

## Chapitre 5. Traces et publication \*

*The owner of the memex, let us say, is interested in the origin and properties of the bow and arrow. [...] First he runs through an encyclopedia, finds an interesting but sketchy article, leaves it projected. Next, in a history, he finds another pertinent item, and ties the two together. Thus he goes, building a trail of many items. Occasionally he inserts a comment of his own, either linking it into the main trail or joining it by a side trail to a particular item. When it becomes evident that the elastic properties of available materials had a great deal to do with the bow, he branches off on a side trail which takes him through textbooks on elasticity and tables of physical constants. He inserts a page of longhand analysis of his own. Thus he builds a trail of his interest through the maze of materials available to him. And his trails do not fade. Several years later, his talk with a friend turns to the queer ways in which a people resist innovations, even of vital interest. He has an example, in the fact that the outraged Europeans still failed to adopt the Turkish bow. In fact he has a trail on it. [...] It is an interesting trail, pertinent to the discussion. So he sets a reproducer in action, photographs the whole trail out, and passes it to his friend for insertion in his own memex, there to be linked into the more general trail.*

Vannevar Bush, *As we may think*, 7.

### 1. Notion de trace

#### a. Principe

Le sens d'un document n'est pas dans sa structure (quelle qu'elle soit) mais dans les conditions de son utilisation et donc dans les conditions de sa lecture. Autrement dit, il est impossible de trouver *le* sens d'un document mais plutôt *des* sens. Toute lecture est

---

\* Ce chapitre fait l'objet d'un article de recherche à paraître dans les Suppléments au Bulletin de Correspondance Hellénique [Benel03].

Les notions de "trace" et de "publication" ont été par ailleurs introduites lors d'une conférence à la "European Conference on Digital Libraries" [BenelEtAl01a] et dans un article de vulgarisation [BenelEtAl01b].

donc une réécriture. Un système adéquat devrait permettre de garder trace des parcours d'interprétation des lecteurs. Vannevar Bush, visionnaire de ce qui allait devenir l'hyper-texte, décrivait d'ailleurs en 1945 un modèle basé sur des traces (ou pistes) de pensée [Bush45]. Cette métaphore de la « trace » est particulièrement riche et mérite que l'on s'y attarde un peu.

Tout d'abord, une trace, c'est la forme qui demeure quand le contenu a disparu. En effet, comme nous l'avons vu, l'ordinateur ne traite que de symboles dénués de signification. C'est à l'utilisateur qu'incombe la difficile tâche de donner un contenu<sup>48</sup> à ce qui n'est que formel. Ensuite, une trace est rarement seule, elle fait partie d'une piste. On retrouve ici la notion de contexte, c'est-à-dire d'autres traces reliées à elle. Enfin, on peut « suivre quelqu'un à la trace » mais aussi « suivre les traces de quelqu'un ». Dit autrement, il existe deux approches complémentaires : interpréter les traces d'un autre selon notre point de vue, ou tenter par la pensée de se mettre « à sa place ». On retrouve ici la notion de situation.

Si les symboles en contexte peuvent être « rentrés » dans une machine, il n'en est pas de même pour les situations. Il est impensable de faire rentrer dans un ordinateur des personnes, des lieux, des cultures, bref le monde. Aussi nous considérerons plutôt des « indices de situation », c'est-à-dire des informations sur les différentes étapes de la « vie » d'un symbole contextualisé (date et auteur de la création d'un contexte, etc.).

Ainsi, notre approche théorique nous permet de proposer le patron de la « Trace » (cf. Figure 5.1). Pour un symbole donné, moyennant une heuristique (ex : locale, globale...), il est possible d'obtenir un contexte pointant vers un certain nombre d'autres symboles. Ce contexte est associé à des indices de situation. Notons qu'une trace peut être le résultat de l'agrégation d'autres traces, l'agrégation étant une mise en contexte particulière.

---

<sup>48</sup> Au sens philosophique.

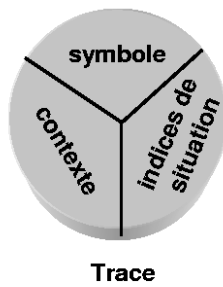


Figure 5.1 : Notion de trace

Dans notre prototype (appelé *Porphyre*) ces traces (cf. Figure 5.2) peuvent être des documents : fragments, sources ou notes. Elles peuvent également être des corpus de documents, des descriptions semi-formelles de ces corpus (à l'aide de graphes acycliques de descripteurs), ou encore des étapes de lecture. Les descriptions sont mises en contexte entre-elles suivant les relations de généralisation/spécialisation qui existent entre leurs graphes respectifs. Les étapes de lecture, pour leur part, sont mises en contexte par les relations de séquence indiquées dans des parcours de lecture.

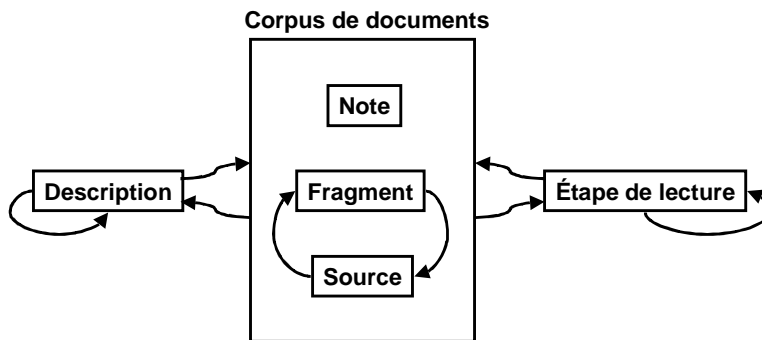


Figure 5.2 : Types de traces et liens contextuels

### b. Architecture

Pour la gestion des documents et des corpus, nous utilisons un *serveur de contenu* (cf. Figure 5.3). Celui-ci stocke et diffuse les documents sources (images ou textes). Il se distingue d'un serveur Web normal en permettant également d'extraire des fragments des documents sources, de « surligner » tous les fragments associés à un document source, ou encore, dans le cas où les documents sont des images, d'obtenir des

zooms ou des vignettes. Les documents sources sont placés grâce à un serveur FTP (en mode passif pour passer à travers les pare-feux).

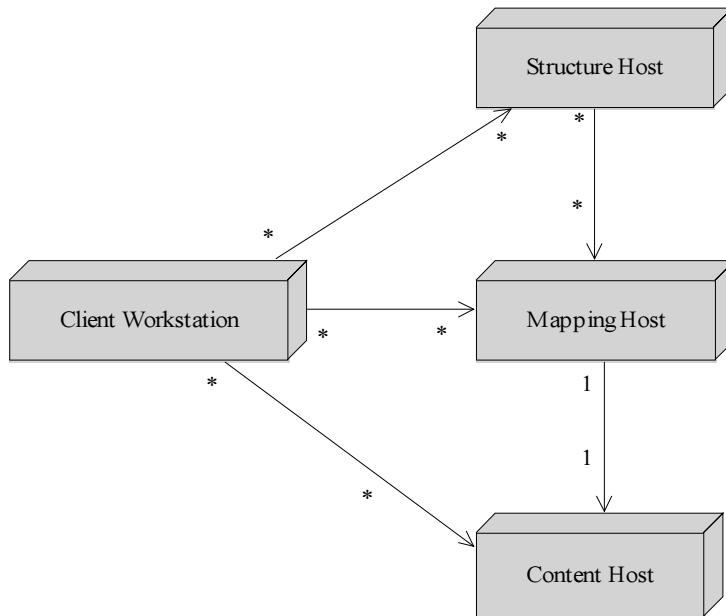


Figure 5.3 : Architecture multi-tiers du système *Porphyre* pour un accès natif (diagramme de déploiement UML)

Pour la gestion des traces semi-formelles, nous faisons appel à un *serveur de structure* (cf. Figure 5.3). Celui-ci stocke ces traces dans une base de données, et permet de les présenter dans un contexte susceptible d'intéresser le lecteur.

Les traces peuvent être consultées soit par un *client web* par le biais d'un *serveur passerelle* (cf. Figure 5.4) soit par le *client natif* (cf. Figure 5.3). Ce dernier destiné au lecteur expert lui permet d'effectuer une lecture-écriture des traces suivant la métaphore de l'annotation. Nous nous plaçons ainsi dans la prolifique descendance des travaux de Jacques Virbel et Bernard Stiegler autour des « Postes de Lecture Assistée par Ordinateur » [Stiegler00].

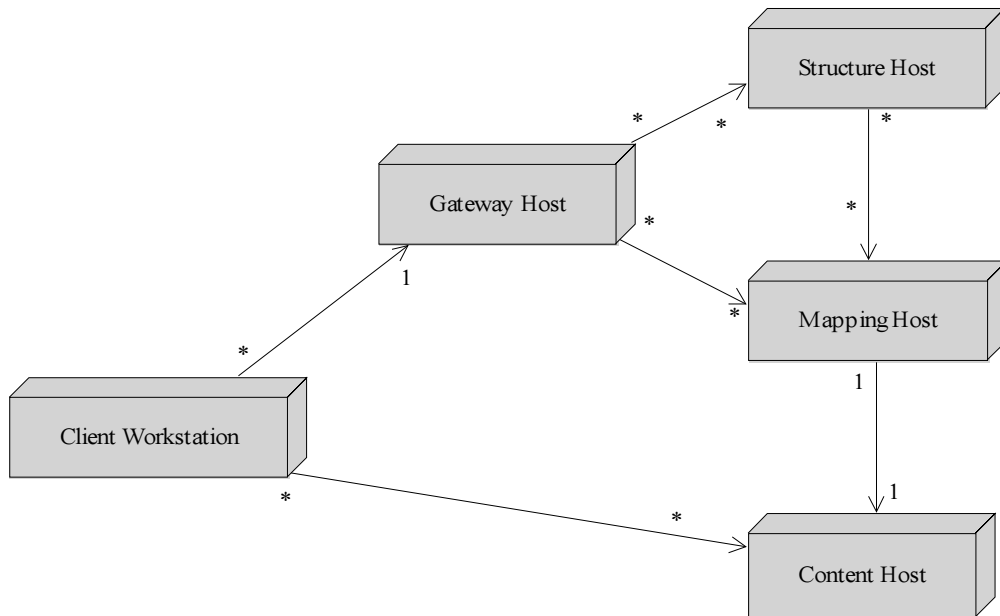


Figure 5.4 : Architecture multi-tiers du système *Porphyre* pour un accès par le Web (diagramme de déploiement UML).

Si nous n'avons pas parlé jusqu'à présent du *serveur de correspondance*, c'est que sa fonction est plus délicate à saisir : il construit l'URL<sup>49</sup> permettant d'obtenir un contenu. Son but est donc d'éviter une redondance des données (stocker à la fois les paramètres du contenu et l'URL qui en dépend).

Le fait que le serveur de correspondance soit distinct du serveur de contenu permet d'interfacer d'autres types de serveurs de contenu : versions antérieures de *Porphyre*, serveur *Transvision*<sup>50</sup>, Serveurs Web classique. Lorsqu'un de ces types de serveurs ne prend pas en charge une fonction donnée, la requête est alors simplifiée<sup>51</sup>.

<sup>49</sup> *Uniform Resource Locator* [URL] : Adresse permettant de référencer toute ressource (statique ou dynamique) accessible sur Internet (entre autres sur le Web). Ex : [http://porphyry.porphyry.org/image/getSource?id=BCH\\_120\\_1\\_492&maxWidth=800](http://porphyry.porphyry.org/image/getSource?id=BCH_120_1_492&maxWidth=800)

<sup>50</sup> *Transvision*® : Système de gestion de bases d'images développé par l'équipe « images » de la Maison de l'Orient et de la Méditerranéen puis commercialisé pendant quelques années par la société « SGBI Entreprise SA » [TVML00]. Il s'agit du système utilisé pour la partie « contenu » du projet CEFAEL [Benel02b].

En étant distinct du serveur de contenu, le serveur de correspondance permet d'utiliser une adresse plus abstraite (comparable aux URN<sup>52</sup>) utilisable pour la comparaison de deux références provenant de deux serveurs de structure différents. Il devient alors possible d'associer à un serveur de contenu, plusieurs serveurs de structure, tout comme un serveur de structure peut l'être à plusieurs serveurs de contenu. Ainsi, plusieurs équipes de recherche peuvent travailler sur un même corpus, et réciproquement une équipe peut travailler sur plusieurs corpus à la fois.

## 2. Notions de publication, de communauté et d'édition

Il ne suffit pas de stocker des traces, encore faut-il les partager... La métaphore que nous choisirons est celle de la *publication* (cf. Figure 5.5). Cette « mise en public » est basée sur une relation de confiance concrétisée par une communauté. L'adhésion d'un expert à une communauté nécessite l'accord commun de l'expert et du représentant de la communauté. Membre de la communauté, l'expert peut lire les documents et les traces publiés par les autres membres. Dans une bibliothèque privée, il peut ajouter ses propres traces d'interprétation relatives à ses propres documents ou aux documents de la communauté.

---

<sup>51</sup> Par exemple, si l'on demande l'URL permettant d'extraire un fragment d'image avec redimensionnement sur un serveur Transvision®, l'URL construite tiendra compte de l'ordre de redimensionnement mais pas de celui d'extraction de fragment.

<sup>52</sup> *Uniform Resource Name* [URN] : Identifiant permanent d'une ressource sur Internet, indépendamment de sa localisation physique. Ex : « urn:hdl:cnri.dlib/august95 », « urn:doi:10.1000/1 », « urn:issn:1082-9873 »...

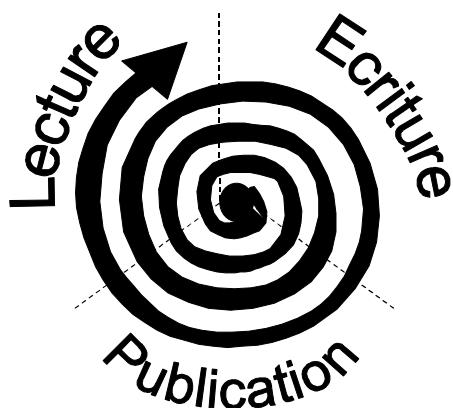


Figure 5.5 : La publication comme moteur de l'interprétation

Si l'expert le souhaite, il peut soumettre au représentant de la communauté une exportation totale ou partielle de ses traces et documents personnels pour les publier. Le responsable de la communauté peut alors choisir de les publier ou non, éventuellement en y apportant des modifications consenties par l'auteur.

La figure du « représentant de la communauté » permet de gagner en généralité. Il incombe ensuite à chaque communauté de décider quelle organisation elle mettra derrière : anarchie, consensus, majorité, comité de lecture, ou dictature !

Dans un souci d'ouverture du système, la soumission d'un ensemble de traces se fait en dehors du système (par l'intermédiaire d'un courriel par exemple). Les traces sont exportées par leur auteur dans un fichier XML, le fichier est transmis au représentant de la communauté qui les importe ensuite dans une zone brouillon où il peut les modifier. Le représentant peut enfin publier (au sens propre) les traces, c'est-à-dire les transférer dans la zone publique de la communauté.

### 3. Histoire d'une trace

Nous venons de voir de manière informelle les processus qui affectaient dans le temps un ensemble donné de trace. Notre but serait maintenant de capter l'historique de ces transformations.

Pour ce faire, nous devons tout d'abord conserver l'ensemble des symboles. Seules les relations contextuelles pourront être modifiées. Elles le seront par les actions suivantes : *création, suppression, archivage, purge*<sup>53</sup>.

Ces actions feront passer les relations contextuelles par quatre états (cf. Figure 5.6) :

- *Brouillon* : présent dans l'espace du scripteur, mais pas encore dans celui du lecteur,
- *Publié* : présent dans les deux espaces,
- *Obsolète* : encore présent dans l'espace du lecteur, mais plus dans celui du scripteur,
- *Au pilon* : absent des deux espaces.

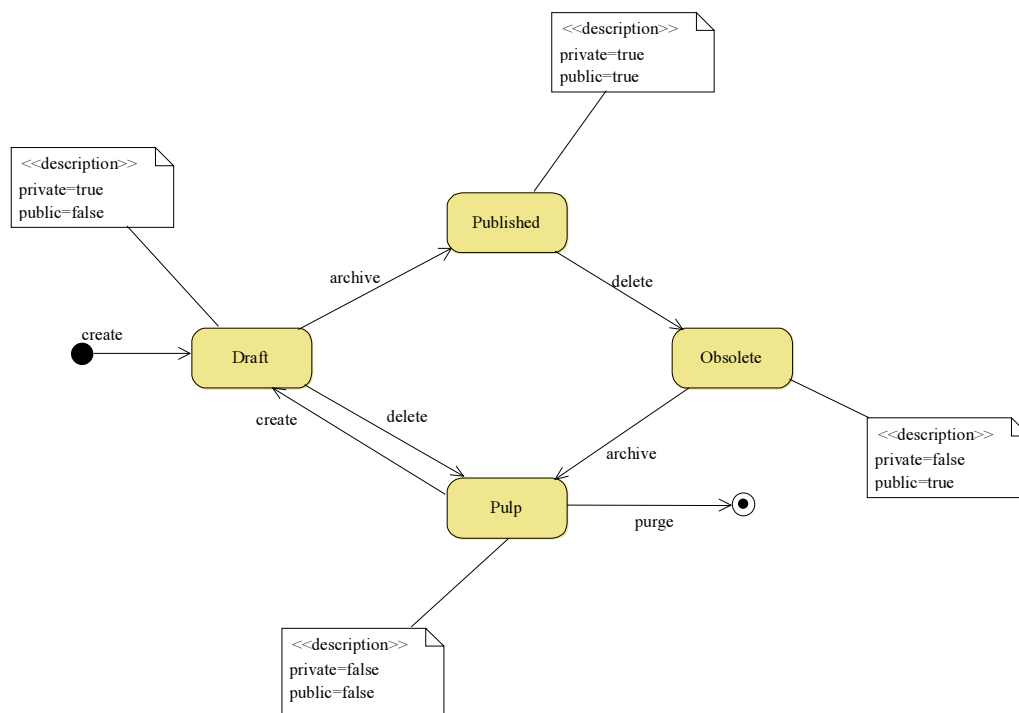


Figure 5.6 : « Cycle de vie » d'un contexte (diagramme d'état UML).

Notons que ce cycle de vie est basé sur la théorie de la connaissance exposée dans le chapitre 2. Ainsi, le parcours menant de l'état initial au *pilon* en passant par le

<sup>53</sup> Nous nous inspirons ici des serveurs de gestion de version (Ex : *Concurrent Version Server, Ms Visual Source Safe, etc.*).



## CHAPITRE 5. TRACES ET PUBLICATION\*

*brouillon*, la *publication* et l'*obsolescence* reprend la théorie poppérienne de l'hypothèse : sa construction par un chercheur puis sa publication par la communauté, sa remise en question par un autre chercheur, et la publication de cette remise en question (réfutation). De même, la transition menant du *pilon* au *brouillon* reprend la théorie kühnienne selon laquelle les « acquis » scientifiques (donc les réfutations) peuvent être remis en question par une révolution des paradigmes. A ces transitions issues de la philosophie des sciences, nous en avons ajouté deux. Celle menant du *brouillon* au *pilon* rend compte de l'avortement de certaines hypothèses avant leur publication. Celle menant du *pilon* à l'état final (et donc à la destruction réelle) rend compte de la pratique des archivistes qui pour mieux conserver certaines choses doivent parfois en jeter d'autres.

Pour que l'historique soit complet, nous devons maintenant conserver l'ensemble des actions effectuées sur les relations contextuelles. Ainsi à chacune de ces relations sera associé un ensemble de situations<sup>54</sup> définies par trois questions :

- *Quoi* : nature de l'action (*création, suppression, archivage*),
- *Quand* : date et heure de l'action,
- *Qui* : nom et prénom du chercheur ayant effectué l'action.

Le modèle obtenu rend possible dans un premier temps certaines fonctionnalités simples mais intéressantes comme :

- la définition de deux espaces, un pour le scripteur et un pour le lecteur,
- l'obtention de l'historique d'une relation contextuelle,
- la possibilité d'annuler une suite erronée d'opérations (*undo/redo*).

Cependant, à terme, le but est de visualiser de manière synthétique l'activité dans le temps d'une communauté. Cet aspect sera développé dans le chapitre 12 comme perspective à nos travaux.

---

<sup>54</sup> Ou pour être plus précis d'*indices de situation*, puisque comme nous l'avons vu une situation ne peut être stockée dans un ordinateur.

## CONSULTATION ASSISTÉE PAR ORDINATEUR DE LA DOCUMENTATION...

Dans les chapitres suivants, nous détaillerons trois types de traces particuliers :

- les contenus documentaires,
- les réseaux de description,
- les parcours de lecture.