

Web 情報システムのソフトウェアアーキテクチャ に関する事例研究

副題：研究室情報システムの ソフトウェアアーキテクチャの構築

99B578 水野雄治 99B639 内田晴子

指導教員 野呂昌満

1 はじめに

現在インターネット上に存在する多くのアプリケーションソフトウェアはユーザインタフェースとして Web ブラウザを用いており、これを Web 情報システム (Web-based Information System, 以下 WIS) と呼ぶ。WIS は様々な Web 技術で実現可能であり開発者の混乱の原因となっていたが、昨年度 WIS のソフトウェアアーキテクチャ [1] が提案され、Web 技術の適用法と構成要素と構成要素間の関係が示された。

昨年度提案された WIS のソフトウェアアーキテクチャでは WIS が Web アプリケーションであることに着目して Web 技術の適用法と開発プロセスを示した。しかしわれわれは WIS が情報システムであることに着目することで新たな構造が発見できるのではないかと考えた。

本研究の目的は、WIS が情報システムであることから生じる構造を整理し、昨年度提案された WIS のソフトウェアアーキテクチャを再構築することである。再構築したソフトウェアアーキテクチャをもとに研究室情報システムの構築をおこなう。

再構築した WIS のソフトウェアアーキテクチャを、固有の応用領域に特化したソフトウェアアーキテクチャに当てはめて検証することで、再構築した WIS のソフトウェアアーキテクチャの有用性が確認できると考える。

われわれは以下の手段で WIS のソフトウェアアーキテクチャの再構築をおこなった。はじめに WIS のソフトウェアアーキテクチャに基づいた実現・考察をおこなった。次に WIS のソフトウェアアーキテクチャの構成要素を詳細化し、Page という構成要素を構成要素 Application から分離した。それぞれの構成要素に存在する状態遷移機械間の協調関係を考えることで、昨年度提案された WIS のソフトウェアアーキテクチャでは述べられていない構成要素内の構造について示した。その結果、画面のデザイン、設計、遷移の変更やデータベースの変更に柔軟に対応できるようになった。

2 WIS のソフトウェアアーキテクチャ

昨年度提案された WIS のソフトウェアアーキテクチャでは、WIS の構造と開発プロセスを示している。一般的なソフトウェアアーキテクチャでは、ソフトウェアの構造について示すだけにとどまっているが、本研究ではソフト

ウェアアーキテクチャは実現のプロセスも含めるものと考ええる。

また、ソフトウェアアーキテクチャの構成要素の抽象度は一般的には同一であると理解されているが、われわれは異なる粒度の構成要素の集まりであるととらえる。抽象度の異なる構成要素と構成要素間の関係は、抽象側面・具象側面・プロセス側面の 3 側面 [3] から記述する。

WIS のソフトウェアアーキテクチャは、

- ユーザインタフェースを保持
- データベースを保持するオブジェクト指向アプリケーション

という 2 つの視点から、MVC モデルと 3 層モデルを用いて構造を整理している。

3 研究室情報システムの概要

本研究で対象とする研究室情報システムでは、氏名・E メールアドレスなどの個人情報と研究テーマ、論文、関連論文、メンバーの研究情報の 2 種類の情報を扱う。また機能として登録・削除・変更・検索の機能を持つ。ユーザの権限としては、管理者と研究室関係者と一般閲覧者の 3 つを考える。一般閲覧者は研究情報の検索、研究室関係者は全ての情報の検索、管理者は全ての情報に対してどの機能も使用できる。

研究室情報システムの試作として学生情報システムを実現した。学生情報システムでは、氏名・学籍番号・E メールアドレスの 3 種類の情報を扱い、登録・削除・変更・検索の機能を持つ。ユーザの権限としては、管理者と研究室関係者の 2 つを考える。管理者は学生情報システムの全ての機能を使用することができるが、研究室関係者は検索機能しか使用することができない。

4 Page と Application を分離した WIS のソフトウェアアーキテクチャ

再構築した WIS のソフトウェアアーキテクチャでは、昨年度提案された WIS のソフトウェアアーキテクチャの構成要素 Application を詳細化し、Page と Application に分離した構造を考える。

昨年度構築された WIS のソフトウェアアーキテクチャに基づいて、個人情報システムを試作し、考察をおこなった。実現した個人情報システムでは、ユーザからの入力の受け付けからブラウザへの出力まで全ての処理を Application という構成要素内でおこなっている。われわれは、画面に関する処理をおこなっている部分を構成要素 Application から独立させ、新しい構成要素とすることが出来ないかと考え、構成要素 Page とした。再構築した WIS のソフトウェアアーキテクチャの抽象側面のクラス図を図 1 に示す。

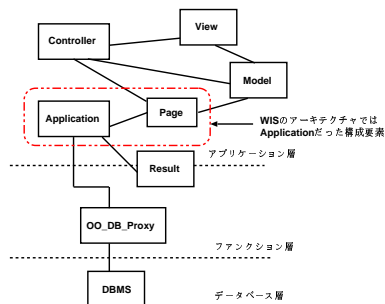


図 1: 再構築した WIS のソフトウェアアーキテクチャの抽象側面のクラス図

本研究で再構築した WIS のソフトウェアアーキテクチャは、ユーザからの入力を構成要素 Application から分離した新しい構成要素である Page が受け取る構造とした。再構築した WIS のソフトウェアアーキテクチャでは、昨年度提案された WIS のソフトウェアアーキテクチャにおいてユーザからの入力を受け取っていた構成要素 Application を構成要素 Page と構成要素 Application に分離した。再構築した WIS のソフトウェアアーキテクチャにおいて、構成要素 Page と構成要素 Application の構成要素がユーザからの入力を受け取るかを決定するために、ユーザからの入力について分析をおこなった。分析の結果、ユーザがなんらかの入力をしたときは必ず画面の遷移が発生していることがわかった。このことから、ユーザからの入力を構成要素 Page が受け取る構造とすれば、構成要素 Application に対して必要な場合のみデータベースアクセスの依頼をすることになり、自然な構造であると考えた。

5 Application と Page の保持する状態遷移機械とその協調関係の分析

本研究で再構築した WIS のソフトウェアアーキテクチャでは、構成要素 Page と構成要素 Application が状態遷移機械を保持し、状態遷移機械間には協調関係が存在すると考え、分析をおこなった。一般的な WIS では、構成要素 Page が保持する状態遷移機械と構成要素 Application が保持する状態遷移機械の相互作用を考えて設計をおこなう。

5.1 Application と Page の保持する状態遷移機械

再構築した WIS のソフトウェアアーキテクチャでは、画面の処理をおこなう構成要素 Page と構成要素 Application を状態遷移機械によって実現する。

WIS ではアプリケーションを使用する過程で画面に表示される内容が次々と移り変わる。これら一つ一つの画面はユーザからのなんらかのイベント(リンクやボタンがク

リックされる)を待っている状態と捉えることができる。1つ1つの画面を待ち状態と考えると、画面の移り変わりは状態の変化であると考えられるので構成要素 Page を状態遷移機械で実現する。

画面に関する処理を構成要素 Page として独立させた後の構成要素 Application は、複数の状態遷移機械の集合と応用の論理で構成される。構成要素 Application 内に存在する状態遷移機械の数は、データベースが保持するテーブルの数によって決定される。一般的な WIS では、データベース内にいくつものテーブルを保持している。状態遷移機械とデータベースが保持するテーブルは1対1に対応しており、それぞれの状態遷移機械が対応するテーブルにアクセスする。構成要素 Application の内部にあるそれぞれの状態遷移機械と構成要素 Page 内に存在する画面に関する処理をおこなう状態遷移機械には協調関係が存在する。構成要素 Application の内部にある状態遷移機械は互いに独立であり、状態遷移機械間の協調関係は存在しない。実行する機能によってアクセスするテーブルは異なる。入力されたイベントによって、構成要素 Application 内のどの状態遷移機械が構成要素 Page と協調関係を持つかが決定される。

5.2 状態遷移機械間の協調関係の分析

構成要素 Application と構成要素 Page の保持する状態遷移機械間の協調関係を示すために状態遷移機械間のメッセージ通信について分析をおこなった結果、複数の協調関係が存在することがわかった。

再構築した WIS のソフトウェアアーキテクチャを研究室情報システムの試作である学生情報システムにあてはめて考えてみると、Application 内には2つの状態遷移機械が存在する。学生情報システムのデータベースは以下の2つのテーブルを保持している。

- 学生情報を管理するテーブル
- ユーザ権限を管理するテーブル

構成要素 Application はデータベースが保持する2つのテーブルに1対1で対応する2つの状態遷移機械を保持する。構成要素 Application が保持する2つの状態遷移機械と、構成要素 Page が保持する画面に関する処理をおこなう状態遷移機械をそれぞれ1つのオブジェクトとして、オブジェクト間のメッセージ通信をあらわしたものが図2である。状態遷移機械間の協調関係を分析した結果、以下のような2種類の協調関係が生じる。これを図3と図4に示す。

学生情報システムの機能を実行する場合の協調関係

学生情報システムの機能を実行する場合には、構成要素 Application が保持する機能を実行する状態遷移機械が構成要素 Page が保持する画面に関する処理をおこなう状態

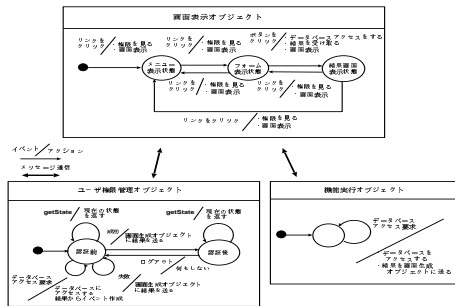


図 2: 状態遷移機械間のメッセージ通信

遷移機械と協調関係を持つ。学生情報登録画面や学生情報検索画面で、登録ボタンや削除ボタンを押された場合がこれにあたる。画面に関する処理をおこなう状態遷移機械にボタンが押されたことがイベントとして通知されると、機能を実行する状態遷移機械にイベントが渡され、データベース内の学生情報を管理するテーブルにアクセスした後、その結果が返される。結果の内容によって画面に関する処理をおこなう状態遷移機械は結果画面を生成する。

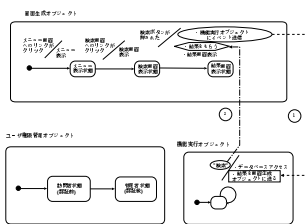


図 3: 学生情報システムの機能を実行する協調関係

ユーザー権限を変更する場合の協調関係

ユーザー権限を変更する場合には、構成要素 Application が保持するユーザー権限を管理する状態遷移機械が、構成要素 Page が保持する画面に関する処理をおこなう状態遷移機械と協調関係を持つ。ユーザー認証画面で、認証ボタンが押された場合がこれにあたる。認証ボタンが押されたことが画面に関する処理をおこなう状態遷移機械にイベントとして通知される。するとユーザー権限を管理する状態遷移機械にイベントが渡され、データベース内のユーザー権限を管理するテーブルにアクセスする。データベースアクセスの結果は、ユーザー権限を管理する状態遷移機械にイベントとして渡され、状態の遷移がおこる。状態の遷移のさいのアクションで、画面に関する処理をおこなう状態遷移機械にイベントを渡し、ユーザー認証の結果画面が生成される。

6 考察

本研究で再構築した WIS のソフトウェアアーキテクチャについて考察をおこなう。

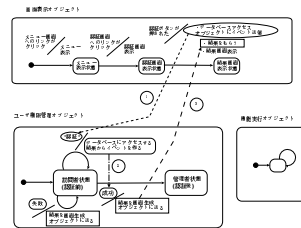


図 4: ユーザー権限を変更する場合の協調関係

6.1 画面に関する変更についての考察

画面の遷移を変更する場合について考察する。再構築した WIS のソフトウェアアーキテクチャでは一画面を一状態にとらえて、画面の移り変わりを状態遷移機械で表現した。これにより、画面の遷移を制御することができる。

画面に関する変更には 2 種類あると考える。

画面の遷移の変更

再構築した WIS のソフトウェアアーキテクチャでは画面遷移の変更は Page の状態遷移機械に遷移先を追加するだけで対応できる。現在は登録フォームが表示されている状態から削除フォームを表示させたい場合には、一度メニュー画面にもどって削除フォームへのハイパーリンクをクリックするようになっている。これを登録フォームから削除フォームに直接ハイパーリンクを貼りたい場合には、登録フォーム表示状態の遷移先に削除フォーム表示状態を追加するだけでよい。

入力画面の設計の変更

現在、登録という機能を実行する場合には登録フォームが一画面に表示され、全ての情報を一度に入力するようになっている。これを一情報ごとに画面を切り替えて入力するようにした場合には、一画面を一状態としてとらえると、状態の数を変化させることで対応できる。この場合、画面が切り替わっても最終的に登録ボタンが押されるまでデータベースをアクセスしないので、一つ一つ入力された情報を保持する必要がある。再構築した WIS のソフトウェアアーキテクチャでは、Application の内部に情報を保持することで、入力画面の設計に対応できると考える。

6.2 昨年度の提案された WIS のソフトウェアアーキテクチャに基づく開発との比較

本研究で再構築した WIS のソフトウェアアーキテクチャでは、昨年度提案された WIS のソフトウェアアーキテクチャでは示されなかった情報システム固有の構造を発見し、その協調関係を考えることで構造を示すことができた。その構造を昨年度の提案された WIS のソフトウェア

アーキテクチャに基づく開発と比較して考察をおこなう。

本研究で示した構造はデータベース内のテーブルが保持する情報に変更をおこなう場合や扱う情報や機能を追加する場合にも柔軟に対応できるようになった。昨年度の提案された WIS のソフトウェアアーキテクチャに基づいて開発すると、図 5 のようにアプリケーションの構造が示されていないのでテーブルの保持する情報の変更、扱う情報や機能の追加に柔軟に対応できない。

本研究のソフトウェアアーキテクチャは画面の遷移、ユーザの権限を管理する遷移、機能の実行による遷移をそれぞれ別々の状態遷移機械で実現し、その協調関係を考えることで構造を定義している。構成要素 Application 内の状態遷移機械はそれぞれデータベースのテーブルと 1 対 1 に対応する構造になっている。したがって図 6 のようにある画面からある画面に遷移するときに構成要素 Application 内のどの状態遷移機械と協調関係を持つか、どのデータベースと関わるかが容易に理解でき、変更箇所が見やすい。

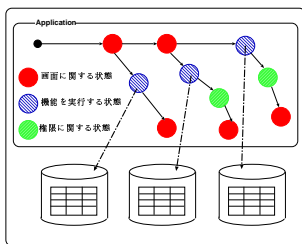


図 5: 昨年度の提案された WIS のソフトウェアアーキテクチャに基づく開発

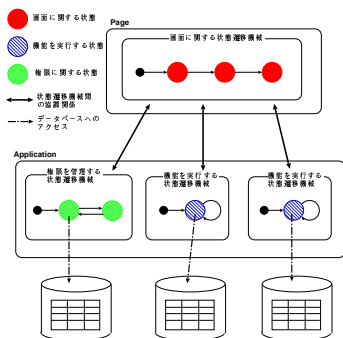


図 6: 状態遷移機械間のメッセージ通信とデータベースとの関係

6.3 まとめ

本研究のソフトウェアアーキテクチャを用いることで画面のデザインの変更、画面の遷移の変更、データベースの変更にも柔軟に対応できるようになった。昨年度構築され

た WIS のソフトウェアアーキテクチャでは全ての処理を Application で実現していたので、アプリケーションが応用の論理以外の画面のデザインの変更などに柔軟に対応できなかった。

本研究のソフトウェアアーキテクチャは構成要素 Application から画面に関する処理を構成要素 Page として分離した。Page を分離することで画面のデザインや画面の遷移の変更にも柔軟に対応できる。また情報システム特有の構造として画面の遷移とアプリケーションの遷移の協調関係を定義した。画面の遷移は構成要素 Page に状態遷移機械を保持することで実現した。アプリケーションの遷移として、ユーザの権限を管理する状態遷移機械と、複数の機能を実行する状態遷移機械を構成要素 Application が保持することで実現した。また Application で保持する状態遷移機械はデータベース内のテーブルと 1 対 1 に対応し、テーブルの数だけ状態遷移機械が存在する。この構造にすることでテーブルの保持する情報を変更する場合に変更する部分を特定できるので、変更にも柔軟に対応できる。

7 おわりに

本研究では、昨年度提案された WIS のソフトウェアアーキテクチャを再構築し、考察をおこなった。昨年度提案された WIS のアーキテクチャに基づく試作・考察をおこない、情報システム特有の構造を新たに Page という構成要素として Application から独立させた。Page と Application にはそれぞれ状態遷移機械が存在し、Page の保持する状態遷移機械と Application の保持する複数の状態遷移機械の協調関係を考えた。

今後の課題として本研究で試作をおこなった研究室情報システムの実現をおこない、さらに本研究の WIS のソフトウェアアーキテクチャの有用性を確認する。

謝辞

一年半の間ご指導くださいました野呂昌満教授、有益なアドバイスをくださった張漢明助教授、蜂巢吉成先生、大学院生の山本英貴さん、森貴彦さん、熊崎敦司さん、朝長武士さん、藤原泰昌さんに深く感謝致します。

参考文献

- [1] 熊崎敦司, 野呂昌満, 張漢明, 蜂巢吉成, Web 情報システムのソフトウェアアーキテクチャ, オブジェクト指向最前線, 2001
- [2] 野呂昌満, 後藤邦夫, ネットワークとマルチメディアトラック / インターネットソフトウェア, 共立出版, 2002
- [3] P.B.Kruchten, "The 4+1 View Model of Architecture," IEEE Software, Vol. 12, No. 6, pp.42-50, Nov. 1995