcé le trait. Disons que la caricature est un procédé pédagogique. *For the sake of argument...*

² Le terme « fonctionnel » est pris, dans ce papier, dans son sens strict : relatif à la fonction, donc lié à l'action, à l'activité. Pour nous, l'architecture fonctionnelle est une architecture logique dont le critère de structuration est la fonction. Je ne cache pas ma perplexité face aux démarches qui concilient un plan fonctionnel et un plan logique. Un cadre aussi verbeux engendre des travaux inutiles.

Le graphe d'architecture logique

L'architecte logique, adoptant le style SOA, élabore la cible d'urbanisation sous la forme d'un graphe d'architecture logique. Au premier abord, il s'agit de la même chose que la représentation établie par l'urbaniste. Néanmoins, l'art de l'architecte logique est très différent : il adopte une notation plus rigoureuse qui lui permet d'étudier la dynamique du système en même temps que sa statique. Quand il découpe le système en grands constituants, aussitôt il s'intéresse aux dépendances que ce découpage suppose, aux appels et échanges qu'il engendre. S'il découpe à tel endroit plutôt qu'à tel autre, c'est qu'il sait que ce choix optimise le système, dans son comportement. De plus, le graphe d'architecture logique n'est que le premier des diagrammes qui révèlent le modèle logique. Avec la même notation, le concepteur logique poursuivra son travail jusqu'à la spécification détaillée des services logiques. Il y a donc continuité de la chaîne d'activité et le développeur exploitera sans heurt le modèle logique : il le traduira en logiciel, en appliquant les choix d'architecture technique. Ce principe de continuité fait taire le reproche adressé aux urbanistes, du point de vue des développeurs : produire une grande carte qui ne concerne pas les projets.

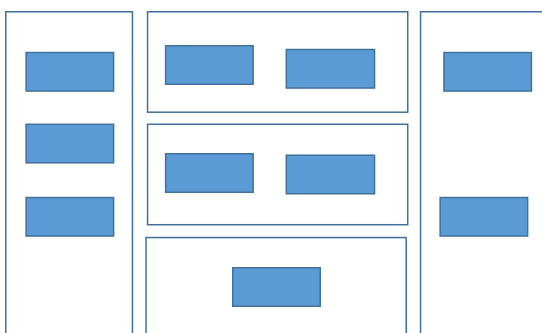
Le rôle de l'urbaniste de SI

Que reste-t-il donc à l'urbaniste de SI, que l'architecte logique ne pourrait assumer ? C'est son rôle de passeur entre le métier et l'informatique. L'urbaniste de SI – ou l'architecte d'entreprise – est le seul à pouvoir recueillir les orientations stratégiques de l'entreprise, à s'intéresser aux préoccupations générales du métier, à anticiper ses évolutions et à pressentir leurs impacts sur l'outil informatique. Ce rôle de passeur, l'architecte logique ne peut pas l'assumer pleinement : il est trop préoccupé par le système informatique lui-même, au risque de se laisser séduire par des considérations formelles, étrangères au métier. Il est plus un homme de la technique qu'un acteur du métier.

En conclusion de ce paragraphe, il y a complémentarité et nécessité de collaboration entre l'urbaniste de SI et l'architecte logique, les deux disciplines intervenant sur le même aspect : l'aspect logique. Cet aspect est volontairement posé comme intermédiaire entre le métier et l'informatique. En cela, il intervient dans la fonction de passage entre le métier et la solution technique, fonction assumée par l'urbaniste ou l'architecte d'entreprise. Au-delà de cette fonction de passage, il s'agit de concevoir totalement le système technique, à tous les niveaux de détails nécessaires pour que le système fonctionne et présente les qualités requises. Ce travail de conception se place sous la responsabilité de l'architecte logique.

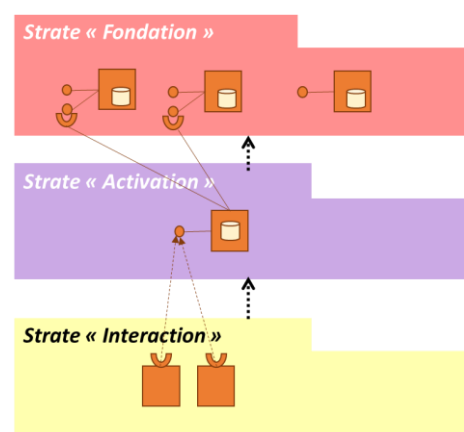
Figure SLB-15_3. Deux types de représentations du système informatique : le POS et le graphe d'architecture

Schéma type d'un POS (plan d'occupation des sols) dans la pratique de l'urbanisation de SI



La représentation montre la statique du système. Elle ne permet pas d'étudier les dépendances créées, ni de révéler la redondance, ni d'anticiper le comportement dynamique du système.

Schéma type pour un graphe d'architecture selon la méthode Praxeme



L'architecture du système obéit à des contraintes topologiques strictes. Elle est décrite de telle sorte que l'on puisse en apprécier la qualité.

SOA : la méthode

Les questions

La mise en œuvre d'une architecture de services soulève de nombreuses questions pratiques. Tout particulièrement :

- Comment trouver les services ?
- Comment structurer le système ?

Faute de répondre à la première question, les projets risquent de ne produire que des accesseurs, des services du genre : créer, modifier, supprimer. Était-il besoin de tant de bruits pour arriver à cela ? Même si on évite cet extrême³, l'absence de réponse établie entretient l'hétérogénéité des pratiques, entre plusieurs concepteurs, entre plusieurs projets... Cette situation ruine tout espoir d'amélioration du système.

La question de la bonne structuration appelle, elle aussi, des réponses précises.

Les réponses

C'est le rôle de la méthodologie que de répondre à de telles questions et d'expliquer comment faire. La méthode publique Praxeme répond à ces deux questions, parmi beaucoup d'autres. Elle se fonde sur un cadre de référence qui ne se limite pas à l'aspect logique mais embrasse tous les aspects du « système entreprise ». Cette couverture constitue une des conditions pour mener correctement des activités comme l'architecture logique et l'urbanisation de SI. En effet, Praxeme préconise un procédé de conception des services, par dérivation des modèles « métier ». Praxeme pose deux modèles comme préalables au modèle logique :

- le modèle sémantique, qui exprime les fondamentaux du métier ;
- le modèle pragmatique, qui décrit l'activité de l'entreprise (organisation, processus, cas d'utilisation).

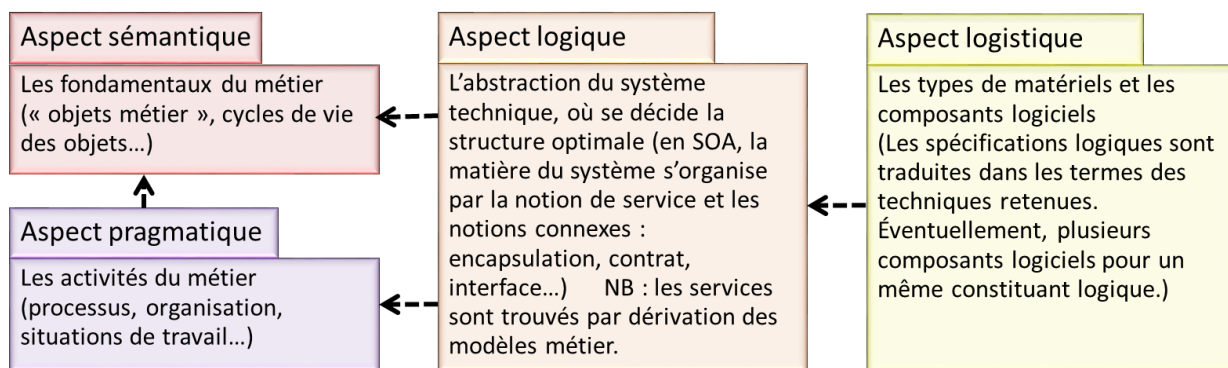
Les modèles « métier »

Ces deux modèles, pour être exploitables en aval, obéissent à des critères précis. Par exemple, le modèle sémantique ne se limite pas à un modèle conceptuel des données.

De même, pour appliquer la dérivation sur un modèle pragmatique, on ne peut pas se contenter des diagrammes de cas d'utilisation tels qu'on les trouve sur la plupart des projets : ils recèlent une trop grande redondance.

Ce n'est qu'à partir de modèles sémantiques et pragmatiques « bien formés » et correctement structurés que l'on peut appliquer les règles de dérivation. Celles-ci produisent, mécaniquement, les services et les constituants logiques, du moins l'essentiel d'entre eux.

Figure SLB-15_4. Le positionnement de l'aspect logique⁴



³ Extrême mais pas caricature : c'est un constat fréquent.

⁴ Pour plus de détail, voir « La séquence fondamentale », référence SLB-56.

Les disciplines de l'aspect logique

Sur l'aspect logique, Praxeme distingue les disciplines de l'urbanisation de SI, de l'architecture logique et de la conception logique. Ces activités s'articulent soigneusement. Le recours à une notation standard, en l'occurrence UML, facilite grandement les travaux. Un autre standard essentiel à cette chaîne de production du SI est le standard de l'OMG : MDA (*model driven architecture*). La technique de transformation des modèles nous fait toucher au but : nous sommes en mesure, maintenant, de mettre en place l'industrialisation du logiciel.

Praxeme prévoit une action cruciale dite « négociation logique / technique », à l'occasion de laquelle les architectes logiques et techniques s'assurent de la convertibilité de l'expression logique en logiciel. Beaucoup des règles de dérivation entre les modèles sont mécaniques et peuvent s'automatiser.

Conclusion

Pour tirer tous les fruits attendus du style d'architecture SOA, il est nécessaire de revisiter, de fond en comble, la chaîne d'activité qui va des objectifs et du métier jusqu'au déploiement. L'architecture technique fixe les conditions de faisabilité de l'architecture de services. L'architecture logique élabore la structure du système informatique, en reprenant les domaines fonctionnels mais en s'inspirant aussi des domaines d'objets, trouvés dans le modèle sémantique. Il en résulte une architecture radicalement neuve et optimisée.

La maîtrise de ces niveaux de préoccupation suppose une méthode complète. Praxeme offre une telle méthode, avec l'avantage pratique et économique de l'*open source* : une méthode de référence, largement partagée et facile d'accès. Le cas du grand projet de la SMABTP est une parfaite démonstration de l'apport de la méthode, quand elle est soutenue par le management. 50% du code était généré automatiquement à partir d'un modèle logique très précis où les services logiques étaient spécifiés en pseudo-langage. Le taux de réutilisation moyen, dès la fin du premier projet, s'élevait à 3.

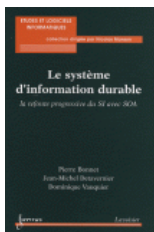
Dominique VAUQUIER - <mailto:dvau@praxeme.org>

Bibliographie

Pour plus d'information et pour obtenir les guides méthodologiques : <http://www.praxeme.org>

Les procédés de conception et d'architecture logiques de la méthode Praxeme pour SOA sont enseignés dans le cours « SOA, conception d'une architecture de services » (voir <http://www.praxademia.com/formation/>).

L'expérience de la refonte du système d'information menée à la SMABTP en appliquant Praxeme a été décrite dans les ouvrages suivants :



« Le Système d'information durable, La refonte progressive du SI avec SOA », Pierre Bonnet, Jean-Michel Detavernier, Dominique Vauquier, Hermès Lavoisier, 2007

Sur ce grand projet, Jean-Michel Detavernier était le DSI adjoint, Pierre Bonnet le responsable de la Cellule d'architecture technique, Dominique Vauquier l'architecte logique et le méthodologue. La méthode a donné lieu à d'autres applications, dans des secteurs variés.



« Sustainable IT Architecture, The progressive way of overhauling information systems with SOA », Pierre Bonnet, Jean-Michel Detavernier, Dominique Vauquier, Jérôme Boyer, Erik Steinholtz, Wiley, 2009

Au fil du temps, la méthode s'est enrichie et consolidée en une version 2, diffusée par la société Praxademia, contributrice à la méthode publique.

Formation

La méthode Praxeme pour SOA est la partie de la méthode publique dédiée aux procédés de conception et d'architecture dans le style SOA.

La formation « SOA, conception d'une architecture » développe les compétences nécessaires pour identifier et spécifier les services, pour les structurer de façon optimale et pour préparer leur conversion en logiciel. Elle traite également les tests, le déploiement et l'exécution des services.

La figure ci-contre donne le contenu, organisé en séquences pédagogiques d'une demi-heure à trois quarts d'heure, conformément à la méthode de la pédagogie par objectifs.

Figure SLB-15_5. Le sommaire de la formation « SOA, conception d'une architecture logique »

La fiche descriptive complète est disponible sur :

<http://www.praxademia.com/formation/soa-conception-et-architecture-dun-systeme-de-services/>

