

# Comment rendre portables les services en ligne ?

Philippe PRADOS

[pp@philippe.prados.name](mailto:pp@philippe.prados.name)

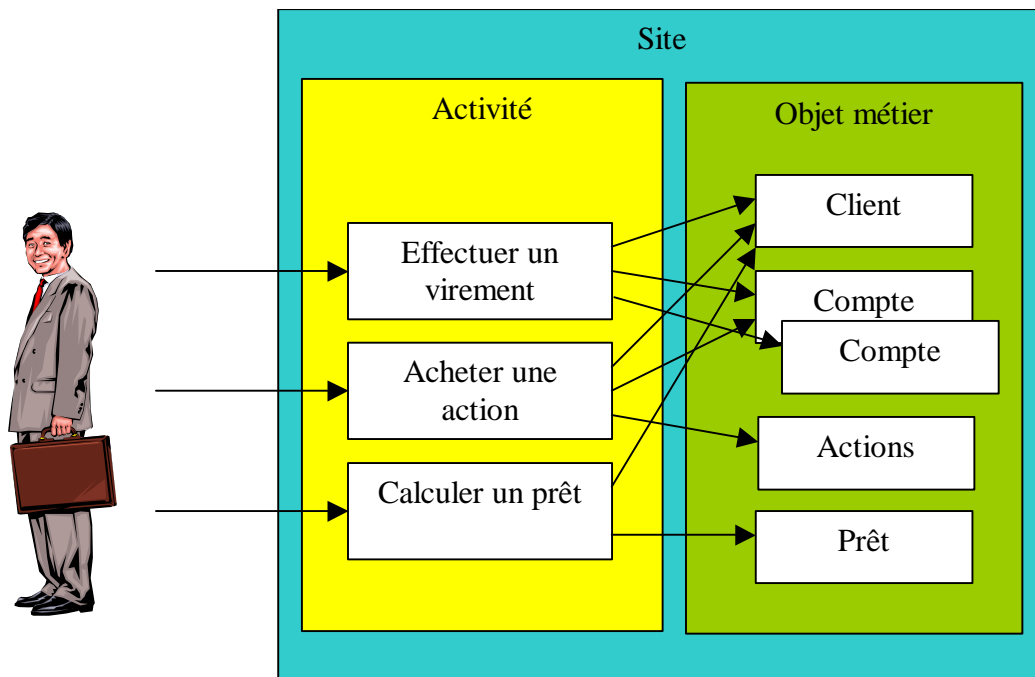
En 2005, la majorité des internautes utiliseront un autre périphérique qu'un ordinateur pour naviguer sur le réseau. D'autres technologies que les ordinateurs personnels permettent de surfer. De plus en plus, les sites devront être capable d'intégrer différents médias : les téléphones cellulaires, les PDA ou plus simplement les téléphones classiques. Cela exige une structuration très en amont des développements. Il faut s'interroger sur l'organisation des développements à mettre en place pour permettre cette diversité.

Afin de retrouver l'essence de la navigation sur Internet, interrogeons-nous sur ce qui caractérise les ergonomies des navigateurs par rapports aux ergonomies classiques. La navigation sur le net s'appuie sur le protocole HTTP qui permet d'obtenir des documents. Les navigateurs comme Internet Explorer de Microsoft ou Communicator de Netscape proposent la simulation d'un livre. Chaque site représente un livre. L'utilisateur visualise des pages. Il est libre de naviguer d'une page à une autre, d'en mémoriser certaines, de sauter des passages, etc. La particularité de la navigation sur Internet par rapport au livre classique est la possibilité d'avoir un livre sur mesure, dont les pages évoluent suivants les chemins utilisés ou les informations renseignées. Un site est le regroupement de page dans un livre.

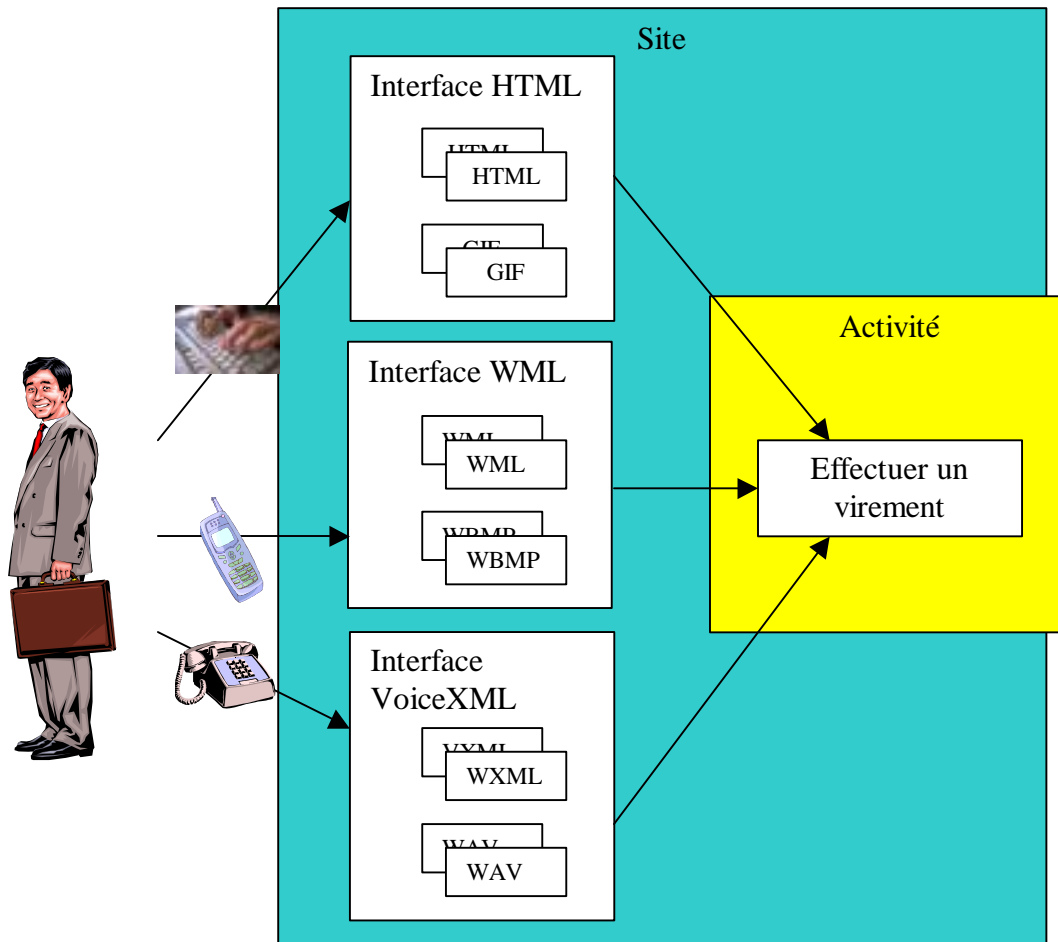
Il existe une structuration supplémentaire dans les livres. En effet, ils sont séparés en chapitre. Un chapitre est un ensemble de pages permettant d'offrir un service. Généralement, une page de formulaire et son résultat constitue un chapitre. Les chapitres peuvent être plus volumineux et nécessiter plusieurs pages. Comme pour les livres-jeux, un chapitre peut offrir plusieurs chemins. Les chapitres sont autonomes les uns des autres. Il est facile d'ajouter, d'en supprimer ou d'en modifier, sans remettre en cause l'ensemble du site. La fonctionnalité offerte par un chapitre est indépendante des pages qui le composent. La notion d'application, que l'on connaît pour les programmes traditionnels, n'existe plus sur Internet. Un site et un ensemble de service.

Le sens décrit dans un chapitre peut être exprimé par différentes pages, dans différentes langues, avec des textes plus ou moins complexes, plus ou moins illustré. Le choix et le design des pages ne sont qu'une des expressions possibles d'un chapitre. Cela n'a pas d'importance pour le contenu du chapitre lui-même. Il existe des publications simplifiées de livre de grand auteur. Certains chapitres sont résumés en une ligne.

Une modélisation multimédia d'un site Internet doit intégrer cette notion de chapitre. Pour utiliser un terme plus proche des développeurs, un chapitre correspond à une *activité* disponible sur un site. Les activités sont indépendantes les unes des autres. Elles permettent à l'utilisateur de demander au serveur de mémoriser et de traiter des informations ou de retrouver des documents. Une activité commence généralement par un verbe. On peut citer comme exemple : « Créer un compte », « Acheter une action », « Effectuer un virement », ... Chacune de ces activités est indépendante l'une de l'autre. En attendant la prise en compte d'un ordre d'achat d'action, l'utilisateur peut effectuer un virement entre ces comptes.



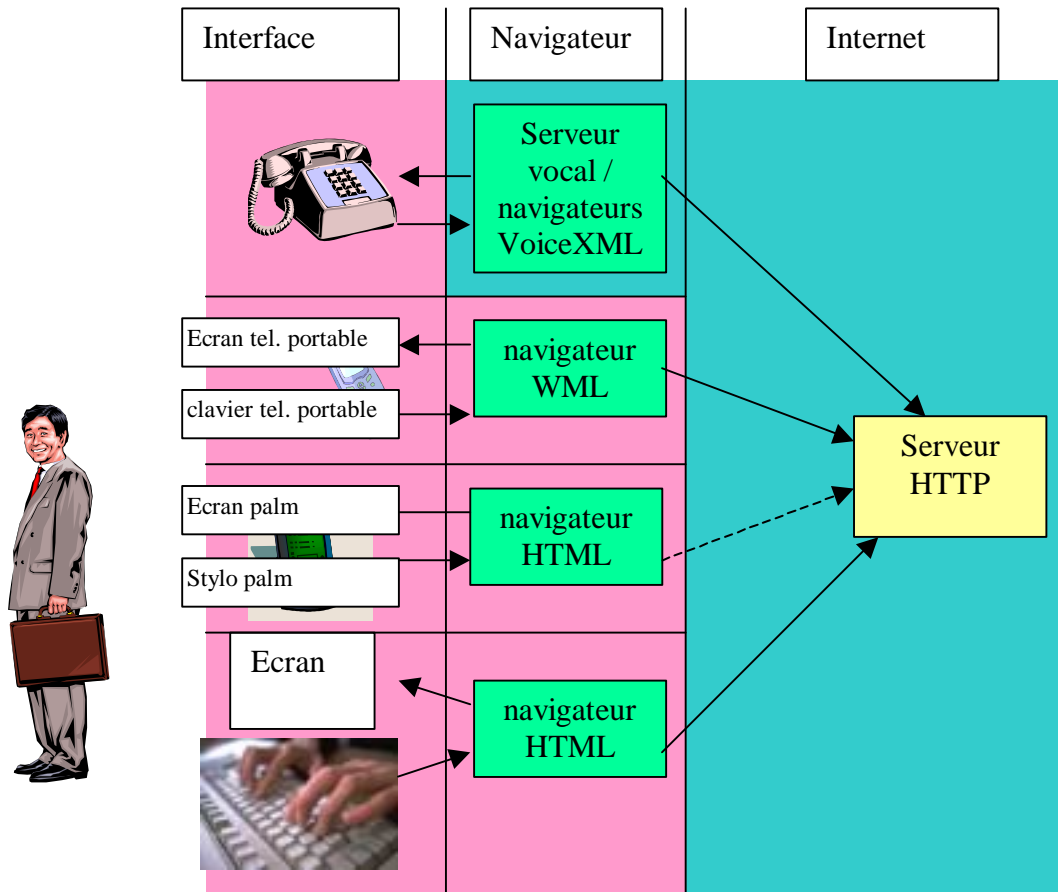
Pour offrir cette vision d'un site Internet, un objet sur le serveur doit être créé pour supporter chaque activité. L'objet de l'activité communique avec le système de l'entreprise, récupère et valorise des objets métiers, invoque les transactions, ... Il doit être indépendant de l'ergonomie ou de la navigation. Son rôle est de proposer un usage particulier du métier de l'entreprise. Par exemple, un virement n'est pas un service d'un compte en banque, mais une façon de l'utiliser. L'internaute doit sélectionner le compte à débiter et le compte à créditer. Le virement s'effectue en invoquant deux traitements différents : le débit du premier compte et de crédit du second. L'ensemble du traitement doit s'exécuter ou pas du tout. Il n'est pas question d'interrompre la procédure au milieu. L'activité permet d'encadrer les différentes transactions de l'application.



Chaque canal utilisera une ergonomie qui lui est propre pour diffuser l'activité. Les informations présentées sur une page HTML peuvent être très nombreuses. Sur un téléphone WAP, ce n'est pas possible. Certaines informations ne seront pas diffusées, d'autres seront organisés différemment, sur plusieurs pages par exemple. Pour un téléphone classique, un moteur de synthèse vocale diffusera une analyse des informations extraites de l'activité.

La simulation d'un livre, imposé par le protocole HTTP, présente un inconvénient important : lors de l'affichage d'une nouvelle page, les informations renseignées dans les pages précédentes sont immédiatement perdues. Seules les informations à n-1 sont disponibles. Les navigateurs imposent une architecture « Client-Léger ». Le client ne mémorise aucune information lors de sa navigation. Si un traitement est demandé à la troisième page, le serveur doit se rappeler des informations entrées dans la première. L'objet représentant l'activité va s'occuper de cela. Toutes les informations nécessaires aux traitements seront mémorisées dans cet objet. Au fur et à mesure de la navigation de l'utilisateur, les différentes informations seront agrégées dans cet objet. De même, tous les résultats seront également mémorisés afin de ne pas dépendre de la présentation. En effet, un résultat complexe peut être diffusé sur une ou plusieurs pages. Le résultat du calcul doit être rémanent.

Regardons maintenant ce qui caractérise les différents périphériques permettant de naviguer sur le réseau. Il y a trois couches distinctes : Interface, navigateur et Internet (Voir schéma).



La première couche est en contact direct avec l'utilisateur. Elle propose des interfaces différentes étant en relation directe avec les sens de l'utilisateur. Cela peut être un clavier, un écran, un micro, un écran tactile, un haut-parleur, un retour d'effort, ou toutes technologies permettant de communiquer avec un être humain. La deuxième couche s'occupe de faire l'interface entre les interfaces et le serveur http. C'est dans cette couche que l'on retrouve les navigateurs Internet ou WAP, mais également les serveurs VoiceXML. VoiceXML est un format permettant de décrire une navigation à l'aide d'un téléphone classique (Voir encart). La troisième couche est simplement le serveur http présent sur le réseau. Un serveur propose différents fichiers interprétables est analysable par la deuxième couche. La localisation de chacune des couches est plus ou moins proche de l'utilisateur. Cela est représenté par des couleurs de fond différentes. Les terminaux possèdent généralement un navigateur, mais ce n'est pas le cas du téléphone classique.

Suivant le canal utilisé, les traitements sont différents. Par exemple, avec un ordinateur personnel, lorsque l'utilisateur utilise des touches de son clavier, elles sont mémorisées par le navigateur jusqu'à la détection de la touche Return ou l'appui du bouton de soumission. Alors, le navigateur construit une requête http et l'envoie au serveur. Celui-ci analyse la demande et construit un fichier pour décrire l'ergonomie que devra proposer le navigateur. Le fichier est analysé et présenté à l'utilisateur, ce qu'il constate sur son écran.

Pour une utilisation avec un téléphone classique, l'utilisateur appelle un numéro de téléphone particulier. Celui-ci est géré par un serveur vocal. Lors de la réception de l'appel, le serveur demande une page particulière au serveur http. Celle-ci a été paramétrée dans le serveur vocal, en association avec le numéro de téléphone. Le serveur http retourne une page au format VoiceXML. Le serveur vocal analyse ce fichier et diffuse le message décrit dans le fichier. Il utilise pour cela un moteur TTS (Text-To-Speech) ou diffuse tout simplement un message pré-enregistré. Il attend alors une intervention de l'utilisateur. Lorsque celui-ci utilise les touches de son téléphone, ou qu'il prononce un mot du dictionnaire, le serveur vocal le détecte et vérifie s'il fait bien partie du vocabulaire proposé par le serveur http. Si l'information est correcte, le serveur vocal diffuse un autre message. Cela continue tant que toutes les informations d'un formulaire ne sont pas correctement renseignées. Lorsque tout est prêt, le serveur vocal construit une requête http et la soumet au serveur http. Celui-ci l'analyse et calcule le nouveau scénario vocal à diffuser à l'utilisateur.

Ces deux descriptions sont très différentes, mais sémantiquement similaires.

Il existe des outils pour convertir un format vers un autre. Par exemple, un téléphone WAP peut consulter une page HTML. Le résultat est souvent décevant. Un site multi-canal ne peut généralement pas utiliser cette approche. Les ergonomies doivent être adaptées à chaque média. Chacun à ces contraintes. Une traduction automatique ne peut y répondre. Par exemple, il est difficile de demander l'entrée d'un texte important sur un téléphone WAP. De même, il est difficile de diffuser vocalement un tableau de chiffre avec un téléphone classique. Le mariage de WML et de VoiceXML permettra de réduire ces contraintes.

La constitution d'un objet informatique par activité permet d'adapter facilement une ergonomie pour tenir compte des différents médias. À partir d'une application constituée d'un ensemble d'activité, rédigez des fichiers HTML, WML ou VoiceXML et ajoutez le code nécessaire pour interfacier le protocole http à l'activité.

Le framework e-Activity d'IBM Global Service propose cette architecture. Un simple fichier XML permet de décrire l'interaction en les médias et les objets d'activités. Il n'est plus nécessaire de rédiger des servlets.

Cette approche, prise très en amont, permet d'ajouter rapidement de nouveaux canaux en bénéficiant des développements précédents. Par exemple, un site de gestion de portefeuille HTML peut migrer facilement vers une ergonomie WML et VoiceXML. Les fonctionnalités offertes ne seront pas toujours exhaustives, mais l'utilisateur pourra consulter, pratiquement n'importe où, la cotation de ses actions.

Il est alors possible d'imaginer l'intégration d'autres canaux. Par exemple, un applet java ou un composant ActiveX peuvent offrir une ergonomie très riche, non orienté page, mais fenêtre. La communication avec l'activité s'effectue toujours avec le protocole HTTP. Une autre approche amusante : des formulaires pré-imprimés sont distribués aux utilisateurs. Lorsqu'ils désirent effectuer un ordre de bourse, ils remplissent des cases sur un formulaire et l'envoient par fax à un serveur qui se charge de traduire la demande en requête HTTP. C'est toujours la même activité, le même objet qui est utilisé.

Il est important d'anticiper sur les évolutions prévisibles des périphériques. Avec un peu de rigueur dans la conception et dans la programmation, il est facile d'intégrer de nouveau média. Le développeur offre des activités les plus indépendantes possibles du média ; des ergonomes spécialisés pour chaque média conçoivent alors les ergonomies à proposer et rédigent un fichier XML. Ils peuvent supprimer des fonctionnalités, en simplifier ou en synthétiser. Les designers peuvent alors rédiger les pages HTML, WML ou VoiceXML.

Liens :

W3C : [www.w3c.org](http://www.w3c.org)

VoiceXML : [www.voicexml.org](http://www.voicexml.org)

WML : [www.allnetdevices.com/faq/](http://www.allnetdevices.com/faq/)

Divers : [www.philippe.prados.net](http://www.philippe.prados.net)

VoiceXML est un format de fichier permettant de décrire en XML une ergonomie pour un téléphone vocal classique. Les navigateurs VoiceXML doivent être capables de diffuser des messages sonores ; posséder un moteur TTS (Text to Speech) pour traduire une information textuelle en un message vocal et détecter les touches multifréquences du téléphone et/ou reconnaître des mots d'un vocabulaire.

```
<form id="meteo">
  <block>Bienvenu sur le service de météo.</block>
  <field name="departement">
    <prompt>Quel département ?</prompt>
    <grammar src="departement.gram" type="application/x-jsgf"/>
    <catch event="aide">
      Indiquez distinctement le nom d'un département.
    </catch>
  </field>
  <field name="ville">
    <prompt>Quelle villes ?</prompt>
    <grammar src="ville.gram" type="application/x-jsgf"/>
    <catch event="aide">
      Indiquez une ville du département.
    </catch>
  </field>
  <block>
    <submit next="meteo.jsp" namelist="departement ville"/>
  </block>
</form>
```

Ce formulaire demande à l'utilisateur d'indiquer le département et la ville où il désire obtenir la météo. Le marqueur `<grammar>` permet d'indiquer un fichier contenant l'ensemble des mots du vocabulaire concerné par le champ.

Le format VoiceXML accepte des scripts au format JavaScript. Cela permet d'effectuer le maximum de traitement sur le serveur vocal, avant d'invoquer le serveur http.