

Java フレームワーク Seasar2 を用いた Web アプリケーション 開発：介護業務支援システム「アイケア」

小出 哲久[†] 和田 成晃[†] 矢口 隆明[†] 白石 善明[†] 岩田 彰[†]

[†]名古屋工業大学大学院 〒466-8555 名古屋市昭和区御器所町

E-mail: {nori, sessionxx} @mars.elcom.nitech.ac.jp,
yaguchi@comand.co.jp, {zenmei, iwata} @nitech.ac.jp

あらまし 介護業務の品質向上を目指した支援システムの開発と愛知県内の複数の介護事業所にて実証実験を行った。本論文では、Web アプリケーションとして開発した介護業務支援システムの事例を通して、Java フレームワーク Seasar2 での開発がユーザの要求を固めながら柔軟にシステム開発ができることに考察を与える。

キーワード Java Web アプリケーション Seasar2 Teeda 介護業務支援システム

Web application development that uses Java framework Seasar2: Nursing business support system "AiCare"

Norihisa KOIDE[†] Nariaki WADA[†] Takaaki YAGUCHI[†]

Yoshiaki SHIRAISHI[†] Akira IWATA[†]

[†] Nagoya Institute of Technology Gokisocho, Shouwa-ku Nagoya, Aichi 466-8555 Japan

E-mail: {nori, sessionxx} @mars.elcom.nitech.ac.jp,
yaguchi@comand.co.jp, {zenmei, iwata} @nitech.ac.jp

Abstract Improving quality of health service is needed. We have developed a nursing business support system called 'AiCare' aiming at the quality improvement of nursing. AiCare system is a Java Web application and is implemented by Seasar2 framework. The reason of using Seasar2 is why it is short term development and requirements to system are unfixed. This paper reports a case study of Web application development by Java Seasar2 framework, and gives some considerations on efficiency of requirement-unfixed system development.

Keyword Java Web application Seasar2 Teeda Nursing business support system

1. はじめに

今日の Web アプリケーション開発は開発期間の短縮や開発コストの低減、大規模化により、開発の効率化が求められている。アジャイルといった迅速かつ適応的にソフトウェア開発を行う手法も利用されている。システム開発においては、開発のスタート時点で仕様が固まっていない場合が多い。これはユーザが発注しようとしているシステムにどのような機能が必要になるかわかっていないためである。システム開発を効率よく、かつ確実にを行うには、ユーザの要求を明確にす

る必要がある。

我々は在宅介護の質的向上を目指した在宅介護支援システムの開発を Java の Web アプリケーションフレームワークである Seasar2 を用いて行った。我々の在宅介護支援システムは介護の業務フローを変更することなく現場知をデータベース化し情報の共有を目指している。このシステム開発においては、先述したように仕様が確定しておらず、要求を明確にすることから始める必要があった。特に、介護業務の現場では IT 化が進んでいないところもあり、情報機器やシステム

への理解が十分であるとは言えないユーザの要求を明確化する必要がある。そこで、ユーザ要求については、業務を遂行する中でのユーザの意図を表層化させ、ユーザ側と開発側の意識が一致するよう努め、ユーザの要求を支援するシステムの設計を容易にかつ一貫したものにするように注力した。

3 か月間の介護支援システムの開発により、Seasar2を用いた Web アプリケーション開発は、Seasar2の機能の1つである Teeda の HTML テンプレート機能を用いて、ユーザ要求を固めながら柔軟にシステム開発ができ、工期の短縮もできることがわかった。以下ではそのシステム開発について述べ、考察を与える。

2. 短期間開発かつ要求未定義

今日、パソコンの高性能化やブロードバンドネットワークの普及、e-文書法の制定などの政府の後押しに伴い、業務システムの Web アプリケーション化が急速に進んでいる。また、開発期間の短縮や開発コストの低減、大規模化により、開発の効率化が求められている。近年、アジャイルを初めとしたソフトウェア開発を効率よく行う手法が数多く提案されている。

システムを開発していく上で重要なことの一つに要求定義が挙げられる。ここではユーザからシステムに必要な機能を聞き出されるが、ユーザの要求が固まっていない場合がある。これはユーザがシステムに必要な機能を理解していないためであることが主要な理由として挙げられる。したがって、開発を進めながら仕様を固めていく必要がある場合も多い。

このように、ユーザである顧客の要求が固まっていないシステムを短期間で開発しなければならないといった要求が生じている。

3. Java の Web アプリケーション開発

HTTP プロトコルが一般的なことと Web インタフェースはユーザがなじみやすいことから、近年では業務システムも Web アプリケーションシステムとして開発されるのが一般的になっている。ここでは、一般的な Web アプリケーション開発に用いられるソフトウェア設計である MVC モデルと、Java アプリケーションフレームワークについて説明する。

3.1. MVC モデル

MVC モデル [1] は、Java Platform, Enterprise Edition (Java EE) における Web アプリケーションを構築する際のアーキテクチャのモデルである。MVC とは、Model, View, Controller のことである。MVC モ

デルでは、プログラムをこの3つの役割に明確に分けてアプリケーションを構築する。

MVC モデルに従うことで、機能ごとの分離が明確になる、開発の分担が行いやすい、コンポーネントの再利用性が高まる等、様々な利点が期待できる。

3.2. Java Platform, Enterprise Edition [2]

Java Platform, Enterprise Edition (Java EE) はサーバーサイドの Java の仕様で大規模なサーバーシステム向けに Java Servlet (サーブレット)、JavaServer Pages (JSP)、Enterprise JavaBeans (EJB) 等の機能を提供している。Java EE の仕様に従って Web アプリケーションを MVC モデルで構築すれば、完成度の高いシステムの構築ができる。

しかし、EJB では実行環境の構築が手間である、単体テストが大変である、といったように様々な問題点がある [3][4]。これらは、Web アプリケーションをアジャイルに開発していかねばならない場合には利用が適切であるとは言えない。そこで Java の Web アプリケーションフレームワークが用いられ、また、EJB コンテナに変わる軽量コンテナが使われることが増えている。次章では軽量コンテナについて述べる。

3.3. 軽量コンテナ

軽量コンテナとは、AOP (Aspect Oriented Programming: アスペクト指向プログラミング) の技術を取り入れた DI (Dependency Injection: 依存性の注入) コンテナのことである。EJB を重量コンテナと呼ぶのと反対の意味として用いる。

DI はオブジェクト間の依存性を低くする。ソースコードには機能を利用する部分のみ記述し、実行時にどのオブジェクトを利用するか決める。オブジェクト間の依存性を管理するプログラムのことを DI コンテナと呼ぶ。DI コンテナを使用することで作業を分担しやすい、仕様変更時に影響が少ない、テストが容易である等の利点がある。

AOP とはクラス本来の処理とは異なる他の処理をそのクラスの外側から織り込むことである。例えば、商品検索を処理するクラスとトランザクション処理のような関係である。トランザクション処理のような本来の処理とは関係ない処理をまとめて別のクラスに記述することでソースコードがシンプルになり管理しやすくなり、生産性も上がる [5][6][7]。

これら DI コンテナと AOP を組み合わせることで、大きな利点が生まれる。

このように、軽量コンテナを用いれば、仕様変更が発生した場合に柔軟に対応でき効率よくシステム開発

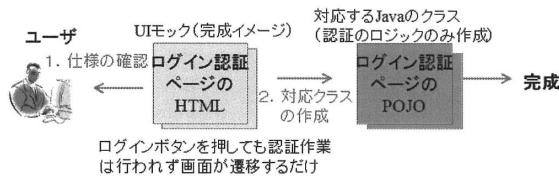


図 1 : HTML テンプレート機能

が可能であると言える。

代表的な軽量コンテナとしてはSpring[8] やSeasar2などが挙げられる。次章ではSeasar2について述べる。

4. Seasar2 フレームワーク

4.1. Seasar2 概要

Seasar2 フレームワークはSeasar ファウンデーション[9]によってオープンソースとして開発・提供されているJavaのWeb アプリケーションフレームワークである。DI と AOP を用いた軽量コンテナであり、S2Container を中心にJava を使ったWebアプリケーション開発の生産性を高めるための様々なフレームワークを提供している。

4.2. S2Container

S2Container は DI と AOP の機能を持つ軽量コンテナである。オブジェクト間の依存性を低くすることでコンポーネントの再利用性、拡張性、テスト容易性を向上させている。各オブジェクトは必要なコンポーネントを取得しに行く必要はなく、受け口を用意しておくことでコンテナから渡してもらう。また、AOP の機能によって、POJO (Plain Old Java Object) のメソッド呼び出しをインターセプトし、トランザクションをはじめとする様々な機能を付加する。S2Container の AOP はバイトコード動的に修正する POJO のメソッドにフックを挿入し、EJB オブジェクトのような代理オブジェクトを使わずに各種コンテナ機能の織り込みを実現している。

4.3. Teeda

Teeda[10] はSeasar ファウンデーションが提供するJSF 実装である。「JSF meets DI ×AOP」というコンセプトの元、JSF の標準機能に加えDI と AOP の機能をベースとした拡張機能を提供し、開発効率と品質の高いWeb アプリケーション開発実現を掲げ、テストの自動化、POJO中心の開発スタイルなどをゴールとしている。

4.4. HTML テンプレート機能

Java の Web アプリケーションフレームワークであ

る Teeda は、ユーザの要求を明確化し、システムの仕様を決定するために作成した UI(User Interface) モックを、そのままシステムの View として取り込むことができる。ユーザはシステムが完成する前に、完成したシステムの挙動についての情報を得られるため、安心して開発を任せることができる。また、開発者は仕様定義を開発における実装の一工程として取り込めるため、効率の良い開発ができる。

Teeda においては、Java EE においてプレゼンテーション層である View や Controller を JSP やサーブレットで作成しているのに対して、View には HTML を使い、Controller として POJO を用いている。Controller の POJO は HTML のファイル名の語尾に Page というプレフィックスを付した名前をつけることで関連付けている。これを HTML テンプレート機能と呼ぶ。この機能は、Web システム開発の初期段階や新たなページの追加などの際に、ユーザの要求を明確にするための機能である。手順は以下ようになる。

1. 開発者はユーザの最初の開発依頼を元に UI モック (システムの画面遷移を HTML のみで再現した完成イメージ) を作成する
2. 開発者は UI モックを元にユーザと仕様の確認を行う
3. 仕様の確認が終了後、開発者は UI モックの HTML に対応する Controller クラス (Teeda においては Page Class) を作成する
4. Web システムの完成

HTML テンプレート機能を使うことで、システムの動きの検証を UI モックとして先に作成した HTML のみで行うことができる。実際の開発において、これを用いて顧客との仕様定義を行い、仕様締結後はこの UI モックに対応する POJO を作成するだけでシステムの開発ができる。

4.5. Seasar2 を用いた Web アプリケーション開発の利点

Seasar2 を用いて Web アプリケーション開発を行う場合、HTML テンプレート機能を用いることでユーザ

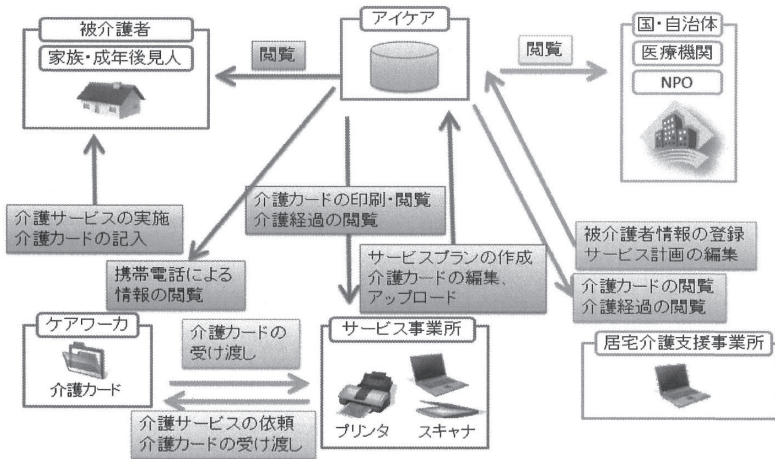


図 2：在宅介護支援システム「アイケア」による在宅介護の流れ

は実際の画面を見ながら完成したシステムのイメージを持つことができる。もし、開発スタート時に仕様が固まっていない場合、UI モックを用いてシステムの検証を行うことができる。以上のように Seasar2 を用いてシステム開発を行えば、仕様の固まっていない場合や開発途中の仕様変更にも柔軟に対応できる。そして、徐々に仕様を固めながら開発していくことが可能である。

また、軽量コンテナにより、

- ・ 設計が容易になる
 - ・ コード訂正を行いやすい
 - ・ テストが容易になる
- 等様々なメリットが期待できる。

5. 在宅介護支援システム「アイケア」の開発

3章、4章で述べたように要求が未定義なWebアプリケーションの開発に対して、Seasar2 を用いて開発を行えば、UIモックを中心とした柔軟な開発が行えると期待できる。

名古屋工業大学大学院岩田研究室において、平成 19 年度に在宅介護の質的向上を目指した Web システムの開発を Seasar2 により行った。その開発について以下では報告する。

5.1. 在宅介護支援システム「アイケア」の概要

近年、介護を必要とする高齢者の数が増加している。そして、増加する介護費用を将来に渡って国民全体で公平に賄う仕組みとして、2000 年 4 月より介護保険制

度が開始された[11]。

我々は、介護サービスの中でも介護保険利用者の増加が著しい在宅サービスに焦点を当て、地域で連携してサービスの品質の維持、向上を図るため、高度介護情報共有システムについての研究を行った。そして、これまでサービス事業所（ケアワーカーを派遣して介護保険サービスを行う事業者。自治体、医療法人、NPO 法人などが担う）やケアワーカー（実際に介護サービスを直接行う援助者）が個々に持っていた被介護者の情報、特に現場知を地域で共有する仕組みを提案した。それは、各々の情報を Web システムによって管理し、コンピュータに不慣れたケアワーカーに負担のないように、紙媒体の介護経過報告書である「介護カード」を利用するというものである。情報を一元管理、共有することで事業者間の情報伝達を容易に行い、事務処理を自動化することで事業者の負担を減らすことができる。また、紙媒体を用いることで情報機器に不慣れたケアワーカーに負担をかけず、ケアワーカーの持つ貴重な現場知を共有することができ、介護サービスの質の向上が期待できる。Web システムを開発した後、実際の介護の現場にて実証実験を行いシステムの有効性を検証している（[12]にて報告する）。

5.2. Seasar2 を用いた「アイケア」の開発

5.2.1. 概要

5.1章で述べた提案を元に、現場知の共有により在宅介護の質的向上を目指した在宅介護支援システム「アイケア」の開発を行った。

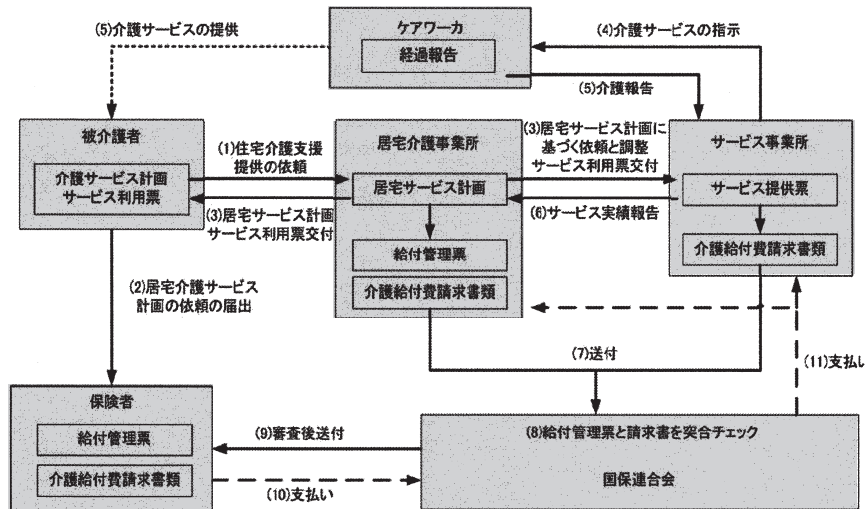


図 3：在宅介護サービスの流れ[13]

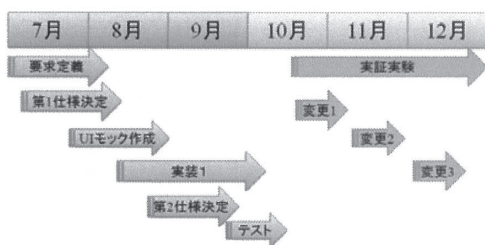


図 4：開発・実証スケジュール

図 2に在宅介護支援システム「アイケア」による在宅介護の流れを示す。システム利用の流れは以下のようである。

1. 在宅介護支援事業者がアイケアに被介護者情報を入力する
2. サービス事業者は入力された被介護者情報を元に介護計画を作成する
3. サービス事業者は介護カードを出力する
4. ケアワーカーは介護カードを用いて、介護を実施し介護カードに記入する
5. サービス事業者に持ち帰った介護カードを読み込み、アイケアに情報を取り込む

この一連の流れを「アイケア」によって実現する。

図 4に開発から実証までのスケジュールを示す。

平成 19 年 7 月から 9 月に開発を行い、10 月から 12 月に実証実験を行った。要求定義と第 1 仕様決定がほぼ

平行して行われた。第 1 仕様決定の途中から UI モックの作成も行った。実装の途中で第 2 仕様変更を行った。また、実証実験中も実証先の要望に合わせて仕様変更を行った。

5.2.2. システム詳細

「アイケア」システムが解決すべき要求は以下の 2 点である。

- Web システムによる被介護者情報の管理
- 介護カードを用いた在宅介護サービスの業務支援システムの作成

図 3に一般的な在宅介護サービスの流れを示す。

1. 被介護者は居宅介護支援事業所（ケアプランの作成や各種連絡・調整・手続きを行う。ケアマネージャが常勤でいることが義務付けられている）に居宅介護支援を依頼する（ケアマネージャ：ケアプランの作成を行う。国家資格が必要）
2. 居宅サービス計画書類（ケアプランとも呼ぶ：要介護者・支援者の心身の状況、環境などからどのようなサービスを利用するかまとめたもの）を作成する。そして、被介護者に対しては「サービス利用書類（兼在宅サービス計画）」を、サービス事業者に対しては「サービス提供書類」を交付する
3. サービス事業者は「サービス提供書類」に基づいたサービス提供をケアワーカーに対して指示し、ケアワーカーは被介護者に介護サービスを提供する
4. ケアワーカーは、被介護者に実際に提供した介護サ

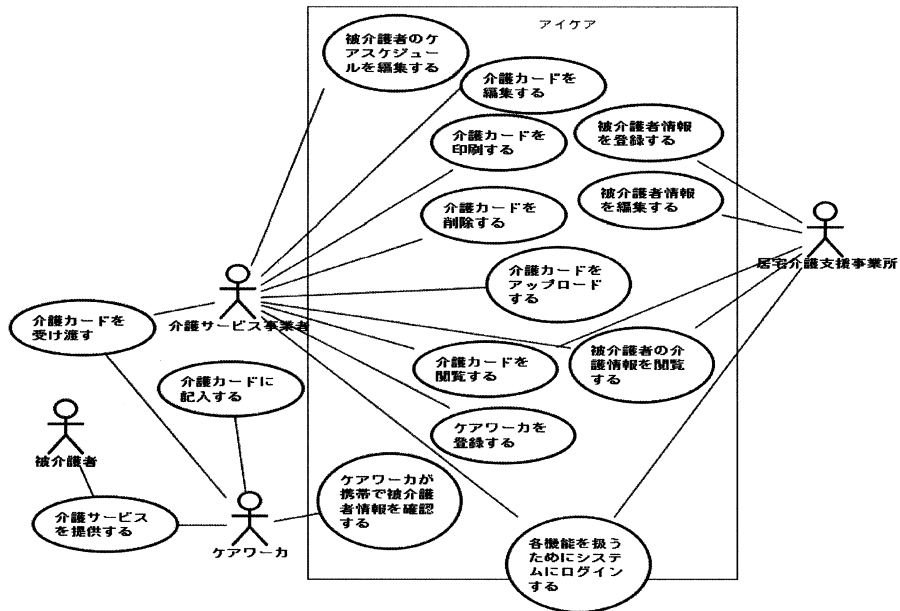


図 5 : 「アイケア」 ユースケース図

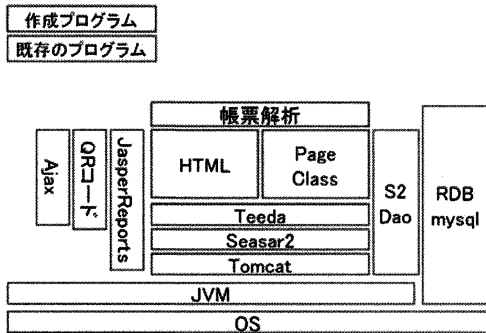


図 6 : 「アイケア」のソフトウェア構成[14]

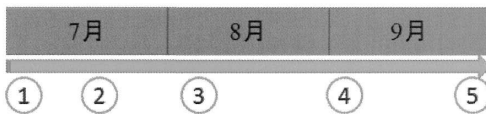


図 7 : 開発フェーズ

ービスを経過報告書等に記入し、所属するサービス事業所に報告する
この業務フローを変更することなく「アイケア」によ

って実現したものが5.2.1章で示した流れである。

「アイケア」が解決すべき要求を元に、一般的な在宅介護サービスの流れから必要な機能の抽出を行った結果、システムの必要とする要件を示したユースケースをユースケース図(図 5)に示す。

- ・ 被介護者情報を登録する
- ・ 被介護者情報を編集する
- ・ 被介護者のケアスケジュールを編集する
- ・ 介護カードを編集する
- ・ 介護カードを印刷する
- ・ 介護カードを削除する
- ・ 介護カードをアップロードする
- ・ 介護カードを閲覧する
- ・ 被介護者の介護経過を閲覧する
- ・ ケアワーカーを登録する
- ・ ケアワーカーが携帯で被介護者の情報を確認する
- ・ 各機能を扱うためにシステムにログインする

これらのユースケースを元に開発を行った。図 6に「アイケア」のソフトウェア構成図を示す。

5.3. 開発の流れ

「アイケア」システムの開発フェーズを図 7に示す。

7月初旬に開発がスタートした。

フェーズ① (7月初旬) : この時点では詳細な要求定

義は終わっておらず、一般的なウォーターフォール型の開発手法では、9月でのシステム完成は難しい状態であった。

フェーズ② (7月中旬)：現状で出来ているユースケース図等を元に、UIモックの作成を行った。これによってユーザの要求を固めることができ、詳細な要求定義を行うことができた。

フェーズ③ (8月初旬)：UIモックを元にそれに対するPageクラスを作成した。ここでPageクラスはビジネスロジックを書くことに集中できるので効率よくコーディングができた。

フェーズ④ (9月初旬)：システムは順調に作られていたが、ここで介護データをグラフ表示する機能の追加を行った。TeedaのAjaxライブラリを用いることで普通にAjaxを用いるよりも効率よく機能を追加できた。

フェーズ⑤ (9月末)：実証実験で使用できるレベルのシステムを完成させることができた。

5.4. 仕様変更への対応

図4に示すように、実証実験中に複数回の仕様変更を行った。今回作成したシステムは実証がより有効に行われるように、実証先の要望を踏まえてシステムを洗練させる必要があった。

【仕様変更の例】

仕様変更の1つとして、「スケジュールが未定の介護カード作成」を挙げる。もともと、介護カード作成時に日程を決めて介護カードを作成するという仕様のみであったが、日程の確定していないカードを予め作成するという機能を追加する仕様変更を行った。この仕様変更に対してはまずカレンダー表示のHTMLの画面上に「空の介護カード作成」ボタンを新たに作成した。そして、そのボタンのリンク先として空の介護カード作成ページのHTMLを新たに作成した。これらの作業をHTMLテンプレート機能の活用を中心として行い、効率よく仕様変更を行うことができた。

6. 考察

要求が未定義なシステムを開発において工期が短い場合、UIを中心とした開発が必要であると考えられる。これを実践する方法として、TeedaのHTMLテンプレート機能が有効であると考えられる。これを用いれば、仕様を固めつつ、UIを作成しなおすことなく開発できる。また、軽量コンテナにより柔軟な開発が可能である。

今回、在宅介護支援システムの開発をSeasar2によ

り行った。介護業務の現場ではIT化が進んでいないところもあり、情報機器やシステムへの理解が十分であるとは言えないユーザの要求を明確化する必要があったため、開発スタート時(フェーズ①)に仕様が固まっておらず、ウォーターフォール型での開発は難しい状況にあった。しかし、開発フェーズ②で示したように先にUIモックを作成してユーザのレビューを逐次行うことで「アイケア」に必要な機能を抽出し仕様を固めることに成功し、開発に一定の目的を付けることができた。また、フェーズ③にあるようにあらかじめHTMLをUIモックで作成していたので、ビジネスロジックのみ書くことに集中することができ、効率よくコーディングが行えた。開発時や実証実験時での機能追加にもフェーズ④にあるようにTeedaのAjaxライブラリを用いることで効率よく機能追加できた。

以上のように3章、4章で述べたSeasar2の有用性が実際の開発で確認することができた。

7. まとめ

今日のWebアプリケーション開発は開発の効率化が求められている。そんな中、アジャイル的なソフトウェア開発を行う手法が利用されている。システム開発においては、開発のスタート時点で仕様が固まっていない場合が多い。ユーザは発注したシステムにどのような機能が必要になるかわかっていないことが大きな原因の一つである。

我々は在宅介護の質的向上を目指した在宅介護支援システムの開発をJavaのWebアプリケーションフレームワークであるSeasar2を用いて行った。本稿ではこのシステム開発の事例を通してSeasar2の機能の1つであるTeedaのHTMLテンプレート機能と軽量コンテナである利点を生かして、顧客である介護事業所やケアワーカーの要求を固めながら柔軟にシステム開発ができ、短期間で開発を終えることができた。

なお、平成20年度は、「アイケア」システムを洗練し、複数の在宅介護・医療関連事業所の連携における有効性を示すことと、地域コンタクトセンター(図8)を構築し、「アイケア」システムを通じた現場知の共有を実現することを目標としている。そのため、介護カード方式を応用したICTバリアフリーな現場知の入力方式の開発を更に進展させ、介護・医療のステークホルダー(利害関係者)間でセキュアに情報共有と情報連携ができるようセキュリティ要素技術を応用した使いやすい方式の開発をおこなう。それと同時に、被

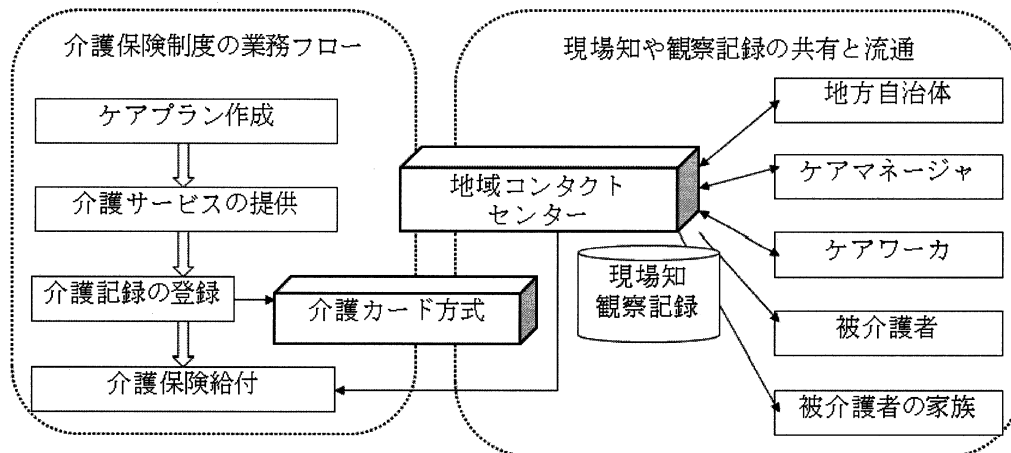


図 8：地域コンタクトセンター

介護者を中心として介護・医療の行政及び関連事業者等の複数の利害関係者によるセキュアな地域コンタクトセンターのプロトタイプを構築して、業務支援（介護）カード方式や関連情報の情報連携や情報共有の実証実験を実施する。それにより、介護・医療の地域社会におけるバリアフリーな情報共有フレームを実現する予定である。

謝辞

本研究の一部は総務省の戦略的情報通信研究開発推進制度（SCOPE）の地域ICT振興型研究開発として行われたものである。

文献

- [1] MVC モデルとは
<http://www.atmarkit.co.jp/fjava/javafaq/j2ee/j2e07.html>
- [2] Java EE
<http://sdc.sun.co.jp/java/javaee/index.html>
- [3] DI×AOP コンテナ「Seasar2 と Spring」
<http://www.thinkit.co.jp/free/compare/15/1/>
- [4] なぜ DI コンテナを使うのか
http://www.atmarkit.co.jp/fjava/rensa3/springdi04/springdi04_1.html
- [5] 今だから聞きたい、DI×AOP 入門
http://mediacast.sun.com/users/getappa/media/Introduction_of_DIxAOP.pdf
- [6] 須賀幸次, 木村聡, 西川麗, 高安厚思, 白井博章, 椎野峻輔, 岡薫, 藤村浩士 著, ひがやすを 監修, 「Seasar 入門 はじめての DI&AOP」, ソフトバンク クリエイティブ, 2006, p. 6, 57, 58, 61
- [7] 今必要な人のための演習 AOP でプログラムをシンプルにする
<http://itpro.nikkeibp.co.jp/article/COLUMN/20070725/278220/>
- [8] Spring Framework
<http://www.techscore.com/tech/Others/Spring/index.html>
- [9] The Seasar Project
<http://www.seasar.org/>
- [10] Teeda
<http://teeda.seasar.org/ja/>
- [11] 厚生労働省：介護・高齢者福祉
<http://www.mhlw.go.jp/index.html>
- [12] 和田成晃, 小出哲久, 矢口隆明, 白石善明, 岩田 彰, “介護業務支援システム「アイケア」の開発”, 電子情報通信学会 OIS 研究会, 2008 年 7 月 (予定)
- [13] 「介護報酬の解釈 サービス提供と制定の実際 平成 15 年 4 月版」, 社会保険研究所, pp.72-73
- [14] JasperReports
http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/