POT Prototype Systemの実装(2) など*

伊戸川暁†

1999 年 9 月 14 日

今日は主に、自分がどういう勘違いをしていたことに気づいたかについて話そうと思います。

0 マクラ: Xanaduのソース

1999年8月27日、かのXanaduのソースが部分的ながら公開されたので(参考資料1)、取ってきて[1]少しいじってみた。

当初筆者は、

- 独自プロトコルにこだわりすぎているようにも見える、とか
- デモがついていたようだが、networkにつなぐデモをしてほしかった

のレベルで批評しようと思ったのだが、きちんと動かないようなので少し中身を覗いてみたところ、「構想40年の末に公開」したはずのソースのあまりの完成度の低さに目を覆ってしまったのだった。例えば、

- まともなconfig fileがない
- 絶対パスがハードコーディングされている
- もちろんdocumentationも適当

などなど。

ちなみに、最もひどいと思った1行は、

```
system("cat /usr3/xu/requests.j");
```

である[1]。

ひどい感想になるが、どうして世間でXanaduがあまりまともに扱われないのか、少し分かった気がした。

1 コンセプトの再確認 – (無謀にも)Xanaduと比較して

そもそもなぜXanaduのソースなどをチェックしようと思ったかだが、それは、目的が比較的似ているからである。

---

*KTYYゼミ発表資料

†川合研M2

[1] おまけに配布物中には、'requests.j' という名前のファイルが全く存在しない。

1.1 共通点

- 電子情報の最終的かつ恒久的な保管場所
- グローバルかつ恒久的な命名体系

1.2 向こうにあってこっちにないもの

- 著作権管理
- ハイパーテキスト作成支援 (自動的リンク作成とか)

1.3 向こうになくてこっちにあるもの

- ストレージのレベルまで踏み入った管理
- 全リソースの一元的取扱い
- 管理計画を明示できる
- むしろ全体的枠組を提供 (要素技術はあるものを利用する)

2 実装した点

現在作成したコードの分量は1300行程度[2]。
最近したことは:

- User Client/Server モドキ (図2中の破線Cの辺りに相当) の製作
- プロトコル4及び5(図2参照) の製作

しかし上記作業中、以下のようなことどもに次々と気づき、相当部分がナンセンスであると分かったところで本日の発表を迎えた。

3 僅かながらも手を動かしてみて気づいたことども

今日の本題はむしろここです。なんだか当たり前のことばかり言っているようですが、本人にとっては新鮮な発見だったので勘弁して下さい。

3.1 層と文書の対応関係

文書の姿は、「層」ではなく、「層の間」に対応する (図1)。層同士が「文書」をやりとりするのだと考えれば当然そういうことになる。
層の中にある「文書」は、バイト列というよりはむしろObjectとして存在しており、それが層の上面と下面とで異なった表現を取ると考えた方がよい。

---

[2] 思いついた順番に、極めて場当たり的に開発しています。

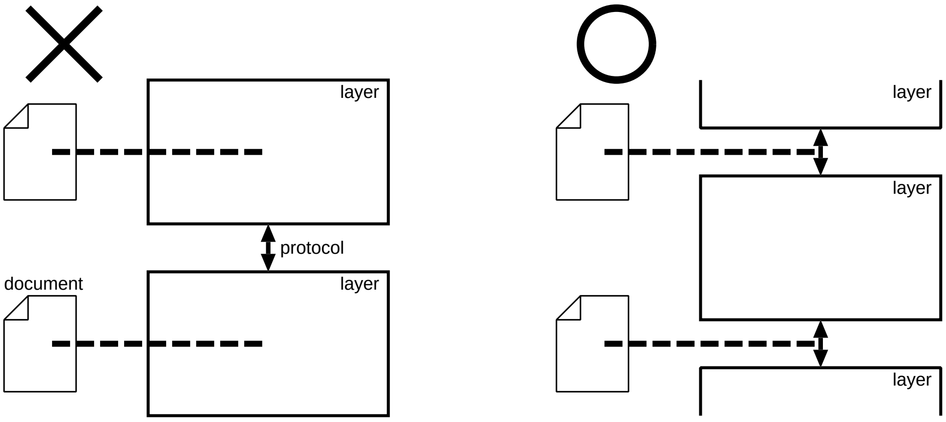

図 1: 層と文書の対応の正誤

3.2　層の切り分け図最新版

実装で試行錯誤中、層の切り分けにも仕様変更が生じた (図 2 を参照)。

図中の矢印は情報 (=文書) のやりとりがある、すなわちプロトコルが存在することを示し、破線は、その両側のやりとりがクライアント/サーバを介することを表している。

3.3　各々のプロトコルにおいて、「文書」はどのように表現されるべきか?

前回の見本 xml(参考資料 2) は、図 2 の番号でいえば「4」でやりとりされるべき文書を考えていたものの、

- システム全体を通じて用いられる XML の形式が一通りでは済まないこと
- クライアント/サーバが複数種類存在すべきこと

の両者によく気づいていなかったために、中途半端なものになってしまった。

従って、各箇所のプロトコル用の「文書」の定義を決めなければならないのだが、現在策定中。

4　やるべきこと

- 各部分のプロトコルの定義及び実装
- 最終的には test case としての協力文書募集[3]

参考文献

[1] roger@xanadu.com: Udanax Green, `http://www.udanax.com/green/`.

---

[3] 当初はこの発表で、ゼミとかの資料を POT に入れたい旨皆様にご協力をお願いしようと思っていたのだが、発表に至る考察の結果、その予定は遠のいてしまった。

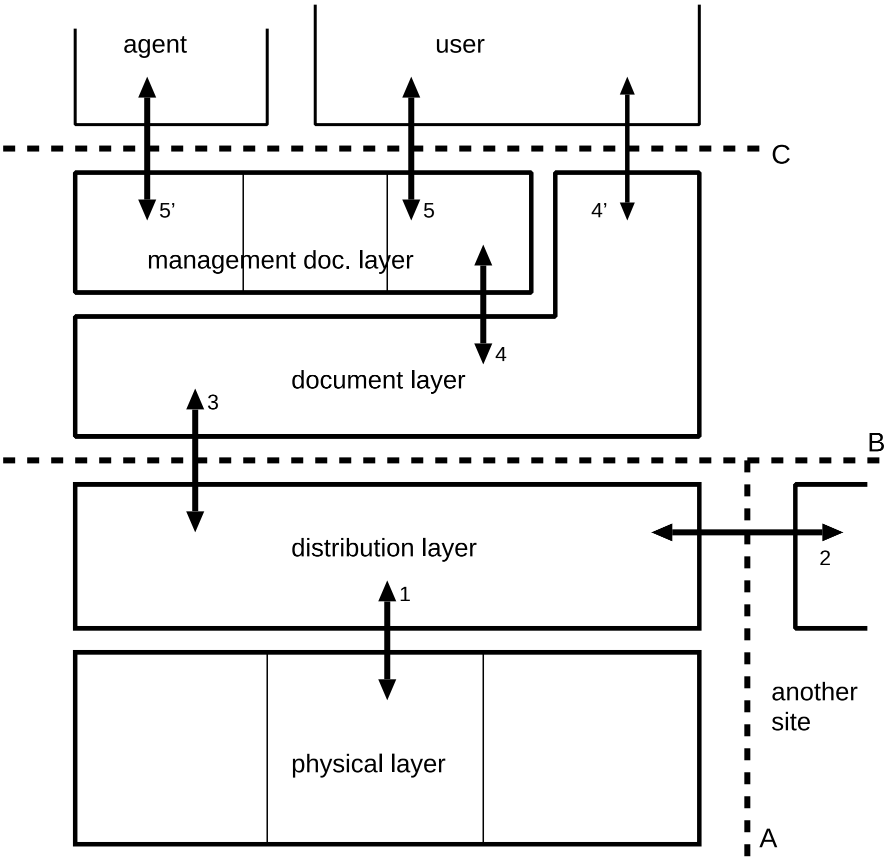

図 2: 階層構造図 (1999 年 9 月 14 日現在)