2018 IEEE 22nd International Enterprise Distributed Object Computing Workshop

A Holistic Model-based Adaptive Case Management Approach for Healthcare

Felix Michel
Department of Informatics
Software Engineering for Business Information Systems
Technische Universität München
85748 Garching b. München, Germany
felix.michel@tum.de

Florian Matthes
Department of Informatics
Software Engineering for Business Information Systems
Technische Universität München
85748 Garching b. München, Germany
matthes@tum.de

***Abstract*—In recent years, personalized connected care has become increasingly important due to the generally aging population and the rising cost pressures in the healthcare sector. Nevertheless, to the best of our knowledge, there are no off-the-shelf solutions available to provide open and adaptable information and communication technology (ICT) for connected care; the reasons for this may differ between use cases. Based on our case studies at hospitals in different European countries, we identified three main challenges of such for ICT solution: 1) the high diversity between hospital sites and treatments, 2) the embedding of information from existing information systems, and 3) the coordination and communication of the many different stakeholders. Our approach is to use a full stack model-based solution that supports the integration, communication, and coordination of the pending work. Currently, our solution is being used for clinical trials.**

***Index Terms*—Modeling, Adaptive Case Management, ACM, Healthcare**

I. Introduction

This paper presents a full stack model-based adaptive case management approach customized for healthcare domain applications. The system design and development are driven by the Personalized Connected Care for Complex Chronic Patients (CONNECARE)[1] project. The objective of CONNECARE is to improve the efficiency of integrated care systems and healthcare results. A consortium exists between four clinical partners in Spain, Netherlands, and Israel that are responsible for the clinical case studies and six technical partners that provide innovative ICT solutions.

Figure 1 illustrates the conceptual CONNECARE system including the relevant stakeholders, components and communication patterns. The system has two main components, Smart Adaptive Case Management (SACM) which includes a decision support system that is used by professionals such as clinicians and nurses, and a Self-Management System (SMS) which provides mobile applications for patients and includes wearable devices for measurements. A more detailed description of the overall project objectives has been published by [1] and [2]. Other relevant projects in the domain of connected healthcare are ICT4LIFE [3], PolyCare [4], ProACT [5], and CAREGIVERSPRO-MMD.

We briefly illustrated the CONNECARE project in the previous paragraph. In Section II we summarize the main challenges of connected care that are relevant to the conceptual system design. Considering the challenges, we derive the conceptual requirements in Section III. In Section IV, we present the resulting core features based on several annotated screens of the productive system. All presented user interface features use an underlying full stack modeling engine that uses the model definitions to execute and display cases. Section V describes the model elements, maps them to their related requirements, and references papers where they are described in detail. The presented approach is evaluated based on CONNECARE clinical trial case studies in Section VI, and a short conclusion is provided in Section VII.

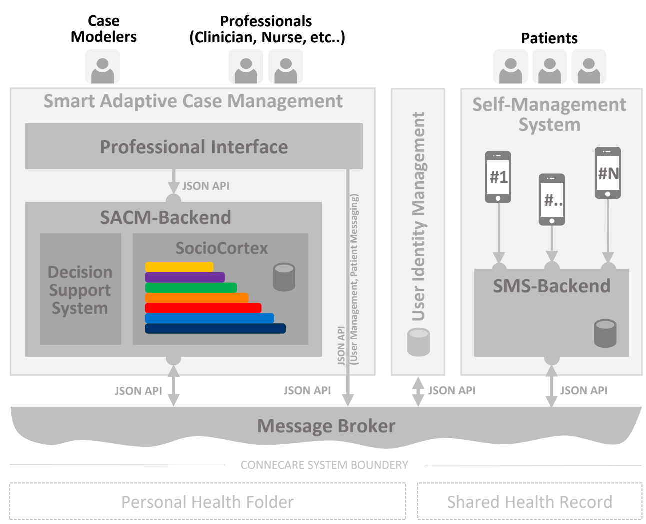

Fig. 1. Overview of the conceptual CONNECARE system architecture including the primary stakeholders. A more detail description of the colored generic SocioCortex full stack modeling engine is provided by [6].

This work has received funding from the European Research Council (ERC) under the EuropeanUnions Horizon 2020 research and innovation programme (grant agreement n 689802).

[1]http://www.connecare.eu/

2325-6605/18/$31.00 ©2018 IEEE
DOI 10.1109/EDOCW.2018.00014

II. Challenges of Integrated Care Approaches

This section summarizes the critical challenges that had a huge impact on the SACM system design. Within the CONNECARE project, several aspects are highly challenging:

- **High diversity between hospital sites and case studies** During the clinical trials, three different case studies were applied at each hospital. The hospitals are located in Barcelona (Spain), Lleida (Spain), Groningen (Netherlands), and Tel Aviv (Israel). Therefore, the variety of legal, ethical, and cultural aspects that need to be considered is high. This leads to three conceptually equal case studies that are highly customized to the individual needs of each site.

- **Embedding information of existing information systems** Integrated care requires integrated ICT solutions in order to work time and cost-effective. The current situation differs enormously from hospital site to hospital site, some using completely integrated Hospital Information Systems (HIS) others working partly paper-based. Data synchronization across systems without homogeneous taxonomies is very challenging.

- **Coordination and communication of many different stakeholders** Key of a successfully integrated care approach is communication and coordination across all involved stakeholders. Ensuring a continuous and direct communication with the patient and as well between clinicians is challenging due to the often distributed stakeholders. Work of the professionals needs to be distributed and coordinated within defined time frames. For professionals, it is challenging to monitor the patients continuously to ensure that they follow their instructions as planned. Unforeseen situations will occur that need manual intervention.

All challenges have been derived from our project case studies but they are probably generic enough to be applied to other sites or projects as well. Our experience of the case studies is that most of the challenges are covered if two different case studies are considered.

III. Requirements

Based on the challenges identified in Section II, we iteratively derived the SACM requirements. The high diversity of the different sites and the distributed project setup led to multiple agile iterations while collecting and adapting the requirements. The conceptual high-level SACM requirements are described as follows:

R1 ***Support a model-based full stack approach*** To cope with the high diversity across treatments and hospital sites, a fully model based approach should be used, and the treatment or site-specific adaptations should be applied within the related meta-model including different clinical questionnaires or languages.

R1.1 *Support data schema models* Data that are generated during the execution of the process models need to be modeled. In addition, they are needed to model data that will be integrated from third-party systems, e.g., patient data from hospital information systems.

R1.2 *Support adaptive process models* The system needs and support to define adaptive treatment plans that are customized to the specific needs of the hospital and treatment. As a reference methodology for defining the processes, Adaptive Case Management (ACM) should be used. To support integrated care, these processes need to be synchronized with other subsystems.

R1.3 *Support role-based access right models* The system needs to support granular role-based access control mechanisms to define which clinicians are allowed to access which patient data. In addition, clinical tasks need to be assigned based on roles.

R1.4 *Support simple user interface models* In general, the user interface needs to represent each model element in a generic model-based manner. To support clinical use cases, special representations need to be generated that can also be reused as a model elements.

R2 ***Support third-party system integration*** The system needs to use centralized user identity management, orchestrate processes across system boundaries, and support external data sources.

R2.1 *Support external user identity management* The system needs to provide a Single Sign-On (SSO) for professionals; therefore, external user identity management must be supported. To simplify integration, foreign identifiers should be used as internal primary keys.

R2.2 *Support process orchestration for third-party systems* To provide integrated care, the system needs to support the orchestration of external systems. External systems processes, such as processes of the SMS, should be seamlessly integrated to provide an aggregated interface for professionals.

R2.3 *Support integration of external data sources* Hospital information systems with extensive amounts of patient data exist. Therefore, the system architecture needs to consider enhancing internal data with external data sources.

R3 ***Support coordination and communication*** The system needs to support notifications, text messages, unstructured case notes, and clarify where individual contributions are needed.

R3.1 *Support notifications* The clinicians can prescribe patients certain tasks via the SMS system such as measure your blood pressure every morning. If the blood pressure exceeds a individually specified threshold, the system should notify the responsible clinician. Unforeseen technical situations should also lead to notifications of the professional users.

R3.2 *Support continuous communication* Case-based clinician-to-clinician messages need to be supported for information exchange. Therefore, all involved clinicians should be able to follow ongoing conversations. To provide integrated care, direct communication with the patient needs to be supported in a separate conversation.

R3.3 *Support case notes* In addition, to the semi-structured workflow, the system needs to provide a wiki-based notes area where clinicians can collaboratively document unstructured information as well. Individual predefined templates should be provided depending on case definitions.

R3.4 *Indicate needed contribution* A single case contains many tasks that need to be accomplished by either a professional or the patient. The system needs to support assigning tasks to professionals based on their roles. Normally, professionals have multiple cases, therefore the system needs to provide a dashboard to indicate where contributions are needed.

IV. User Interface Features

This section describes the conceptual core features that are visible on the user interface. Technical concepts and meta-models are described in Section V.

❶ **Multi-tenancy** The entire application is provided as SaaS, which means multi-tenancy support is necessary. A concept of workspaces allows all model and instance elements to be strictly encapsulated based on access rights.

❷ **Dashboard** The dashboard provides a user-specific overview of possible actions, notifications, messages, and pending tasks (see Figure 2). The first third of the screen shows all possible actions, such as navigating to the my-case view, instantiate a new case and navigating to the user management view. The second two thirds is divided into three columns that represent 1) the not yet acknowledged notifications, 2) the unseen case messages sent by clinicians or patients, and 3) the pending tasks for that user, where the overdue tasks are highlighted in red. A search bar allows to filter the dashboard elements according to patients by typing the patient name. In general, the dashboard helps the user identify pending work and directly navigate to the related views where contributions are need.

❸ **My Cases** The my-case view shows a table that lists all cases where the currently logged in user has at least read access. Information such as the case definition name, current state, patient name including the age, case owner, number of notifications per case, number of my notifications per case, number of my unread messages, number of pending tasks, and number of my pending tasks are shown for each case. A search field allows to filter the list of cases based on different patient-identifying criteria.

❹ **Case Representation** All case-related views show 1) a breadcrumb to the current location, 2) a case header that illustrates the most relevant information, such as the case definition name, patient name including the age, case identifier, and the case contextual actions, and 3) the case-related subview tabs, such as the summary page, case process, case data, case team, case notifications, case messages, and case notes (see Figure 3).

❺ **Case Summary** The summary page is a configurable dashboard (see Figure 5) for each case definition and consists of a section that links to data that have been generated by the case workflow. Every section can be placed in a grid layout of three columns. Normally, the summary page contains several special representations that are defined for the generic representation such as the presented SVG based body representation. The goal is to help clinicians quickly identify critical patient parameters.

❻ **Case Workflow** The case workflow is modeled based on ACM principles that are described by Keith Swenson [7]. The SACM user interface allows only to execute already defined models. On the top, all stages are shown that allow to nest related process elements into one stage (see Figure 3). Supported process elements are Stages, HumanTasks, AutomatedTasks, and DualTasks, which combine a HumanTask followed by an AutomatedTask. HumanTasks expect a human input on the SACM user interface, and AutomatedTasks executed in the background from the SACM user perspective, e.g., a clinical questionnaire that is completed by a nurse will be a HumanTask and a task that involves interactions with the patient via the SMS system respectively the patient's smartphone or tablet is usually a DualTask.

❻ **Flexible Process Adaption** A key principle of ACM is to allow adapting the case execution flow as needed, e.g., the work-plan stage contains only optional tasks that need to be manually activated by the clinician (see Figure 3). This allows to individually define a treatment plan for the needs of each patient.

❽ **Task Representation** All task types are represented by a title that can be dynamically adapted depending on the selected values (see Figure 4). The gray box below illustrates relevant meta-data such as, the task owner, the roles that are allowed to own this task, a due date by when the task should be completed, and a simplified state diagram. All meta-data fields can be set with previous task if linked in the data model or manually. Below the gray box, readable and writable task parameters are illustrated.

❾ **Custom Data Representation** The model-based approach uses a generic representation for all data types, i.e., one data type is bound to one representation. Nevertheless, the summary page shows a body representation that allows clinicians to quickly identify critical organs based on the previous evaluation (see Figure 5). Other examples are the color highlighting of values that are above a certain threshold or representing time series measurements as line diagram (see Figure 4). Custom representations can be used to override the default rendering behavior on the summary page and task pages.

❿ **Case Data** The workflow execution engine stores all resulting case data as linked entities that consist of attributes. In addition, existing data from hospital information systems can be integrated into the system. This results in a tree-based key-value representation similar to a file system explorer with folders and files (not yet implemented on the front-end).

⓫ **Case Team** The case team shows the groups and persons who have direct or inherited access to the case. For every case instance, readers and writers are visible. Only case writers have the right to apply changes to the case, such as manually activate or complete questionnaires and write case internal or case external messages. In addition, the case-specific roles including the currently assigned users are shown and can be modified. (only partly implemented on front-end)

⓬ **Case Notifications** The case notifications view shows all notifications related to the selected case. A notification can be generated automatically by the process engine internally when an error occurs, e.g., data should be sent to a third-party subsystem that is not available. Custom domain specific notifications can be defined as well by third-party systems e.g., an integration task exceeds predefined thresholds. Every notification is related to one process element and the owner of those process elements will receive a notification on their dashboard. A notification can be acknowledged by the user and is removed from the dashboard (see Figure 2) but is still accessible in the case notification view for accountability reasons.

⓭ **Case Messages** The case message view enables two different communication threads: 1) internal clinician-to-clinician messaging to exchange information regarding the patient without involving the patient directly and 2) external clinician-to-patient messaging, that allows direct communication with the patient if needed. All clinicians that have write access to that case can send a message. Every message needs be marked as seen by every clinician of the case individually, unseen messages are shown on the dashboard of the individual user.

⓮ **Case Notes** The case notes view allows to document unstructured clinical information related to a case, similar to a wiki page. It is possible to define an initial HTML template that is instantiated when the case is created, e.g., it is important to have a predefined treatment-specific documentation structure for each case. A WYSIWYG editor allows to modify the content by every user that has write access to the case.

⓮ **Manage Users** In general, the mange-user view is separated into professional user management and patient user management and allows to create, edit, and delete users. In addition, roles can be assigned to users. Those roles are used to allow only certain users to instantiate case definitions or to restrict the ownership of tasks to certain roles. All changes are applied directly in the central user identity management system, which propagates all changes to the connected subsystems. This allows an SSO across all systems.

V. Supported Model Elements

Our modeling environment, the SocioCortex engine evolved from a data modeling system [9] toward a fully integrated process modeling platform [6] that is used by the SACM (see Figure 1). This section describes the supported model elements in detail, and the summary in Table I illustrates their dependencies on the related requirements.

Workspace A workspace is a container that contains model definitions and model instances. Several access right levels, such as readers, writers, and contributors that can only create new instances, can be set. In our presented use case, one workspace represents one tenant (one hospital).

Group Groups are used to provide granular access control, e.g., for the workspace readers, writers, and contributors. The group element in combination with the user object represents a composite pattern that allows to nest groups in multiple levels. Users and groups are synchronized with the centralized UIM, and therefore support foreign primary keys as ids. On case instance level, groups are used to provided constraints for the ownership of tasks.

Entity Definition Entity Definitions are a container for the data schema definition based on an ordered list of attribute definitions and derived attribute definitions. The attribute definition can link itself to other entity definitions, this allows creating a tree-based linked object structure during case execution.

Attribute Definition Attribute Definitions are used to define the resulting data fields of the executed tasks. Supported types are: `string`, `longtext`, `boolean`, `number`, `enumeration`, `date`, `json`, `link` complex types that link to other entities and `notype`. Expected multiplicities such as `maximumOne`, `exactlyOne`, `atLeastOne` and `any` can be defined. In addition, constraints such as min and max for numbers, before and after for dates and dedicated entity definitions for the type link can be modeled. E.g. only nurses of Groningen hospital should be selectable for a field.

Derived Attribute Definition Derived Attribute Definitions are computed values that are used to present a final score of a medical questionnaire. Technically, a powerful model-based expression language (MxL) is used to define these queries (see [10] and [11]).

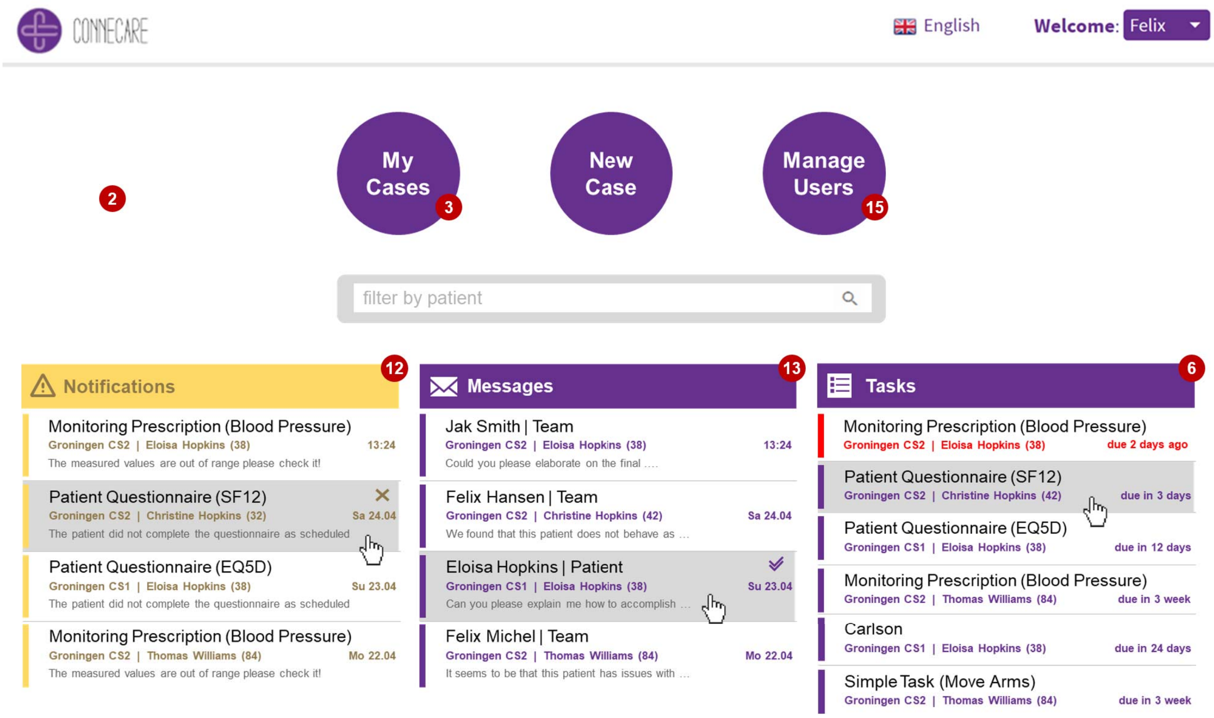

Fig. 2. The dashboard indicates where contributions are needed and links to the related notifications, messages, and tasks. All items are dynamically aggregated considering the case access rights and task ownership.

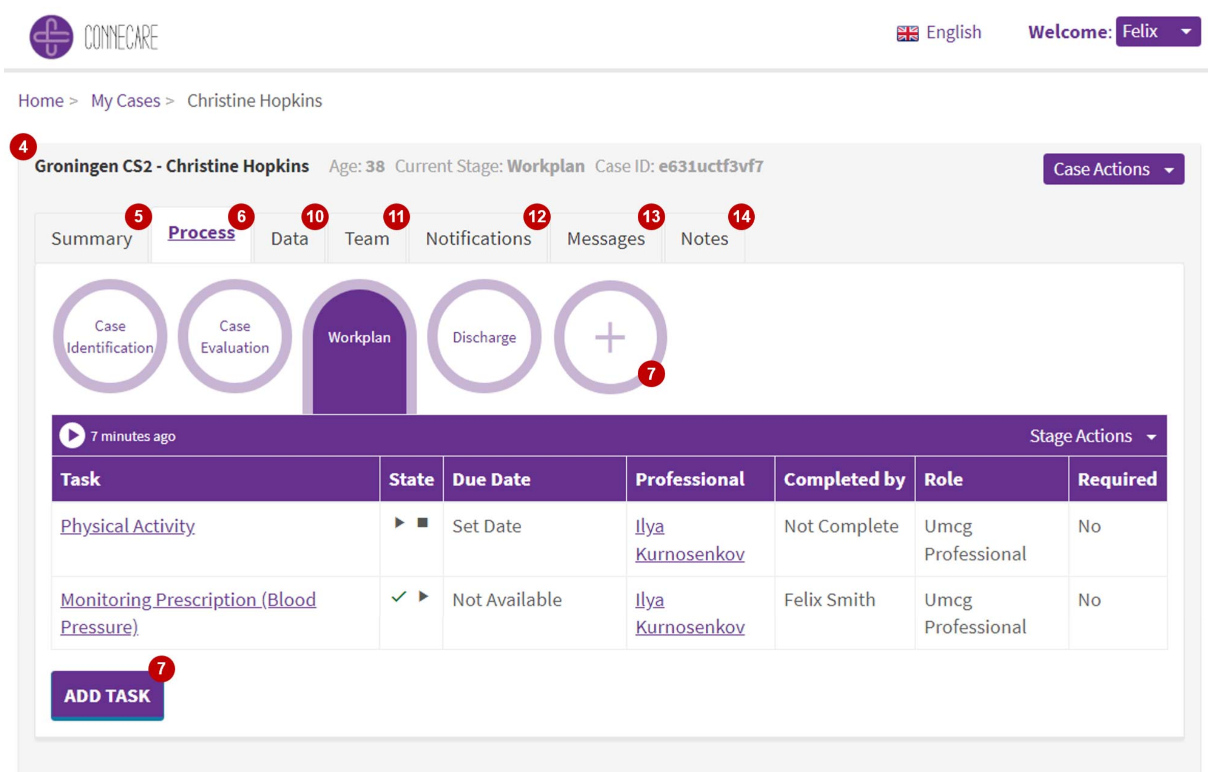

Fig. 3. Case overview showing the case workflow tab. Currently, the case identification and the case evaluation stage are completed, the workplan stage is active, and optional tasks can be manually activated and assigned to the patient who will receive the task on the SMS app.

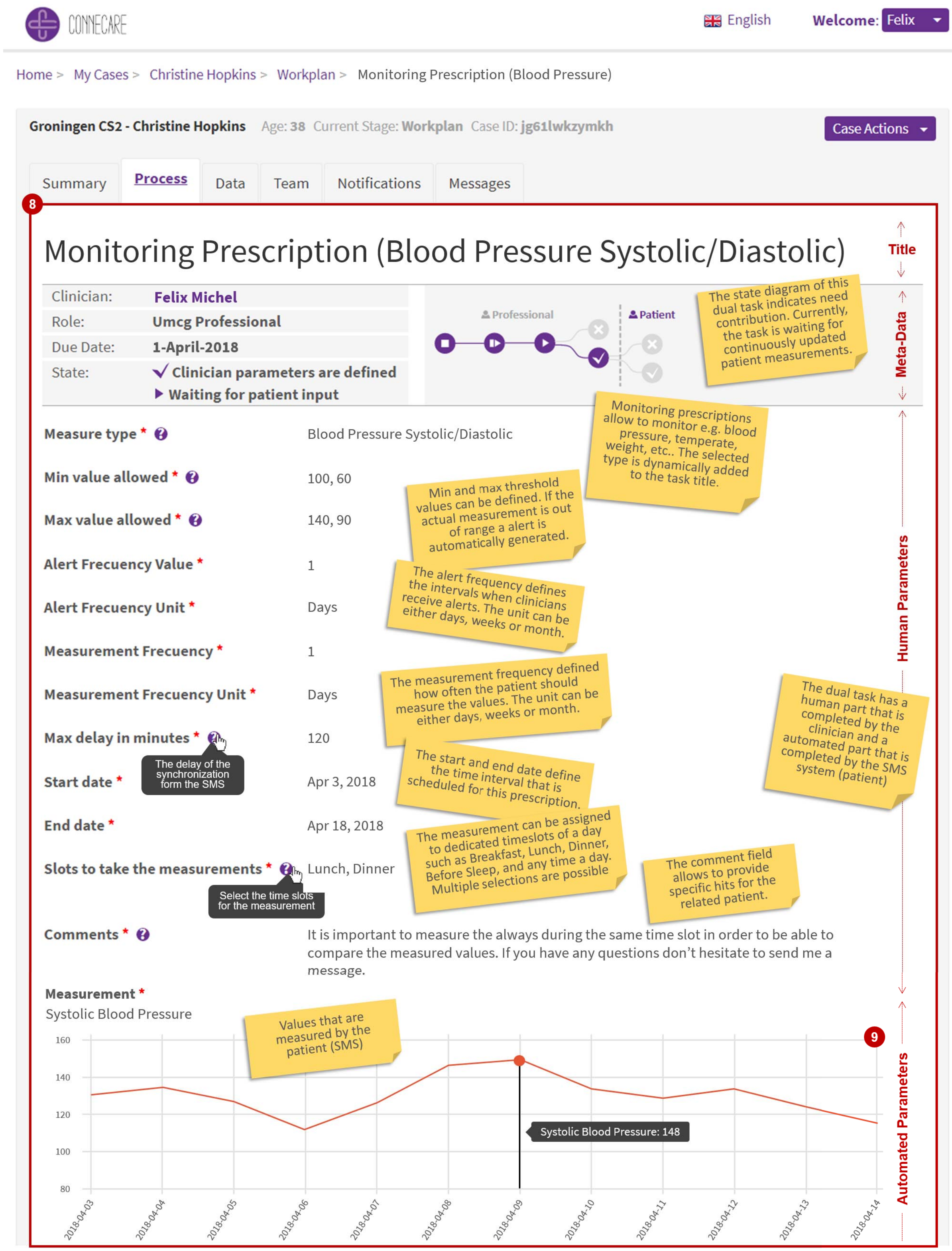

Fig. 4. The task representation shows a dual task which allows defining different patient self-monitoring tasks during the case execution. First, the clinician defines the parameters that should be monitored and then the patient measures those parameters in defined intervals. When measurements are above a individually defined threshold, notifications are generated for the clinicians.

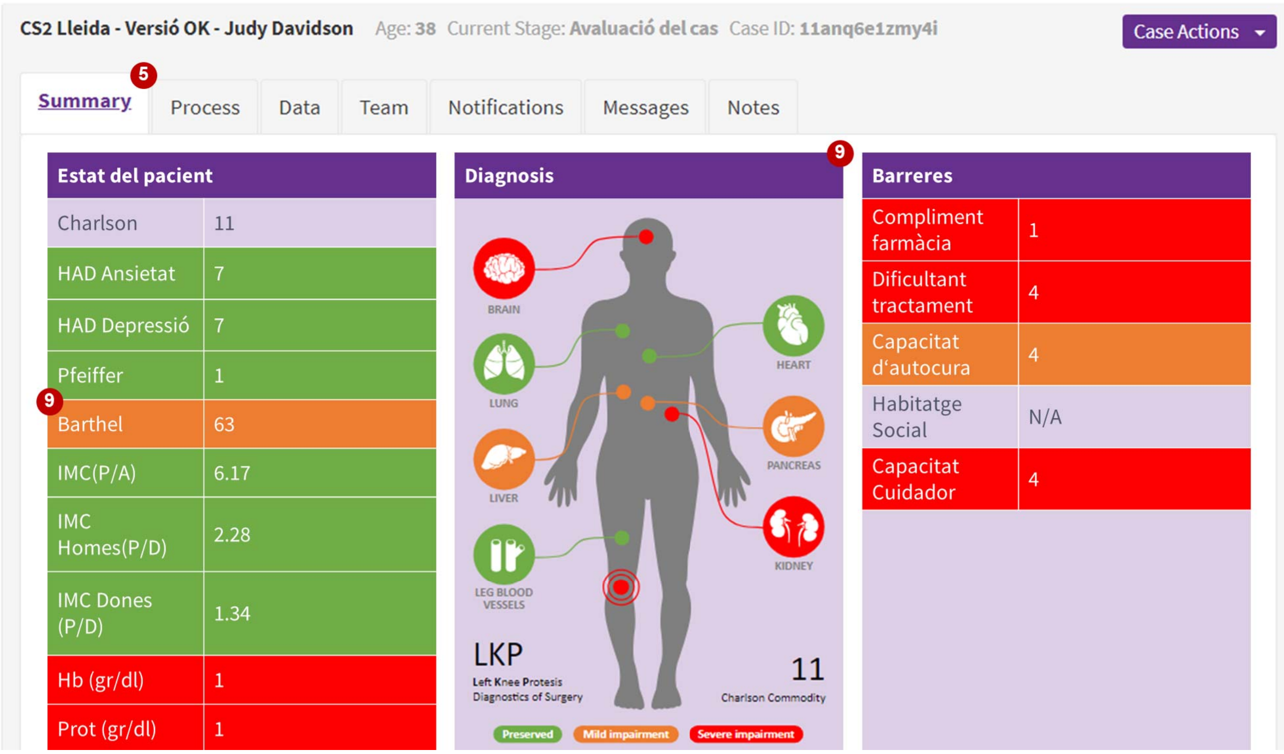

Fig. 5. The case summary aggregates the patient state (Lleida CS2), e.g., the Peifer links to the score of the Peifer questionnaire task and is colored according to the thresholds. The body highlights the diagnosis (LKP, RKP, LHA, RHA) and shows the organs colored (green, orange, red) according to the damage.

TABLE I
SUPPORTED MODEL ELEMENTS MAPPED TO THE RELATED REQUIREMENTS.

Model Element	**Full stack mode-based approach (R1)**				**Integration (R2)**	**Communication (R3)**	**Ref.**
	Data models	Adaptive process models	Role-based access right models	Simple user interface models			
Workspace	●	●	●	●	●	○	[8]
Group	○	●	●	○	●	○	[8]
User Definition	◐	◐	●	○	●	○	[8]
Entity Definition	●	◐	○	○	○	○	[8]
Attribute Definition	●	◐	○	◐	○	○	[9]
Derived Attribute Definition	●	◐	○	◐	○	○	[10] [11]
Case Definition	○	●	●	○	○	◐	[6]
Stage Definition	○	●	○	○	◐	○	[6]
HumanTask Definition	○	●	○	○	○	○	[6]
AutomatedTask Definition	○	●	○	○	●	○	[6]
DualTask Definition	○	●	○	○	●	○	N.A.
HttpHook Definition	○	●	○	○	●	○	[6]
Sentry Definition	○	●	○	○	○	○	[6]
SummarySection Definition	●	◐	○	◐	○	○	[6]
User Interface Reference Def.	●	○	○	●	○	○	[6]
Case Messages*	○	○	○	○	●	●	[6]
Case Alerts*	○	○	○	○	●	●	[6]
Case Notes*	○	○	○	○	○	●	N.A.

Legend: ● supports ◐ partly supports ○ no support *only on instance level e.g., the model element DualTaskDefinition supports adaptive process modeling and the integration with third-party systems.

Case Definition The Case Definition is a container for all case-specific model elements (StageDefinitions, HumanTaskDefinitions, AutomatedTaskdefinitions, DualTask-Defintions, SentryDefinitions, HTTPHookDefinitions and SummarySectionDefinitions) and contains all information that is needed to instantiate a case. In addition, role constraints for the case owner can be defined, e.g., only a clinician can be a case owner, not a nurse. Every instantiated case has an internal state machine similar to all process elements (see Figure 6).

Stage Definition Stage Definitions cluster related Task Definitions into one container that can be used as a precondition for a SentryDefinition. The implementation uses a composite pattern, the composite is the StageDefinition and the leaves are HumanTaskDefinitions, AutomatedTaskDefinitions, or DualTaskDefinitions, the abstract component is the process definition. Figure 6 illustrates possible states and state transitions for instantiated process definitions.

HumanTask Definition HumanTask Definitions support defining tasks that are completed by a human-centric input on the SACM UI. Every human task definition has meta-data such as a title, a dynamic tile that is adapted related to the content, a due date and an owner that can be initialized with other tasks, a flag to manually activate the task, an enumeration that expresses if the task can be repeated (`ONCE`, `SERIAL`, `PARALLEL`), and a footnote to reference copyrights of the original clinical questionnaires if needed. All task definitions have task parameter definitions that consist of a path that links to the related attributes in the generated case data tree. All required information how to generically represent the task parameter is taken from the related attribute definition. In addition, every task parameter has a flag that indicates whether the field is mandatory and a flag that indicates if a parameter is readable or writable, insufficient combinations are prevented. During the activation of a task, new entities attached according to their entity definition and the related task parameter paths are resolved to the actual attribute references.

AutomatedTask Definition AutomatedTask Definitions support defining tasks that are completed by third-party systems and need no human input on the SACM UI. The key difference to the HumanTask Definition is that no due dates are supported. The owner is supported because a human needs to be responsible in case an unforeseen interaction is required.

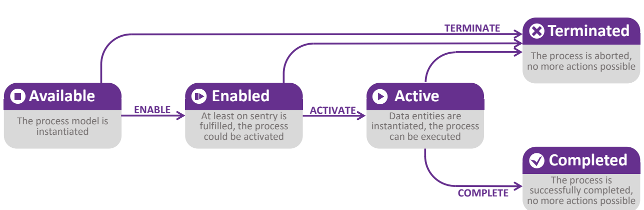

Fig. 6. State diagram, including all possible state transitions.

DualTask Definition DualTask Definitions support defining tasks that first expect a human-centric input and then automatically execute the second part. This concept is a combination of a HumanTask and an AutomatedTask. Theoretically, this could be modeled with a stage that contains a HumanTask and an AutomatedTask that has a sentry definition with a precondition to the HumanTask. This model element enormously simplifies the user interface representation and integration with third-party systems because only one process element needs to be tracked.

HTTPHook Definition Hook Definitions allow the integration of third-party systems. For a state change event (`AVAILABLE`, `ENABLE`, `ACTIVATE`, `COMPLETE`, `TERMINATE`) of the process elements HTTP requests can be defined. The allowed HTTP methods are `GET`, `PUT`, `POST`, and `DEL`. Optional an error alert is generated based on a predefined error message for none 200 HTTP status response.

Sentry Definition Sentry definitions allow to define entry conditions for each process element. The preconditions can be based on activities or MxL expressions on the case related data. One sentry supports multiple preconditions that are linked with a logical `and`. A process definition element can have multiple sentry definitions that are linked with a logical `or`.

SummarySection Definition A case can have multiple summary section definitions, whereas each summary section has a list of paths that refer to the attribute values. The paths are evaluated on every request to show the latest data produced by the workflow. Simple layout flags allow defining a three-column-layout.

User Interface Reference Definition The user interface uses a generic representation that shows input files, drop-down fields, and date selectors depending on the linked attribute definition. The default rendering behavior can be overridden with custom UI representations such as illustrated with the derived attribute that dynamically generates the SVG for the body representation on the summary page. Another example is the coloring values depending on thresholds, e.g., $uiReference = "SVG", uiReference = "colors(0 <= green <= 1 < orange <= 3 < red < 11)"$

Case Messages Every case writer can create new messages, supported communication channels are clinician-to-clinician SACM internal and clinician-to-patient SACM-SMS external. Each user can individually mark messages as read.

Case Alerts During the execution of tasks, several unforeseen events can occur that are represented by the following alert types: `ERROR` alerts, e.g., indicate that the system synchronization with an HTTP hook execution has failed, `CORRECT` alerts indicate that values of already completed tasks have been modified and may need to be checked and `CUSTOM` alerts are mostly domain specific and are generated by third-party systems.

Case Notes In addition to the semi-structured treatment plan according to the ACM design principles an unstructured wiki-based rich text documentation approach with concurrent edit prevention is provided.

VI. Evaluation

During the system design and implementation, improvements were iteratively applied based on regular feedback meetings with the clinical trial hospitals. Currently, case study 2 is being modeled and uploaded into production[2] for all four sites. Case study one will follow soon.

Figure 7 illustrates the Groningen case study 2 based on an extended CMMN [3] notation. The stages are colored in light purple, and the tasks in dark purple. Within the case identification stage, the patient inclusion takes place. The first task is to assign an existing patient-user to the case and assign the professional-users for each role such as case manager, clinician and nurse. Upon completion of this task, the assigned roles are then propagated to the related tasks and the access rights are granted to the case. All tasks pending on the completion of this task, are activated according to the sentry definitions. The activated tasks include completing the patient consent form, accomplishing the technological test and completing the ASA questionnaire. After these tasks are completed, the case identification stage is automatically completed because not task is pending any more.

Following the first iteration of the case evaluation stage is automatically activated and the evaluation date task becomes active. When the due date is set, all other questionnaire task are activated. In general, this stage helps evaluate the detailed parameters of the patient to decide what intervention actions should follow. Clinical questionnaires include the Charlson[4], GFI, ADL, iADL, NRS, MNASF, and the Hospital Anxiety and Depression Scale. Once the first iteration of the evaluation stage is completed, a second iteration can be manually activated by the clinician to reevaluate the results at any time to determine the success of treatment.

After the completion of the first case evaluation stage, the workplan stage is automatically activated where professionals can manually prescribe dual tasks to the patient, e.g., for a physical activity dual task, the clinician defines the start date, the end date, and the number of steps that the patient should walk every day. After the clinician completes the task definition, an HTTP hook containing the configured parameters is sent to the SMS system. The SMS system is responsible for notifying the patient to fulfill the specified task and synchronizes the results into the SACM system. The SACM visualizes the resulting time series as a line chart diagram. When the patient does not fulfill the specified number of daily steps, the SMS system appends a notification to the related task in the SACM system and the responsible task owner obtain a notification on their dashboard. The workplan tasks are all optional and can be repeated in parallel as often as individually required by the patient. Similarly, the monitoring prescription task allows to asking the patient to measure his blood pressure or other parameters on a regular basis. Simple tasks allow the clinician to prescribe very basic tasks such as move your hands or simply define any ad-hoc action based on a textual description to the patient. When the patient treatment is finally completed, the workplan stage is manually completed by the clinician and the discharge stage with the discharge form task is activated.

Since we setup the production environment in production mode, 52 users have registered and the RESTFul SACM Backend API has received 476,524 requests. Table II shows the API usage aggregated based on resources according to our logs until the 8th of May 2018. Only requests for the model execution are considered, all model definition requests are removed. Overall, the requests are distributed across regions as follows: 56% from Spain, 27% from Israel, 10% from Germany and 7% from the Netherlands. Approximately one-third of the requests could not be resolved to a specific region due to a change in the logging mechanism.

TABLE II
Resource usage of the SACM backend API.

RESTFul Resource	Method	#Requests
Workspaces	GET	2,862
Groups	POST	251
	DELETE	2
Users	GET	1,960
	DELETE	5
	PATCH	192
	POST	180
Cases	DELETE	245
	GET	52,277
	POST	760
Processes	GET	138,845
Stages	POST	49
HumanTasks	POST	26,512
	GET	9,969
AutomatedTasks	POST	411
	GET	100
DualTasks	POST	12,478
	GET	3,162
Alerts	POST	145
	GET	98,627
Messages	POST	1,437
	GET	117,055
SummarySections	GET	9,000
$\sum$	*	476,524

VII. Conclusion

We presented a holistic model-based ACM approach for the healthcare domain. Our approach was motivated by challenges that occurred during an H2020 integrated care project called CONNECARE. Based on case studies, we derived the high-level requirements and presented the conceptual user interface features. In addition, we explained the relevant modeling elements and mapped these with the derived requirements. As an evaluation, the workflow of one case study was described in detail including a brief API usage summary.

We believe that challenges such as 1) the high diversity between hospital sites and each case study, 2) the embedding information of existing information systems, and 3) the coordination and communication of many different stakeholders,

[2]https://system.connecare.eu/sacm/
[3]https://www.omg.org/spec/CMMN/1.1/
[4]http://www.bgs.org.uk/pdfs/assessment/cci.pdf (modeled in the SACM)

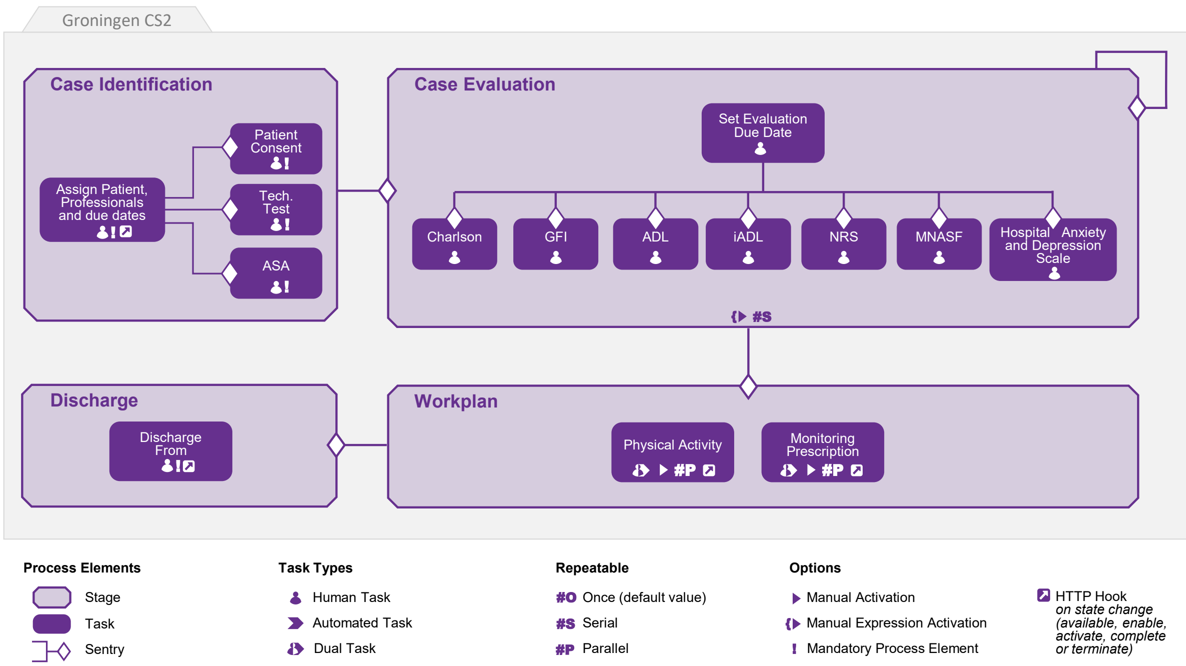

Fig. 7. CMMN diagram representing the Groningen CS2 (version: 11/05/2018).

can be covered by a holistic model-based ACM approach customized for the healthcare domain. The full stack model-based approach reduces dependencies and complexity compared to partially model-based approaches due to unnecessary model transformations. Our approach allows to orchestrate third-party systems and embed the resulting information into our data structure. Direct communication is provided via messaging. Several features such as task ownership, due dates, unstructured case notes, and a dashboard support the coordination of the involved stakeholders.

However, the main limitation of our approach is that customizations beyond the supported model elements would create additional complexity. The alternative approach is to use commercial ACM engines such as Camunda and build completely customized Single Page Applications for each treatment plan that are not easily reusable.

Acknowledgment

We thank all members of the CONNECARE consortium for providing continuous feedback to improve the system design.

References

[1] E. Vargiu, J.M. Fernndez, F. Miralles, I. Cano, E. Gimeno-Santos, C. Hernandez, G. Torres, J. de Batlle J. Colomina, R. Kaye, B. Azaria, S. Nakar, M.M.H. Lahr, E. Metting, M. Jager, H. Meetsma, S. Mariani, M. Mamei, F. Zambonelli, F. Michel, F. Matthes, J. Goulden, J. Eaglesham, and C. Lowe. Integrated care for complex chronic patients. In *ICIC17 International Journal of Integrated Care 17(5), Dublin*. International Foundation for Integrated Care, 2017.

[2] Rachelle Kaye, Khaled Abu-Hossien, Felip Miralles, Eloisa Vargiu, and Bella Azaria. From connected care to integrated care a work in progress. In *ICIC17 International Journal of Integrated Care 17(5), Dublin*. International Foundation for Integrated Care, 2017.

[3] P. Osvath, V. Voros, A. Kovacs, I. Greges, S. Fekete, T. Tenyi, and S. Fekete. The ict4life projectdesign and development of a new information technology platform for patients with alzheimer's disease. *European Psychiatry*, 41:S545 – S546, 2017. Abstract of the 25th European Congress of Psychiatry.

[4] Carlos A. Velasco, Yehya Mohamad, and Philip Ackermann. Architecture of a web of things ehealth framework for the support of users with chronic diseases. In *Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*, DSAI 2016, pages 47–53, New York, NY, USA, 2016. ACM.

[5] Julie Doyle, Evert-Jan Hoogerwerf, Janneke Kuiper, Emma Murphy, Caoimhe Hannigan, , John Dinsmore, Thomas van der Auwermeulen, Jonas Albert, An Jacobs, Lorenza Maluccelli, Lorenzo Desideri, and Valentina Fiordelmondo. Designing a proactive, person-centred, digital integrated care system. In *Proceedings of the International Journal of Integrated Care, 17(5)*, 2017.

[6] Adrian Hernandez-Mendez, Felix Michel, and Florian Matthes. A practice-proven reference architecture for model-based collaborative information systems. *Enterprise Modelling and Information Systems Architectures*, 13:262–273, 2018.

[7] Keith Swenson. *Taming the unpredictable: real world adaptive case management: case studies and practical guidance*. Future Strategies Inc., 2011.

[8] Thomas Bchner. *Introspektive modellgetriebene Softwareentwicklung*. Dissertation, Technische Universitt Mnchen, Mnchen, 2007.

[9] Florian Matthes, Christian Neubert, and Alexander Steinhoff. Hybrid wikis: Empowering users to collaboratively structure information. *ICSOFT (1)*, 11:250–259, 2011.

[10] Thomas Reschenhofer and Florian Matthes. Supporting end-users in defining complex queries on evolving and domain-specific data models. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2016, Cambridge, United Kingdom, September 4-8, 2016*, pages 96–100, 2016.

[11] Thomas Reschenhofer. *Empowering End-users to Collaboratively Analyze Evolving Complex Linked Data*. PhD thesis, Technische Universität München, 2017.