

双方向サービス向けフレームワークを考慮した仕様詳細化

加来田 裕和† 綾野雅之† 内山尚子‡
†西日本電信電話株式会社 研究開発センタ
‡西日本電信電話株式会社 中国技術総合センタ

概要

インターネットは、商取引やオークションなどを行ったりユーザ同士や企業とユーザのコミュニケーションを行ったりする場として発展してきている。筆者らは、それらのサービスを提供する情報流通加工型のシステムのための情報集約・配信フレームワークを構築している。

本稿では、この情報集約・配信フレームワークをベースにして、様々な双方向性のあるサービスの仕様を詳細化する手順を示す。本手法は、ソフトウェア開発における実装方式である情報集約・配信フレームワークを、仕様検討の段階からスムーズに適用するためのものであり、その仕様の詳細化において考慮しなければならない条件を整理している。情報集約・配信フレームワークと、この仕様詳細化の手順を適用することによって、ソフトウェア開発におけるフレームワークの円滑な適用と生産性向上、システム設計作業の効率化を図ることができる。

The method which defines detailed specifications based on the framework for interactive systems

Hirokazu KAKUDA†, Masayuki AYANO†, Kazuhiro UCHIMOTO‡
† NTT West Research and Development Center
‡ NTT West Chugoku Engineering and Technology Support Center

abstract

The Internet is used as a place of communication of users and companies. Moreover, commercial transaction service and auction service are offered in the Internet. These systems are information gathering, processing, and delivery systems.

This paper shows the method which details the specification of interactive systems. This method is based on information-gathering / distribution framework. The feature of this method is applying the framework smoothly from the stage of specification examination, and is having arranged the conditions when applying this method. If this framework and method are used for software development, the productivity of development can be improved and system design work can be simplified.

1 はじめに

現在、インターネット環境が企業だけでなく家庭にまで急速に浸透してきた。これにより、ネットワーク上では様々なサービスが提供されている。例えば、各企業がホームページを開設し、提供する商品やサービスの宣伝を行ったり、多種多様の情報から目的の情報を探し出す検索サービスを行ったり、インターネットショッピングやオークションを行ったりしている。このように、初めは一方的に情報を発信するだけであったものが、インターネットを利用する人口が増えるにつれ、その情報を基にして取り引きやオークションなど双方向性を生かした、ユーザ同士や企業とユーザのコミュニケーションの場としての利用を行うように発展してきている。

これらのサービスを提供するシステムは、情報を発信し、ユーザなどの応答を集約し、様々な情報を加工し、配信するという一連の手順を追うことによってサービスを提供する情報流通加工型のシステムである。筆者らは、この情報を流通させ加工し双方向のサービスを行うことのできるシステムを安価に構築できる技術が必要だと考え、ソフトウェア開発の工程の効率化を図るために、情報集約・配信フレームワークを構成し[1]。そのフレームワークを適用する際の仕様詳細化の手順を定めた。

本稿では、情報集約・配信フレームワークを用いて仕様の詳細化を行う手順を説明し、ソフトウェア開発へ適用した事例をもとに、情報集約・配信フレームワークとサービスの仕様詳細化の手順を考察する。

2 背景

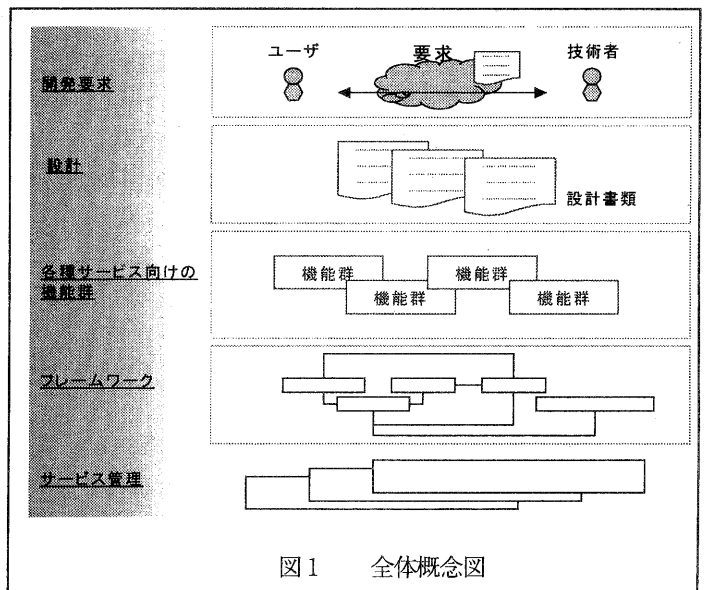
ソフトウェアの仕様を、技術者が作成する際には経験によるところが大きい。また、要求仕様から、基本設計、詳細設計、実装と進めていくことも経験によるところが大きく、非常にコストのかかるものである。さ

らには、設計や実装段階での要求仕様変更は、品質やコストへの影響が大きい。また、サービスの淘汰のスピードが早くなっていることや短期間での開発が要求されることなど、開発工数の削減要求がさらに高まっている。

そこで、筆者らは Internet 上の双方向性のあるサービスの生産性向上に向けて、図1のようなソフトウェア構成の方法が効率よいソフトウェア生産につながると考えている。

まず、ユーザと開発する技術者において、ドメインの要求仕様をまず把握して確定する。その要求しようをもとにして基本仕様、構成仕様、詳細仕様とドメインにあわせて詳細化する。このとき実装段階のフレームワークを念頭において検討する。実装に際しては、双方向性のあるサービス向けのフレームワークを基本として、他ツールの存在も意識し、各種サービスの機能群を作成する。以上のことをソフトウェア開発の一連の流れとして捕らえ、スムーズに実施していく必要がある。

そこで筆者らは、まず双方向性のあるサービスに必要な基本機能を抽出し、フレームワークを構築することによって、実装しやすい環境を整えることとし、[1]高木では、情報集約・配信フレームワークを提案した。



2-1 情報集約・配信フレームワーク

従来、情報を流し加工するようなシステムを用いて双方向サービスを提供するためには、ソフトウェア開発に対してかなりの投資が必要であった。また、今後ネット上の同様のサービスは増加すると考えられる。そこで筆者らは、容易にカスタマイズを行え安価にシステムを構築できる情報集約・配信フレームワークを提案した[1]。このフレームワークで扱う集約配信対象は、これまでのプッシュ型情報配信技術同様、テキストやHTMLを扱うメッセージ型、ならびにファイルを扱うバルク型であり、以下の3点を考慮したものである。

- 協調動作する複数サーバによるサービス提供。
- 同一サーバ上で複数サービスの同時提供
- 構成管理の容易化

このフレームワークをベースとして利用することにより、情報流通加工型のシステムを効率的に構築できるようになると考えた。

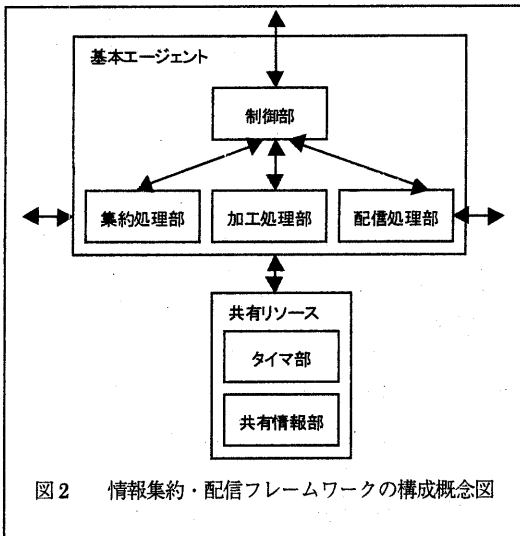


図2 情報集約・配信フレームワークの構成概念図

2-2 仕様詳細化とフレームワーク

提案した情報集約・配信フレームワークでは、実装のために必要な典型的な概念と基本的な枠組みを提案した。しかし、図1に示したように、情報集約・配信フレームワークをソフトウェア開発へ適用するには、要求分析や設計へスムーズに組

み込むための手続きや方法論が必要である。経験によって補うことを行えば、情報集約・配信フレームワークの適用に際して概念の違いや、実装に際して要求や設計情報の変更が生じることもあり、かえって生産コストがかかってしまう。また、一般的な方式を定めてみても、実際の現場では一般的な方式のままでは活用し難い。そこで、情報集約・配信フレームワークを活用可能であり、複数サーバによって構成される、ユーザとの双方向性のあるサービスにドメインを限定して、仕様詳細化の方法を検討した。

3 仕様詳細化の方法に対する要求条件

まず、仕様検討とその詳細化する時点において、提案した情報集約・配信フレームワークを実装の際に使用することを念頭におく。つまり、以下の項目を考慮する。

- 所望の要求を、機能群として機能分割可能
- 機能分割された機能群の分散配置可能
- 機能群の集中配置も可能
- 機能群同士のインタフェースは HTTP, FTP, DB, FILE
- オブジェクト指向の継承概念を用いて実装
- 実装の言語は Java
- サーブレットの実行環境が必要

また、インターネット上のサービスを考慮することから、バースト的なユーザのリクエストに対して処理可能である必要がある。さらに、近年各種の有用なツールが市販されており、開発コスト削減のために市販ツールを活用するほうがよいことがある。今回は、このツール類の活用も念頭におき、ツール類と開発したソフトウェアとのインタフェースを明確にする必要がある。

4 仕様詳細化の手順

上記の仕様詳細化の要求条件を踏まえ、明確にされた要求仕様をもとにして、仕様詳細化の手順を示す。

4-1 要求条件の確認

要求仕様をもとに、まず以下のように検討を行う。

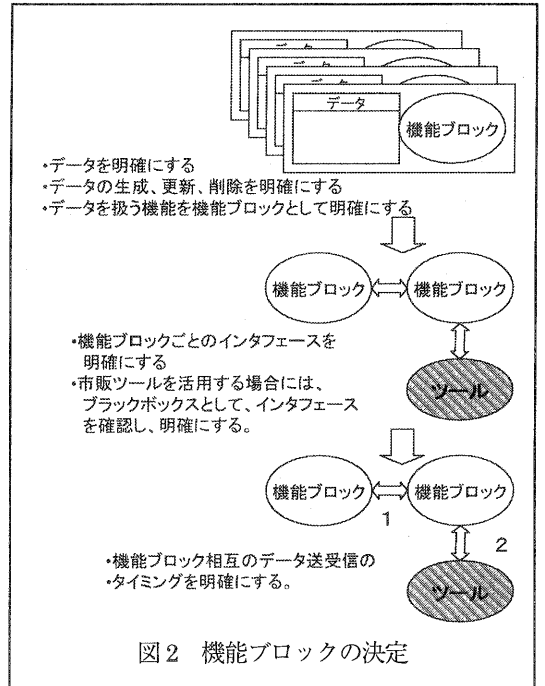
- A) 取り扱うデータを明らかにする
- B) それらのデータの生成・更新・削除のタイミングを明らかにする
- C) それらのデータを取り扱う機能ブロックを明確にする
- D) 複数サーバが必要であるか検討する。
 - ▶ ユーザのリクエストが短時間に集中するようなサービスであるか？
 - ▶ 集中するようなサービスであれば、サーバの分散化が必要か？
- E) データの存在するサーバとそのデータを取り扱う機能ブロックのあるサーバは同じであるか検討する。
- F) データと機能ブロックを他の市販ツールで補うことは可能であるか検討する。
- G) 開発するソフトウェアと市販ツールとのインタフェースは存在するか調査する。
- H) 機能ブロック同士の通信は可能であるか検討する。

4-2 機能ブロックの決定と詳細化

以上のように、要求条件に照らし合わせ検討した結果を踏まえ、機能ブロックを定める。また、その機能ブロックを情報集約・配信フレームワークを用い実装可能なレベルまで詳細化を行う。機能ブロック相互の受け渡すデータの種類やプロトコル、送受信のタイミングなどインタフェース部分も明確に定めておく。この際に、市販ツールを検討し活用するのであれば、インタフェースを明確にしておき、その機能ブロックはブラックボックスとしておく。

まず、図2のように機能ブロックを決めてゆく。

1. 上記、検討したデータの存在および機能ブロックの名称を定める。
2. 定めた機能ブロックごとのインタフェースを検討する。この際には、送信するデータ項目、データの種類、データの構造、受信する送信確認のためのデータ項目、データの種類、データ構造を明確にする。
3. 市販ツールを活用する場合には、ブラックボッ

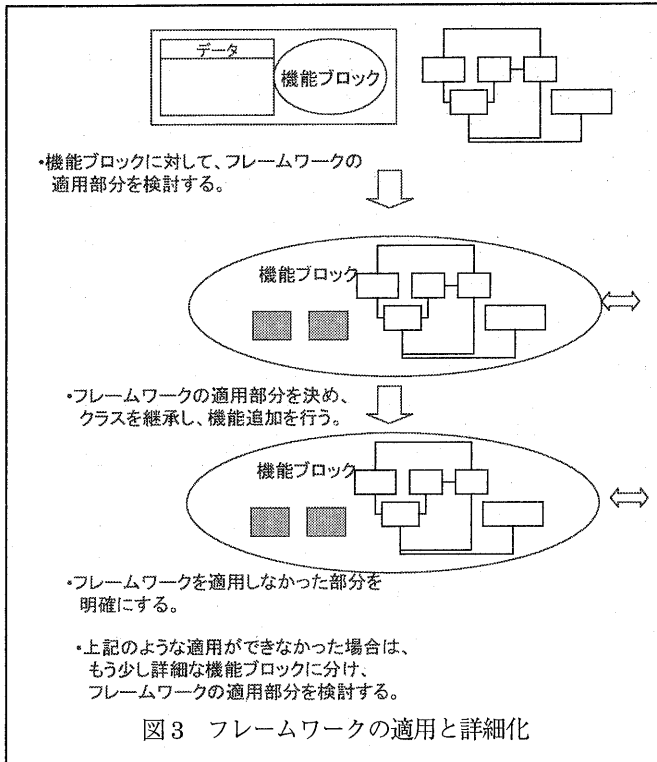


クスとして、入力および出力を調査して送信するデータ項目、データの種類、データの構造、受信する送信確認のためのデータ項目、データの種類、データ構造を明確にする。

4. データの送受信のタイミングを定める。

このあと、図3のように機能ブロックごとに情報集約・配信フレームワークを適用していく。

5. 機能ブロックごとに、フレームワークを適用する部分を明確にする。
 - (ア) 情報集約・配信フレームワークは、データフローを制御することを意識した分散型の設計が可能な形式である。
 - (イ) 複数の機能ブロックが複数のサーバへ分散配置されそれらの機能ブロックがデータを受け取り、渡しながら制御してゆく形式である。
 - (ウ) 情報集約・配信フレームワークはオブジェクト指向の継承概念を活用した実装方式である。機能追加はクラスヘメソッドを追加する形式である。
6. 情報集約・配信フレームワークを適用する部分



を継承概念を用いたクラスとして定める。

7. 上記クラスとして定められなかった、データ構造を示すオブジェクトや、特別な計算を行うクラスを定める。
8. 必要であれば、機能ブロックをさらに詳細な機能ブロックに分ける。この前の段階で、情報集約・配信フレームワークを適用できていないのであれば、機能ブロックの詳細化が不十分であるので、適用できる程度の機能ブロックになるまで、1から繰り返す。
9. 全体のクラス数、フレームワークの継承を行った数、全く新しく作ったクラスの数、フレームワークの使用頻度を観察しておく。

5 考察

以上、情報集約・配信フレームワークとこのフレームワークを念頭において検討した仕様詳細化の手順を示した。今回考察のため、このフレームワークと詳細化の手順を、実際のインターネット上の双方向性のあるシステム開発へ適用した。そ

の適用結果が表1である。

この表は、各システムごとに、プログラム規模とそこで作成したクラス数、および情報集約・配信フレームワークを活用したクラス（派生クラス数、機能ブロックごとの総行数に対する派生クラス行数の割合）、活用していないクラス（非派生クラス数、機能ブロックごとの総行数に対する非派生クラス行数）を表したものである。

まず、クラス数を見ると、総クラス数264個中、情報集約・配信フレームワークを活用した派生クラス数は186個であり、全体のクラス数の約70%にあたる数である。情報集約・配信フレームワークを活用しない非派生クラス数は78個であり、約30%である。この結果より、フレームワークを定義して、そのフレームワークを適用する際の手順を明確にすることで、このフレームワークを活用した派

生クラスの数70%にもおよぶことになったと考えられる。

また、非派生クラスしかない機能ブロックがある。これは情報集約・配信フレームワークで対象にしないクライアントの画面表示の部分であることと、機能ブロックを分割したあとで、実装に必要な機能を作成した部分であるためである。

次に、派生クラス数や非派生クラス数との割合ではなく行数を見ると、総行数中、情報集約・配信フレームワークを活用した派生クラスの行数の割合は、約39%である。情報集約・配信フレームワークを活用しない非派生クラスの行数の割合は、約61%である。この結果は、派生クラスは情報集約・配信フレームワークを活用することにより、実装の際のコーディング量が減少していると考えられる。また、非派生クラスでは、1クラスあたり4百数十行となっており、1クラスあたりの行数としては大きなものとなっていると考えられる。これは、フレームワークとして構成が整備されており、決まった形式で実装したクラスであるものと、自由に作成したクラスであるものの違いのためと

表1 プログラム規模とクラス数

システム名	機能ブロック名	派生クラス数	機能ブロックごとの派生クラス行数の割合	非派生クラス数	機能ブロックごとの非派生クラス行数の割合	総クラス数
システム A	配送スケジュール	25	45%	8	55%	33
	ユーティリティ	0	0%	5	100%	5
	フィルタリング	19	48%	15	52%	34
	選択履歴更新	3	23%	10	77%	13
	セッション管理	21	22%	10	78%	31
	ブラウザコントロール	0	0%	1	100%	1
システム B	コンテンツ管理	59	48%	7	52%	66
	コンテンツ配信制御	2	13%	2	87%	4
	広告コンテンツ管理	21	40%	4	60%	25
システム C	加工内容入力	11	40%	3	60%	14
	分析ファイル中継	23	71%	4	29%	27
	データ加工変換	2	3%	9	97%	11
合計		186	39%	78	61%	264
総クラス数における割合		70%	-	30%	-	-
			クラス数	総行数における割合		
情報集約・配信	フレームワーク		60	13%		
フレームワーク	構成管理ツール		18	13%		
合計			78	26%		

考えられる。

これらの結果より、今回の適用においては、情報集約・配信フレームワークを使用し、仕様詳細化の手順を決め、ある特定のドメインに適用することによって、概念的に整理可能なクラスはフレームワークも含めると77%程度であり、約54%の行数がそのクラスとして整理することが可能であった。このように、情報集約・配信フレームワークと、この仕様詳細化の手順を適用することによって、ソフトウェア開発におけるフレームワークの円滑な適用と生産性向上、システム設計作業の効率化を図ることができる。

6 おわりに

情報集約・配信フレームワークを使用し、仕様詳細化の手順を決め、ある特定のドメインに適用することによって、概念的に整理可能なクラスはフレームワークも含めると77%程度であり、約54%の行数がそのクラスとして整理することが可能であった。これによって、ソフトウェア開発工程の効率化が望めると考える。今後は、このフレームワークの適用領域の拡大を目指し、様々なサービスを提供する際に活用して行き、情報集約・配信フレームワー

クの品質の向上、各種サービスに流用可能な機能群の整理を行っていく予定である。

【参考論文】

高木、日浦、小泉、フレームワークをベースにした情報集約・配信システム構築方式、情報処理学会研究報告、マルチメディア通信と分散処理、第95回