

在宅高齢者を対象とした困り事・対処法共有サービスの実装

雲丹亀和希[†] 佐伯 幸郎[†] 中村 匡秀^{†,††} 安田 清^{†††}

[†] 神戸大学 〒657-8501 神戸市灘区六甲台町 1-1

^{††} 理化学研究所・革新知能統合研究センター 〒103-0027 東京都中央区日本橋 1-4-1

^{†††} 大阪工業大学 〒535-8585 大阪市旭区大宮 5-16-1

E-mail: [†]uni-kaz@ws.cs.kobe-u.ac.jp, ^{††}sachio@carp.kobe-u.ac.jp, ^{†††}masa-n@cs.kobe-u.ac.jp,

^{††††}fwkk5911@mb.infoweb.ne.jp

あらまし 在宅高齢者の自助や互助を支援するため、我々の研究グループでは、在宅高齢者を対象とした困り事・対処法共有サービス Compass4SL (Community-based Problem and Solution Sharing Service for Senior Living) の研究開発を進めている。しかしながら、現状の Compass4SL は PoC (Proof of Concept) 用のプロトタイプであり、在宅高齢者が実際に使用できる状態になっていない。そこで本研究では、Compass4SL のシステム化を行うことを目的とし、以下の手順に従ってシステムを設計・実装する。まず、Compass4SL に必要なユースケースやエンティティを洗い出し、全体のシステム構成をレイヤード・アーキテクチャで設計する。次に、得られた設計に基づき、Compass4SL を Java および Spring Boot フレームワークを活用した、Web アプリケーションとして実装する。

キーワード Compass4SL, Web アプリケーション, ユースケース, ドメインモデリング, レイヤード・アーキテクチャ

Implementation of Problem and Solution Sharing Service for the Elderly at Home

Kazuki UNIGAME[†], Sachio SAIKI[†], Masahide NAKAMURA^{†,††}, and Kiyoshi YASUDA^{†††}

[†] Kobe University 1-1 Rokkodai-cho, Nada-ku, Kobe-shi, Hyogo-ken 657-8501, Japan

^{††} Riken AIP 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan

^{†††} Osaka Institute of Technology 5-16-1 Omiya, Asahi-ku, Osaka 535-8585, Japan

E-mail: [†]uni-kaz@ws.cs.kobe-u.ac.jp, ^{††}sachio@carp.kobe-u.ac.jp, ^{†††}masa-n@cs.kobe-u.ac.jp,

^{††††}fwkk5911@mb.infoweb.ne.jp

Abstract In order to support self-help and mutual-help of the elderly at home, our research group has been working on developing Compass4SL (Community-based Problem and Solution Sharing Service for Senior Living), a problem and solution sharing service for the elderly at home. However, the elderly cannot use the service because the current service is a prototype for PoC (Proof of Concept). Therefore, this research aims systemizing Compass4SL. We design and implement the system according to the following steps. First, we find the use cases and entities. Then, we design the whole system structure as a layered architecture. Finally, we implement the system as a web application based on obtained design using Java and Spring Boot framework.

Key words Compass4SL, Web Application, Use Case, Domain Modeling, Layered Architecture

1. はじめに

現在、我が国は超高齢社会に突入している。内閣府が公表した高齢社会白書 [1] によると、65 歳以上の人口は、2019 年には約 3,589 万人に達し、総人口に占める割合は 28.4% となっている。また、高齢者の割合は増加傾向にあり、2025 年には総人口の約 30% に及ぶと推計されている。このような高齢化に伴

い、介護職員の不足、社会保障費の増加といった課題が存在している。このような背景から、政府は高齢者の在宅生活・在宅介護の支援に力を入れており、厚生労働省が推進している地域包括ケアシステム [2] における、自助や互助への取り組みが重要視されている。

在宅高齢者には、身体機能や認知機能の低下によって様々な生活上の困り事が生じる。自助や互助の観点からは、在宅高齢

者が自分で困り事に対処する、あるいは、高齢者同士が互いに助け合う機会を創出する必要がある。しかしながら、困り事への一元化された対処法の基盤の薄さ、都市化や核家族化などによる高齢者の孤独化といった課題が存在することから、高齢者にとって自助・互助を実践することは困難である。

これらの課題を解決するために、先行研究[3]では、在宅高齢者を対象とした困り事・対処法共有サービス **Compass4SL** を提案している。Compass4SL は、在宅生活における様々な困り事とそれらに対する対処法を Web 上で共有するサービスであり、在宅高齢者の**自助支援・互助支援**を目的としたサービスである。

これらの目的を実現するために、Compass4SL は、F1（対処法を共有する）、F2（困り事を共有する）、F3（対処法を実践・評価する）の3つの機能で構成される。先行研究[3]では、Compass4SL の PoC（Proof of Concept）を試作し、医療・介護従事者にヒアリングを行った結果、サービスに対して一定の満足度や価値が期待できることが分かった。しかしながら、現状の Compass4SL の実装は PoC 用のプロトタイプであり、在宅高齢者や介護者が実際に使用できるシステムにはなっていない。

そこで本研究では、Compass4SL のシステム化を行うことを目的とする。具体的には、Compass4SL の3つの機能 F1、F2、F3 を実現するために、A1（ユースケース定義）、A2（ドメインモデリング）、A3（アーキテクチャ設計）、A4（システム実装）の4つのアプローチに従ってシステム化を行う。

2. 先行研究：Compass4SL

2.1 解決したい課題

我が国の高齢化率は年々増加傾向にあり、高齢者を支える介護職員や社会保障費は慢性的に不足している[1]。このような背景から、日本政府は高齢者介護の施策に関して、従来の施設介護から**在宅介護へのシフト**を進めている。厚生労働省が推進している地域包括ケアシステム[2]の骨子として、4つの助、すなわち「自助」「互助」「共助」「公助」が掲げられているが、少子高齢化や財政状況を考慮すると「共助」「公助」の拡充は難しく、「自助」「互助」を意識した取り組みが一層重要となる[4]。

在宅高齢者には、身体機能や認知機能の低下によって、様々な生活上の困り事が生じる。これらの困り事が生じた時、高齢者が自助や互助で対処することは容易ではない。我々は以下の2つの課題を考えている。

（課題 P1）一元化された対処法の基盤の薄さ 在宅生活における困り事への対処法がまとめられた基盤が発展・普及していない。したがって、在宅高齢者は、生活上の困り事が生じた時に、対処法や困り事の相談先がすぐに分からない。また、対面で人に相談しづらい困り事の場合、対処法を得ることが難しい。対処法をインターネットで検索する場合、大量の情報から自分に合った対処法を取捨選択することが困難である。

（課題 P2）高齢者の孤独化 都市化や核家族化などにより、高齢者の孤独化が進行しており、近所付き合いやコミュニティの繋がりが減少している。こうした仲間づくりの機会が減少することで、困り事とその対処法に関する経験や知恵を共有するこ

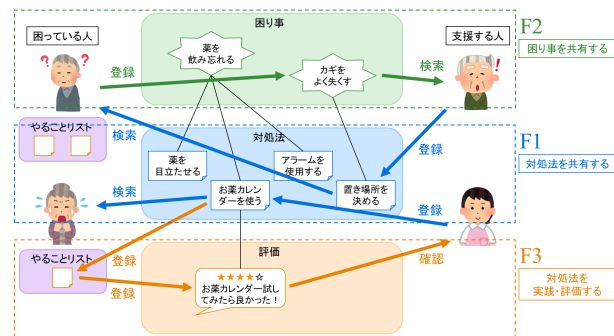


図1 Compass4SLの概要

とが困難になっている。

2.2 Compass4SLの概要

第2.1節に挙げた課題を解決するために、我々の研究グループでは、先行研究[3]において、在宅高齢者を対象とした困り事・対処法共有サービス **Compass4SL**（Community-based Problem and Solution Sharing Service for Senior Living）を提案している。Compass4SL は、高齢者の在宅生活においてよく発生する困り事とその対処法を、**集合知**として共有する**プラットフォーム・サービス**である。より具体的には、Web上のコミュニティで以下の2点を実現することを目的にしている。

自助支援 困り事を持つ人が、自分の状況にあった対処法を自力で検索し、解決できるようにする。

互助支援 困り事への対処法を知っている人が、アイデアやツールを共有し、互いに助け合う機会を創出する。

図1にCompass4SLの機能の概要を示す。Compass4SLには2種類のユーザが存在する。図の左端にいる「困り事を抱えているユーザ」（**困っている人**と呼ぶ）と、右端にいる「対処法を知っているユーザ」（**支援する人**と呼ぶ）である。Compass4SLはこれらの2種類のユーザの間に存在し、困り事とその対処法の共有を行う場を提供する。具体的には、以下の3種類の機能を提供する。

（機能 F1）対処法を共有する 困っている人が対処法を検索するとともに、支援する人が自分の知っている対処法をシステムに登録する機能。これによって、困り事に対処法を、支援する人から困っている人に伝えることができる。

（機能 F2）困り事を共有する 困っている人が自分の困り事をシステムに登録するとともに、支援する人がシステムに登録されている困り事を確認する機能。これによって、どのような困り事が存在しているのかを、困っている人から支援する人に伝えることができる。

（機能 F3）対処法を実践・評価する 困っている人が、登録されている対処法を実践し、それが役に立ったかどうかを評価する機能。困っている人は、複数の対処法から自分に合いそうな対処法を「やることリスト」に登録する。そして、リストに登録した対処法を実践し、その効果を評価することで、対処法の有効性・信頼性をコミュニティで共有する。また、支援する人が、自分の登録した対処法が誰かの役に立ったことを確認することで、コミュニティ内のコミュニケーション促進や、対処法



図2 Compass4SLのプロトタイプ画面

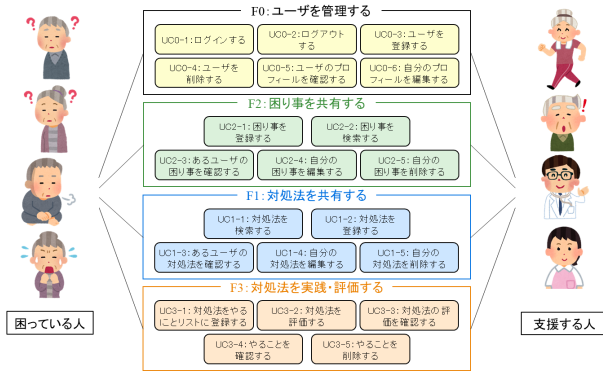


図3 Compass4SLのユースケース図

登録へのモチベーション向上につながる。

2.3 開発の現状

先行研究 [3] では、Compass4SL の PoC (Proof of Concept) を試作した。図 2 に PoC 用のプロトタイプ画面を示す。この画面は、困っている人の操作画面であり、「困り事」や「やること一覧」と書かれたボタンが見える。画面操作を容易化するために、キャラクターに扮したバーチャルエージェント (VA) [5] が説明する仕様になっている。

先行研究 [3] においては、Compass4SL の目的や意義を、このプロトタイプ画面の動画とともに、医療・介護従事者に説明し、意見を伺った。その結果、サービスに対して一定の満足度や価値が期待できることが分かった。

しかしながら、現状の Compass4SL の実装は PoC 用のプロトタイプであり、在宅高齢者や介護者が実際に使用できるシステムにはなっていない。Compass4SL のシステム化、および実際の現場での運用・評価はこれからの課題である。

3. Compass4SL のシステム化

3.1 ユースケース定義

3.1.1 ユースケース図

Compass4SL で実現すべき機能を明らかにするために、ユースケースを定義する。また、システムに外部から関与するアクタを定義し、どのアクタがどのユースケースに関与するかを明らかにする。システムが持つ全ユースケースと、アクタとの関与を明確化するために、ユースケース図を描く。

図 3 に Compass4SL のユースケース図を示す。図において、角が丸い四角は各ユースケースを表している。ユースケースは第 2.2 節で述べた Compass4SL の機能 F1, F2, F3 に従ってグループ化されている。これらに加えて、ユーザを管理する機能

F0 に関するユースケースも存在する。

以後の節で、機能 F0, F1, F2, F3 のそれぞれのユースケースの詳細を述べる。

3.1.2 F0: ユーザを管理する

Compass4SL では、困っている人と支援する人がユーザとなって、システム (Web アプリケーション) にログインし、サービスを利用する。したがって、システムのログイン、ログアウトに加えて、ユーザの新規登録や登録情報の変更に関するユースケースがある。

UC0-1: ログインする ユーザ (困っている人や支援する人) が、Compass4SL にログインする。

UC0-2: ログアウトする ユーザが、Compass4SL からログアウトする。

UC0-3: ユーザを登録する ユーザが、ユーザ ID、パスワード、ニックネーム、生年月日、性別といった項目を入力し、自分のユーザ情報をシステムに登録する。

UC0-4: ユーザを削除する ユーザが、自分のアカウント情報をシステムから削除する。

UC0-5: ユーザのプロフィールを確認する ユーザが、ある特定のユーザを選択すると、システムは、選択したユーザのプロフィールを取得し、画面に表示する。

UC0-6: 自分のプロフィールを編集する ユーザが、自分のプロフィールの内容を編集する。

3.1.3 F1: 対処法を共有する

Compass4SL の機能「F1: 対処法を共有する」に関連するユースケースを述べる。主要なユースケースとして、困っている人が対処法を検索する、および支援する人が対処法を登録する、の 2 つがある。また、支援する人は、自分が登録した対処法を編集したり削除したりすることが可能である。

UC1-1: 対処法を検索する 困っている人が、自分の困り事に基づいてシステムを検索すると、システムは、困り事に紐づけられている対処法をすべて取得し、画面に表示する。検索は、システムが用意する困り事のカテゴリ、または、困り事に含まれるキーワードに基づいて行う。

UC1-2: 対処法を登録する 支援する人が、困り事に対して自分の知っている対処法をシステムに登録する。登録においては、対処法のタイトル、対処する困り事、対処法の詳細説明、出典、リソース (対処に用いる道具や機械、ツール等のモノを指す) を入力する。

UC1-3: あるユーザの対処法を確認する 支援する人、または、困っている人が、ある特定のユーザを選択すると、システムは、選択したユーザが登録したすべての対処法を取得し、画面に表示する。対処法をユーザで検索することにより、そのユーザの他の対処法を参考にできる。

UC1-4: 自分の対処法を編集する 支援する人が、自分がシステムに登録した対処法の内容を編集する。

UC1-5: 自分の対処法を削除する 支援する人が、自分がシステムに登録した対処法を、システムから削除する。

3.1.4 F2: 困り事を共有する

Compass4SL の機能「F2: 困り事を共有する」に関連するユース

スペースを述べる。主要なユースケースとして、困っている人が自分の困り事を登録する、および支援する人が困り事を検索する、の2つがある。また、困っている人は、自分が登録した困り事を編集したり削除したりすることが可能である。

UC2-1：困り事を登録する 困っている人が、自分の困り事をシステムに登録する。登録前に、同様の困り事がシステムに存在しないことを確認したうえで、困り事のタイトル、困り事の詳細説明、困り事カテゴリといった項目を入力して登録する。

UC2-2：困り事を検索する 支援する人が、システムに登録されている困り事を、キーワードやカテゴリで検索する。

UC2-3：あるユーザの困り事を確認する 支援する人、または、困っている人が、ある特定のユーザを選択すると、システムは、選択したユーザが登録したすべての困り事を取得し、画面に表示する。

UC2-4：自分の困り事を編集する 困っている人が、自分がシステムに登録した困り事の内容を編集する。

UC2-5：自分の困り事を削除する 困っている人が、自分がシステムに登録した困り事を、システムから削除する。

3.1.5 F3：対処法を実践・評価する

Compass4SLの機能「F3：対処法を実践・評価する」に関連するユースケースを述べる。主要なユースケースとして、困っている人が実践する対処法をやることリストに登録する、困っている人が対処法を評価する、支援する人が自分の対処法への評価を確認する、の3つがある。また、困っている人は、自分がやることリストに登録した対処法の内容を確認したり、リストから削除したりすることが可能である。

UC3-1：対処法をやることリストに登録する 困っている人が、対処法を実践するために、対処法を自分のやることリストに登録する。

UC3-2：対処法を評価する 困っている人が、やることリストに登録した対処法を実践し、その評価をシステムに登録する。登録においては、1から5までの評価値とコメントを入力する。

UC3-3：対処法の評価を確認する 支援する人が、自分がシステムに登録した対処法の中から、対処法への評価を見たいものを選択すると、システムは、選択した対処法への評価をすべて取得し、画面に表示する。

UC3-4：やることを確認する 困っている人が、やることリストに登録した対処法の中から、内容を見たいものを選択すると、システムは、選択した対処法の内容を画面に表示する。

UC3-5：やることを削除する 困っている人が、自分がシステムに登録したやることリストへの登録情報を、システムから削除する。

3.2 ドメインモデリング

3.2.1 ドメインモデル図

前節でのユースケース記述に基づいて、エンティティを発見し、それらの間の関連を定義して、**ドメインモデル図**にまとめる。図4にCompass4SLのドメインモデル図を示す。この図において、四角はエンティティ（あるいはドメイン・クラス）を表し、システムで管理（生成、読み出し、更新、削除）すべきモノやコトの情報を指す。エンティティ間の線は**関連**を表し、線

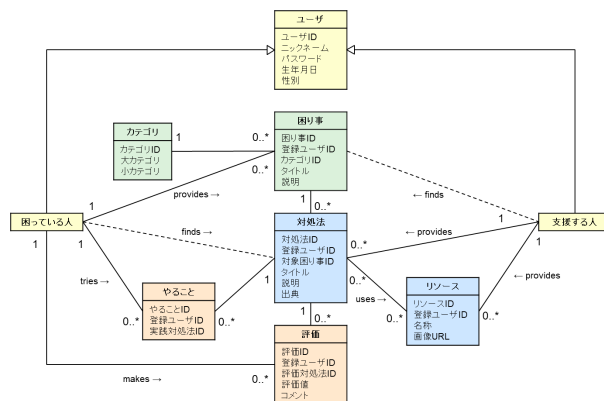


図4 Compass4SLのドメインモデル図

に付与された文字列は関連名を表す。白抜き矢印は派生関係を表している。関連の両端に付与された数字は**多重度**を表し、一方のエンティティのインスタンス（オブジェクト）を1としたとき、他方のエンティティのインスタンスが何個関連付けられるかを表す。

図4において、黄色のエンティティ群は、図3の「F0：ユーザを管理する」の機能に使用するものである。困っている人、および支援する人は、ともにユーザの派生クラスとなっている。

図4中の青色のエンティティ群は、主に図3の「F1：対処法を共有する」の機能に使用するものである。各対処法は、1つの困り事に紐づいており、0以上複数のリソースを使用することがある。逆に、1つのリソースは複数の対処法で使用されることがある。対処法とリソースはともに、ある支援する人によってシステムに登録される。登録された対処法は、困っている人から検索される。

図4中の緑色のエンティティ群は、主に図3の「F2：困り事を共有する」の機能に使用するものである。1つの困り事には、検索のためのカテゴリが紐づいている。また、1つの困り事には複数の対処法が存在する。困り事は、ある困っている人によってシステムに登録される。登録された困り事は、支援する人から検索される。

図4中の橙色のエンティティ群は、主に図3の「F3：対処法を実践・評価する」の機能に使用するものである。ある困っている人がある対処法を実践しようと決めたら、やることとしてリストに登録する。やることは、1人の困っている人と1つの対処法を関連付ける役割を果たす。また、評価は、1つの対処法と1人の評価者を関連付ける。1人の困っている人は、複数の対処法に対して評価をつけることができ、1つの対処法には異なる人によって複数の評価がつけられる。

以後の節では、各エンティティの属性について詳細に説明する。

3.2.2 ユーザ関連エンティティ

図5にユーザエンティティのデータスキーマを示す。下線付きの属性は、エンティティの識別子（**主キー**）を表す。

ユーザ (User) は、Compass4SLを利用する人の個人情報を持つエンティティである。uidはユーザを識別する主キーであり、nicknameはCompass4SL上で表示されるユーザ名である。

また、password はユーザが Compass4SL にログインするためのパスワードであり、暗号化されている。birthday はユーザの年齢を計算するために使用する生年月日であり、sex はユーザの性別（男性 = 0、女性 = 1）である。

3.2.3 対処法関連エンティティ

図 6 に、対処法に関連するエンティティのデータスキーマを示す。

対処法 (Solution) は、対処法の内容を持つエンティティである。solutionId は対処法を識別する主キーであり、userId は対処法を登録したユーザの ID、problemId は対処法の対象となる困り事の ID である。また、title は対処法のタイトル、description は対処法の詳細説明、source は対処法の出典である。

リソース (Resource) は、対処法で利用する道具の内容を持つエンティティである。resourceId はリソースを識別する主キーであり、userId はリソースを登録したユーザの ID である。また、name はリソースの名称であり、imageURL にはリソース画像の URL が保持される。

図 6 において、対処法とリソースとの間に、**割当情報 (Assignment)** というエンティティが存在する。これは、図 4 における対処法とリソースの多対多関連を実現するためのものである。すなわち、どの対処法に対して、どのリソースが登録されているかという情報を保持する。assignmentId は割当情報を識別する主キーであり、solutionId と resourceId にはそれぞれ、関連付ける対処法の ID とリソースの ID が保持される。

3.2.4 困り事関連エンティティ

図 7 に、困り事に関連するエンティティのデータスキーマを示す。

困り事 (Problem) は、困り事の内容を持つエンティティである。problemId は困り事を識別する主キーであり、userId は困り事を登録したユーザの ID である。また、categoryId は困り事が属するカテゴリの ID である。title は困り事のタイトルであり、description は困り事の詳細説明である。

カテゴリ (Category) は、カテゴリの内容を持つエンティティである。categoryId はカテゴリを識別する主キーである。また、mainCategory と subCategory にはそれぞれ、「記憶」カテゴリの中の「各種忘れ」のように、大カテゴリと小カテゴリが保持される。

3.2.5 評価関連エンティティ

図 8 に、評価に関連するエンティティのデータスキーマを示す。

やること (Todo) は、どのユーザが、どの対処法を実践するかという情報を持つエンティティである。todoId はやることを識別する主キーである。また、userId と solutionId にはそれぞれ、対処法を実践するユーザの ID と実践する対処法の ID が保持される。

評価 (Evaluation) は、対処法に対する評価の内容を持つエンティティである。evaluationId は評価を識別する主キーである。また、userId と solutionId にはそれぞれ、評価を登録したユーザの ID と評価する対処法の ID が保持される。value は 1~5 の評価値、comment は評価のコメントである。

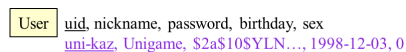


図 5 ユーザ関連エンティティのデータスキーマ

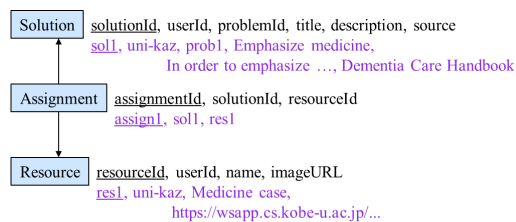


図 6 対処法関連エンティティのデータスキーマ

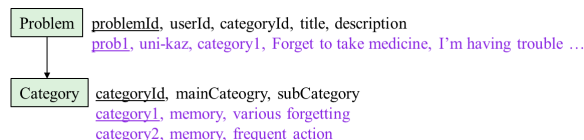


図 7 困り事関連エンティティのデータスキーマ

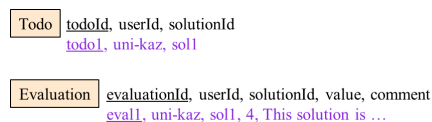


図 8 評価関連エンティティのデータスキーマ

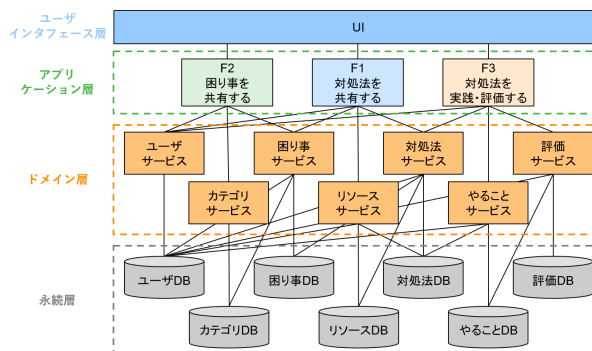


図 9 Compass4SL のアーキテクチャ

3.3 アーキテクチャ設計

前節で述べた各種のエンティティをサービスから操作するための全体のシステム構成を設計する。Compass4SL は Web アプリケーションとして実装することを想定しているため、Web アプリケーションで典型的に用いられるレイヤード・アーキテクチャ [6] を採用した。図 9 に Compass4SL のアーキテクチャを示す。図に示す通り、Compass4SL は以下の 4 層で構成される。

ユーザインタフェース層は、ユーザとの対話を行う画面を提供する。一般的に Web アプリケーションでは、各画面を HTML で定義された画面要素、CSS によるレイアウト・装飾、JavaScript による振る舞いで構成する。

アプリケーション層は、UI からの入力を受け、ドメイン層のサービス呼び出し、処理結果を UI に返す。Web アプリケーションでは、UI から HTTP リクエストを受け付け、対応するドメインサービス呼び出す。



図 10 Compass4SL トップメニュー画面

Copyright 2009-2018 Nagoya Institute of Technology (MMDAgent Model “Mei”)

ドメイン層は、各ユースケースを実行する業務ロジックを定義する。アプリケーション層から呼び出されると、ユースケースに従って下位のデータベースを操作し、エンティティの読み書きを実行する。

永続層は、ドメインモデルにおける各種エンティティのデータを保存(永続化)する。

4. システム実装

4.1 利用した技術

Compass4SL の実装に用いた技術は以下の通りである。

- 開発言語：Java1.8, HTML5, CSS3, JavaScript
- データベース：MySQL 5.7.31 [7], Nextcloud [8]
- Web アプリフレームワーク：Spring Boot 2.4.0 [6]
- テンプレートエンジン：Thymeleaf [9]
- JavaScript ライブラリ：jQuery 3.5.1, Moment.js
- Web サーバ：Apache Tomcat 9.0.33

4.2 画面の説明

本節では、実装した Compass4SL の画面について述べる。画面には、名古屋工業大学が開発した MMDAgent「メイちゃん」[10]の画像を改変したものをバーチャルエージェント(以下、VA と表記する)として使用している。

Compass4SL のトップメニュー画面を図 10 に示す。画面左側には VA が表示されており、画面によっていくつかの画像に切り替わる。また、VA の上部にはエージェントのセリフのようなものが表示されている。画面右下には「終了」ボタンがあり、選択すると Compass4SL からログアウトする。

4.3 画面遷移

本節では、実装した Compass4SL の画面をいくつか紹介する。

対処法検索結果画面を図 11 に示す。画面に困り事に対する対処法の一覧が表示されており、内容を見たい対処法を選択すると図 12 の対処法内容画面に遷移する。ここで、「やってみる」ボタンを選択すると、画面に表示されている対処法が自分のやることリストに登録される。また、図 13 は対処法登録画面である。必要な項目を入力して「送信する」ボタンを選択すると、入力した内容の対処法がデータベースに登録される。

困り事登録画面を図 14 に示す。必要な項目を入力して「送信する」ボタンを選択すると、入力した内容の困り事がデータベースに登録される。また、図 15 は困り事をキーワードで検索した時の困り事検索結果画面である。画面に困り事の一覧が

表示されており、内容を見たい困り事を選択すると図 16 の困り事内容画面に遷移する。ここで、「対処法をみる」ボタンを選択すると、図 11 の対処法検索結果画面に、「困り事に回答する」ボタンを選択すると、図 13 の対処法登録画面に遷移する。

評価登録画面を図 17 に示す。必要な項目を入力して「送信する」ボタンを選択すると、入力した内容の評価がデータベースに登録される。また、図 12 の「評価を見る」の部分を選択すると、図 18 に示す対処法への評価一覧画面に遷移する。

5. まとめ

本稿では、先行研究[3]で提案されている、在宅高齢者を対象とした困り事・対処法共有サービス Compass4SL のシステム化を行った。Compass4SL は、F1(対処法を共有する)、F2(困り事を共有する)、F3(対処法を実践・評価する)の3つの機能で構成される。

Compass4SL のシステム化は、A1(ユースケース定義)、A2(ドメインモデリング)、A3(アーキテクチャ設計)、A4(システム実装)の4つのアプローチに従って行った。

今後は、実装した Compass4SL を在宅高齢者に使用してもらい、どれだけ在宅高齢者の支援に役立ったかについての評価を行う。また、我々の研究グループで開発している、「こころ」の見守りサービス[11]のような他のサービスと連携して、ユーザの過去の情報を取得し、ユーザに合った対処法を自動で提案する機能や、高齢者がサービスを利用しやすくするために、音声認識による文字入力、音声合成による文字読み上げといった機能の実装を行う。

文 献

- [1] “令和2年版高齢社会白書(全体版)-内閣府,” <https://www8.cao.go.jp/kourei/whitepaper/w-2020/html/zenbun/index.html>. (Accessed on 02/06/2021).
- [2] “地域包括ケアシステム | 厚生労働省,” https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/hukushi_kaigo/kaigo_koureisha/chiki-houkatsu/. (Accessed on 02/06/2021).
- [3] 高槻大貴, 佐伯幸郎, 中村匡秀, 安田清, “Compass4SL:在宅高齢者を対象とした困り事対処法共有サービス,” 電子情報通信学会技術研究報告, 第 119 巻, pp.145-150, March 2020. 沖縄, 大嶺信泉記念館.
- [4] “地域包括ケアシステム構築に向けた制度サービスのあり方に関する研究事業報告書【本編】,” https://www.murc.jp/sp/1509/houkatsu/houkatsu_01/h28_01.pdf. (Accessed on 02/14/2021).
- [5] “『鑑紀乃さん作画 ver. のりたまさん MMD モデル』 / ささかやさんの作品 - ニコニ立体,” <https://3d.nicovideo.jp/works/td28612>. (Accessed on 02/14/2021).
- [6] “Spring Boot,” <https://spring.io/projects/spring-boot>. (Accessed on 02/09/2021).
- [7] “MySQL,” <https://www.mysql.com/jp/>. (Accessed on 02/14/2021).
- [8] “Nextcloud,” <https://nextcloud.com/>. (Accessed on 02/14/2021).
- [9] “Thymeleaf,” <https://www.thymeleaf.org/>. (Accessed on 02/14/2021).
- [10] “メイ&タクミ公式ウェブサイト | 国立大学法人名古屋工業大学,” <http://mei.web.nitech.ac.jp/>. (Accessed on 02/09/2021).
- [11] 三浦雅咲, 前田晴久, 佐伯幸郎, 中村匡秀, 安田清, “在宅高齢者を対象とした「こころ」の見守りサービスの試作と予備の評価,” 電子情報通信学会技術研究報告, 第 119 巻, pp.1-6, Aug. 2019. 栃木, 帝京大学宇都宮キャンパス.

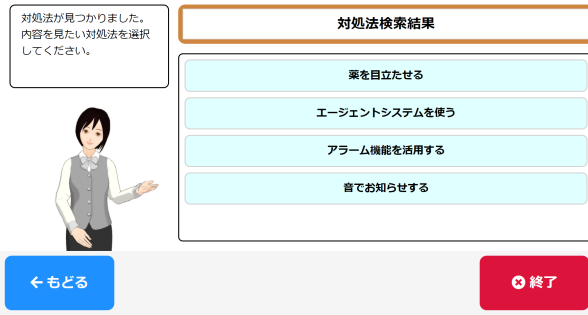


図 11 対処法検索結果画面



図 15 困り事検索結果画面

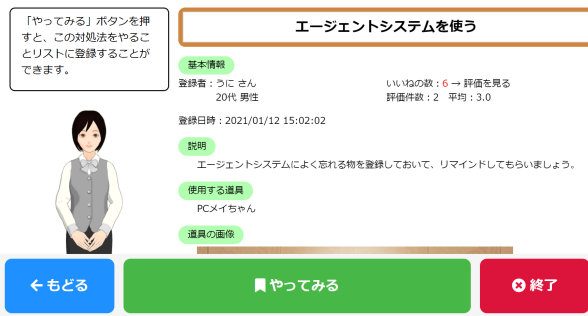


図 12 対処法内容画面

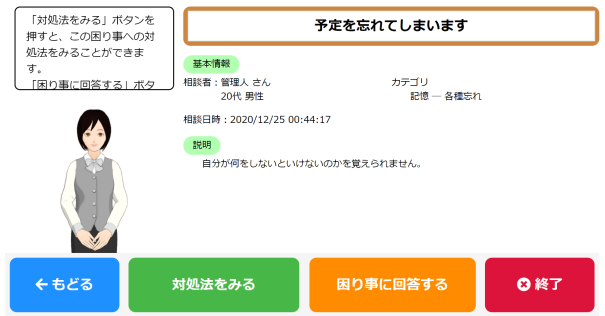


図 16 困り事内容画面



図 13 対処法登録画面



図 17 評価登録画面



図 14 困り事登録画面



図 18 評価一覧画面