



# The Case Manager: An Agent Controlling the Activation of Knowledge Sources in a FHIR-Based Distributed Reasoning Environment

Giordano Lanzola<sup>1</sup> Francesca Polce<sup>1</sup> Enea Parimbelli<sup>1</sup> Matteo Gabetta<sup>2</sup> Ronald Cornet<sup>3</sup>  
Rowdy de Groot<sup>3</sup> Alexandra Kogan<sup>4</sup> David Glasspool<sup>5</sup> Szymon Wilk<sup>6</sup> Silvana Quaglini<sup>1</sup>

<sup>1</sup> Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Pavia, Italy

<sup>2</sup> Research and Development Division, Biomeris S.r.l, Pavia, Italy

<sup>3</sup> Medical Informatics, Amsterdam Public Health Institute, Methodology & Digital Health, Amsterdam University Medical Centers, Amsterdam, The Netherlands

<sup>4</sup> Department of Information Systems, University of Haifa, Haifa, Israel

**Address for correspondence** Giordano Lanzola, Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Via Ferrata 5, 27100 Pavia, Italy (e-mail: giordano.lanzola@unipv.it).

<sup>5</sup> Deontics Ltd., London, United Kingdom

<sup>6</sup> Research and Development Division, Institute of Computing Science, Poznan University of Technology, Poznan, Poland

Appl Clin Inform 2023;14:725–734.

## Abstract

**Background** Within the CAPABLE project the authors developed a multi-agent system that relies on a distributed architecture. The system provides cancer patients with coaching advice and supports their clinicians with suitable decisions based on clinical guidelines.

**Objectives** As in many multi-agent systems we needed to coordinate the activities of all agents involved. Moreover, since the agents share a common blackboard where all patients' data are stored, we also needed to implement a mechanism for the prompt notification of each agent upon addition of new information potentially triggering its activation.

**Methods** The communication needs have been investigated and modeled using the HL7-FHIR (Health Level 7-Fast Healthcare Interoperability Resources) standard to ensure proper semantic interoperability among agents. Then a syntax rooted in the FHIR search framework has been defined for representing the conditions to be monitored on the system blackboard for activating each agent.

**Results** The Case Manager (CM) has been implemented as a dedicated component playing the role of an orchestrator directing the behavior of all agents involved. Agents dynamically inform the CM about the conditions to be monitored on the blackboard, using the syntax we developed. The CM then notifies each agent whenever any condition of interest occurs. The functionalities of the CM and other actors have been validated using simulated scenarios mimicking the ones that will be faced during pilot studies and in production.

**Conclusion** The CM proved to be a key facilitator for properly achieving the required behavior of our multi-agent system. The proposed architecture may also be leveraged in many clinical contexts for integrating separate legacy services, turning them into a consistent telemedicine framework and enabling application reusability.

## Keywords

- HL7 FHIR
- multi-agent systems
- clinical decision support
- telemedicine

received

November 15, 2022

accepted after revision

May 12, 2023

accepted manuscript online

June 20, 2023

DOI <https://doi.org/10.1055/a-2113-4443>.

ISSN 1869-0327.

© 2023. The Author(s).

This is an open access article published by Thieme under the terms of the Creative Commons Attribution-NonDerivative-NonCommercial-License, permitting copying and reproduction so long as the original work is given appropriate credit. Contents may not be used for commercial purposes, or adapted, remixed, transformed or built upon. (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Georg Thieme Verlag KG, Rüdigerstraße 14, 70469 Stuttgart, Germany

## Background and Significance

Innovations in information and communication technologies provide a promising way to bridge the gap between a patient's home and the clinic through telemedicine services.<sup>1,2</sup> This new digital health care model (often known as *e-health*) aims to promote patient self-sufficiency, self-management, and empowerment.<sup>3,4</sup> Smartphones, tablets, and smartwatches support the collection and transmission of clinical data—patient-reported outcomes (PROs) and patient-generated data (PGD)—which can be integrated with clinical data in the electronic health record (EHR) for a comprehensive overview of the patient state. Data mining and artificial intelligence (AI) techniques help in automating the analysis of the huge amount of information collected.<sup>5</sup>

This work was conducted in the context of the European Union (EU)-funded CAPABLE project. CAPABLE focuses on the exploitation of big data (in particular PROs and PGD) using AI techniques to effectively monitor cancer patients. The authors developed a multi-agent system (MAS) that acquires data from patients and provides them with coaching and their doctors with clinical decision support. This article focuses on the distributed software agent architecture of CAPABLE and its core component—the Case Manager (CM)—that acts as an orchestrator for the agent activities. CAPABLE will be deployed at two clinical centers, Istituti Clinici Scientifici Maugeri (ICSM) in Pavia, Italy, and National Cancer Institute (NKI) in Amsterdam, The Netherlands, where the concept will be tested in a pilot study involving different cancer types.

## Opportunities and Challenges of Multiagent Systems

MASs consist of a network of loosely coupled, intelligent, autonomous entities that dynamically interact to solve problems falling beyond the expertise of a single entity.<sup>6</sup> There is a growing interest in using MASs in a medical context where problems are inherently complex and interdisciplinary, requiring heterogeneous knowledge sources to process data collected at different sites.<sup>7–10</sup> Agents involved in a MAS often communicate through a shared data repository called a *blackboard* (the so-called Blackboard Architecture).<sup>11</sup> To save agents from continuously querying the blackboard, a control agent is usually introduced. When an agent can potentially contribute to the problem-solving process, it is activated by the control agent in response to specific triggering events generated by changes on the blackboard. Thus, each agent must first inform the control agent about the type of events it is interested in. Such an event-based Blackboard Architecture is adopted in several contexts exploiting MASs and is widely referenced in the literature.<sup>12–15</sup>

MASs rely on interoperability at technical, semantic, and process levels. Communication standards allow technical interoperability, but semantic interoperability is also needed since agents must know how to properly interpret any exchanged data in order to use them appropriately. A well-known syntactic and semantic standard for improving interoperability between different health care applications is HL7 (Health

Level 7) FHIR (Fast Healthcare Interoperability Resources).<sup>16,17</sup> FHIR has gained strong acceptance by health care information system developers and is now widely adopted for representing and sharing clinical information in systems providing various functionalities including decision support.<sup>18–21</sup>

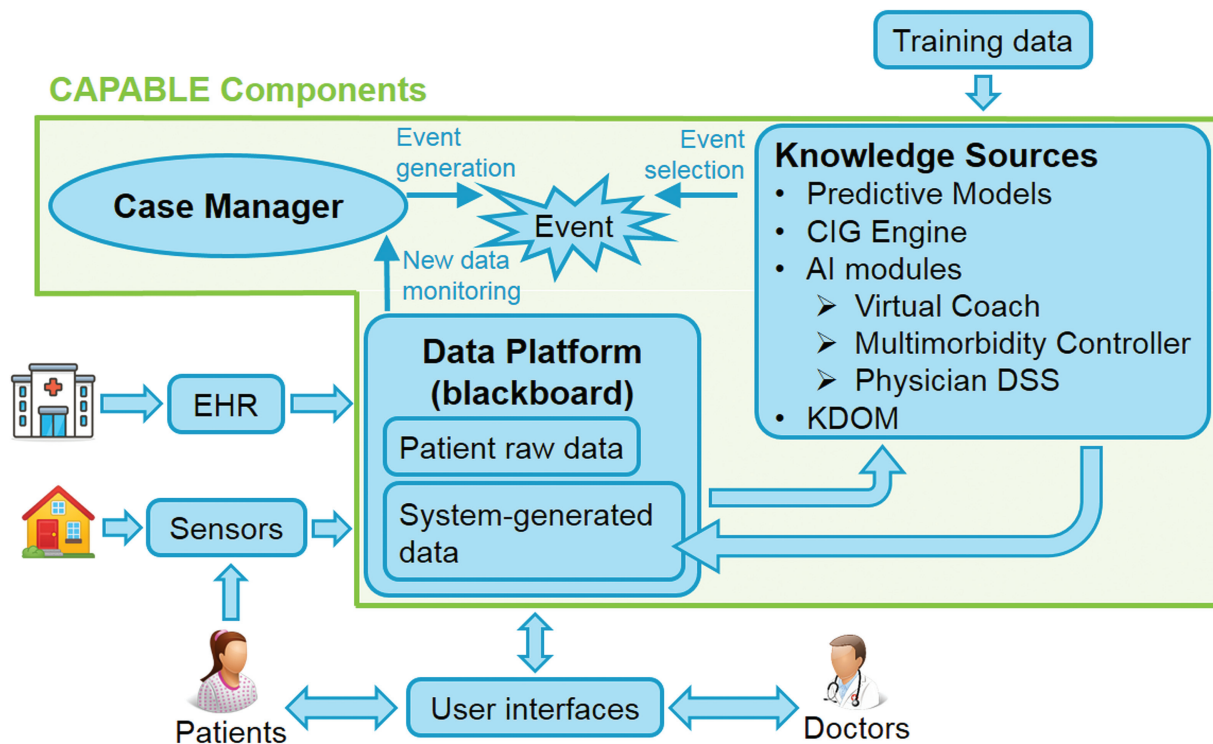
Agents rely on various Data Sources including the EHR and other hospital information systems, various types of sensors, and other applications targeting patients and doctors. Data items are mapped to unique concepts using standardized terminologies such as SNOMED CT,<sup>22</sup> RxNorm,<sup>23</sup> LOINC,<sup>24,25</sup> or the ICD Family.<sup>26</sup>

## The CAPABLE System as a MAS

The high-level architecture adopted by CAPABLE is shown in **Fig. 1**. PROs are entered by patients through forms or questionnaires on smartphones while PGD are acquired using wearable or environmental sensors. Data collected at regular visits are available from the EHR. All these data are saved on a shared Data Platform (DP), representing the system blackboard, which makes them available to patients and doctors for perusal and to other agents for automated processing. Because of the multidisciplinary nature of the knowledge involved, we have leveraged standards-based and service model approaches of other architectures to implement a distributed decision support system (DSS) based on Computer Interpretable Guidelines (CIGs) applied by agents.<sup>27,28</sup> This provides flexibility in representing knowledge and makes agents responsible for recognizing when they can use their specific problem-solving skills and contribute new data and information. The main agents that encapsulate the required knowledge sources for processing data are:

- *CIG Engine*: manages CIGs by instantiating and executing them for specific patients in cooperation with the other agents.
- *Virtual Coach* (VC): provides patients with coaching and motivational support controlled by CIGs and workflows, expert-based rules, and models based on behavior change theories, in particular the Fogg behavior model.<sup>29</sup> These support modalities follow the recommended intervention strategies proposed in the ICT4PEM framework.<sup>30</sup>
- *Multimorbidity Controller* (GoCom): checks for and mitigates any conflict caused by recommendations coming from multiple CIGs. GoCom is goal-oriented and uses hierarchical goal-tree structures to represent the patients' problems and treatments. Drug–drug interactions and guideline recommendation interactions are detected and mitigated automatically. Alternative options for non-conflicting treatment combinations are stored in the DP and afterwards presented to the physician by the graphical user interface.
- *Physician DSS* (PDSS): supports clinicians with hints and action suggestions from active CIGs.
- *Knowledge-Data Ontology Mapper*: computes clinical abstractions on raw data, PROs, and PGD concerning the patient.

In PDSS and VC, most knowledge is captured through CIGs expressed in the PROforma language, which models clinical



**Fig. 1** The big picture illustrating the CAPABLE architecture and its primary components acting as independent agents that encapsulate the Knowledge Sources required for processing patient data.

processes as formal networks of tasks with logical conditions governing their execution.<sup>31</sup> Additionally, VC relies on a few algorithms implemented as internal modules. These algorithms are more computationally demanding and are more effectively implemented outside of PROforma. GoCom relies on domain knowledge provided by external sources (i.e., repositories with drug characteristics including potential interactions) and on a specialized planning algorithm which constructs goal trees from PROforma CIGs and is implemented directly in GoCom.

In CAPABLE agents are triggered by *Events* that are associated with conditions occurring on the blackboard. This is shown in **Fig. 1** through narrow arrows representing the control flow as opposed to wide arrows representing data and information flow. Events allow agents to set up a cooperative process dynamically shaped on an opportunistic basis following the incremental addition of information to the DP. Consequently, there is a strong need to orchestrate the efforts of all the agents. This translates into triggering an agent whenever a data pattern recognized and handled by that agent appears in the DP. This is accomplished by the CM which is described in the next sections.

## Materials and Methods

### Exploiting the HL7 FHIR Standard in CAPABLE

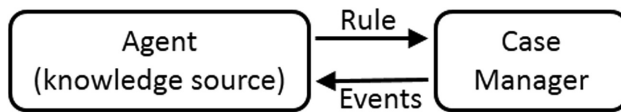
All data in CAPABLE, either generated by the system or extracted from the EHR, are internally stored in the DP using OMOP (Observational Medical Outcomes Partnership)

Common Data Model (CDM). OMOP-CDM was chosen to store patient data, instead of direct FHIR persistence, to simplify the longitudinal analysis at the end of the pilot studies. This is possible with tools developed within the OHDSI (Observational Health Data Sciences and Informatics) initiative that leverage the ontology and connected vocabularies on which OMOP-CDM is based.

However, for exchanging data within CAPABLE, instead of using the traditional clinical document-centric approach, we preferred the compositive approach adopted by HL7 FHIR R4 where health care information entities are represented as independent modular elements called *resources*.<sup>32</sup> Using the FHIR REST Application Programming Interface (API), the only way to access the data, all patient information is exchanged externally between data sources and the DP and between the DP and the agents in the form of FHIR resources. FHIR resources exchanged with the DP are transformed on-the-fly and mapped into OMOP elements leveraging the *OMOPonFHIR* open-source software.<sup>33</sup>

### Designing the FHIR-Based Case Manager

FHIR natively allows a push-based mechanism for notifying systems through its Subscriptions, but we decided not to use it in CAPABLE for two reasons. First, Subscriptions have a maturity level score of 3, which means that they are halfway between draft and normative, and we were unsure about their implementation state in *OMOPonFHIR*. Second, and foremost, that mechanism considers each resource individually while we wished to aggregate resources into clinically meaningful patterns.



**Fig. 2** Interaction of an Agent with the Case Manager: Agents subscribe with the Case Manager through *Rules* and are notified by it through *Events*.

The CM was designed for programmable behavior, to avoid hard-coding any relationship among data required to trigger the various agents. Doing so would have linked the CM closely to all other agents, mixing up the responsibilities of the developers and deeply affecting the possibility of easily adding new agents.

The key to the programmable CM architecture is the *Rule*, which is a directive representing an event and encapsulating the conditions upon which an agent is triggered by the CM. As shown in ▶Fig. 2, rules are dynamically loaded/unloaded to/from the CM by the agents whenever they need to introduce or alter the data patterns causing their activation. Following the loading of a new rule, the CM starts monitoring the DP and generates events that are signaled to the interested agent whenever the rule conditions are met.

Devising a syntax for the rule language was simplified by the adoption of FHIR. Rules are written in JSON and filter conditions for data on the DP leverage the syntax adopted by FHIR searches (▶Fig. 3 shows the rule productions expressed in the enhanced Backus–Naur form). Besides clauses with filter conditions, agents may also add JavaScript code to check the FHIR resources retrieved by the rule as a group, which represents the main difference from FHIR Subscriptions. If the group passes the check, the event is generated and signaled to the agent.

▶Fig. 4 shows a rule triggering events for patients whose recent fever Observations constitute a monotonically

increasing series including at least three values in the past 72 hours. This is an artificial example that does not represent any specific piece of knowledge in CAPABLE and only serves the purpose of contrasting the FHIR subscription approach that always considers each resource individually. Lines 5 to 8 search for an active Patient, while lines 9 to 14 search for his/her fever Observations in the past 72 hours. Both queries rely on the FHIR syntax for searches. The operator *rtge* (line 13) implemented by the CM is an extension of the FHIR *ge* operator. It computes the time reference offsetting the current time by a period expressed with the ISO 8601 standard. Since the time period is negative (i.e., –72 hours), it searches for Observations with an *effectiveDateTime* greater than or equal to 72 hours in the past. The monotonic fever increase is expressed by adding a JavaScript function *checkFeverIncreasing()* to the rule, which scans all the fever readings extracted and increments a variable (*bucket*) if they are increasing. Finally, the last JavaScript predicate returns true if a Patient was actually found (*Var1.length == 1*) having at least three fever readings (*Var2.length ≥ 3*) and with *checkFeverIncreasing()* counting at least two intervals (i.e., among three values). If the predicate returns true, the event is triggered and delivered to the interested agent.

## Results

The CM has been developed as a Web service on top of the Jersey Framework API and the Grizzly HTTP Engine. It is a separate module, written in Java, following the *Observer* pattern along with the DP which plays the role of the *Observable* notifying the CM about every change in data. This process is depicted in ▶Fig. 5 illustrating the acquisition of a new piece of data (*New Data*) concerning a patient-reported symptom. The CM is informed about the new

```

Rule:=
{
  eventName      : "string",           // Unique name for the event rule
  eventResources : [ EventResource+ ],
  eventExpression: "string"           // JavaScript Expr exploiting bindings returning a Boolean
}
=====
EventResource:=
{
  resourceType: "string",             // FHIR Type
  (searchParams: [ SearchParam+ ] )?,
  resourceVarName: "string",          // Variable name for JavaScript Expr bindings
}
=====
SearchParam:=
{
  paramName: "string",                // FHIR Search parameter
  paramValue: "string"                // FHIR Search expression
}
  
```

**Fig. 3** The syntax adopted by *Rules* used by the Case Manager expressed in enhanced Backus–Naur form. Nonterminal symbols are in bold with *Rule* being the top-level goal. Keyword terminals are in italic. Repetition symbols have the usual meaning such as: “+” (one or more) and “?” (at most one).

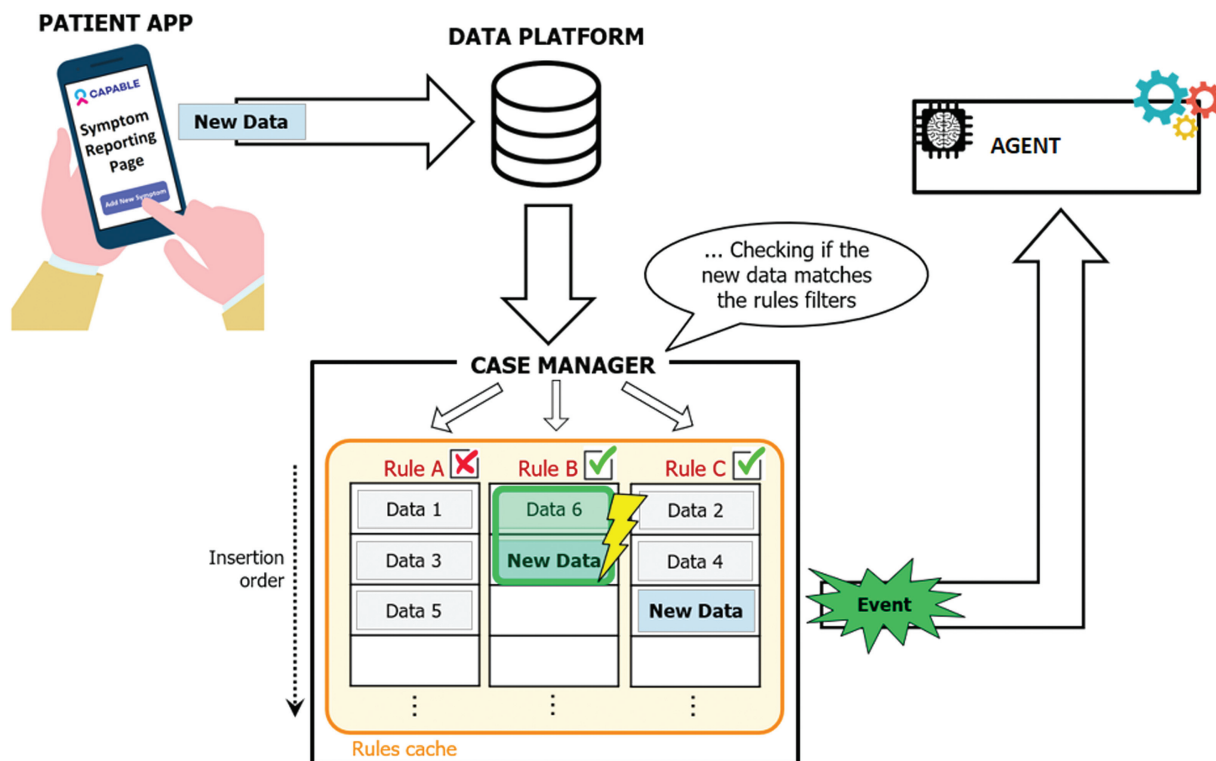


```

01 # Look for active Patients with fever (i.e. > 37.5) monotonically increasing
02 # over at least 3 readings (2 intervals) in the past 3 days (72 h)
03
04 { eventName: "Patients_with_increasing_fever",
05   eventResources: [ { resourceType: "Patient",
06                     searchParams: [ { paramName: "active",
07                                   paramValue: "true" } ],
08                     resourceVarName: "Var1" },
09                     { resourceType: "Observation",
10                     searchParams: [ { paramName: "code",
11                                   paramValue: "http://snomed.info/sct|386661006" },
12                                   { paramName: "effectiveDateTime",
13                                   paramValue: "rtge-PT72H" } ],
14                     resourceVarName: "Var2" } ],
15   eventExpression: "function checkFeverIncreasing() {
16                     var bucket=0; var i;
17                     for (i=0; i < (Var2.length - 1); i++) {
18                       if ( Var2[i+1].valueQuantity.value >= 37.5 &&
19                         Var2[i].valueQuantity.value < Var2[i+1].valueQuantity.value)
20                         bucket++;
21                       else
22                         bucket = 0;
23                     }
24                     return bucket;
25                   }
26   return Var1.length == 1 &&
27         Var2.length >= 3 &&
28         checkFeverIncreasing() >= 2;"
29 }

```

**Fig. 4** Example of a Case Manager rule. The rule triggers events for patients whose fever Observations end with a monotonically increasing series including at least three values collected during the past 72 hours.



**Fig. 5** The Case Manager acting as an Event scheduler. New information added to the Data Platform is signaled to the Case Manager that matches it against all known rules and adds it to the evaluation sets of the relevant ones (i.e., B, C). Following any change in the evaluation set, the JavaScript code is eventually invoked and possibly an Event is generated (i.e., B).

resource by the DP and compares it to the conditions of every existing rule. If the resource matches a rule's conditions, it is added to the evaluation set of that rule and the JavaScript code is invoked on that set for possibly generating the corresponding event. This approach enables the CM to behave as a highly responsive scheduler notifying agents about their events of interest. Moreover, expressing the conditions for generating events through declarative rules is an effective way for decoupling the implementation of the CM from that of the agents. In the **Supplementary Appendices** (available in the online version), we provide two sample use cases for better understanding how the process orchestrated by the CM develops.

Further flexibility is achieved through additional features offered by the rule syntax. These allow (1) periodically flushing the evaluation sets to discard obsolete resources; (2) periodically re-evaluating rules to generate additional (time-dependent) events even in the absence of new resources; and (3) evaluating a rule for a single patient or multiple patients (even though for diagnostic or therapeutic purposes data must be analyzed on a per-patient basis, for higher level decisions data collected across all patients may need to be checked). Finally, rules may be marked as *Message Queues* modeling the exchange of messages between agents, so that the CM can also play the role of a plain broker. In this case, notifications will be triggered by every single resource (referenced by the rule) when it is added to the DP and the notification contents will include by design just the single resource representing the message exchanged. This brokering role of the CM removes the need for direct interaction among the agents and helps further enforce their decoupling as separate entities.

### Validation

The CM together with other agents has been validated through approximately 20 clinical scenarios addressing the treatment of cancer side effects. We used real patients' data to create the scenarios with each one encompassing several segments representing different time periods. For each segment, we prepared FHIR resources and we developed a simulator to load them automatically preserving their relative time distances.<sup>34</sup> Based on the information loaded in the DP, the CM generates the right events for the specific agents that manage the clinical case, according to the embedded knowledge (e.g., PROforma CIGs). The simulator also allows checkpoints to be set up to automatically verify the generation of the relevant events by the CM as well as the expected actions and recommendations by the agents. All simulations confirmed the correct behavior of the CM. The CAPABLE system is now being installed at the two hospitals, ICSM and NKI, where the pilot studies will be accomplished, and enrollment is expected to start shortly.

### Discussion

When the CAPABLE consortium was assembled, each partner brought their methodological and development expertise to devise an infrastructure composed of multiple independent agents supporting both patients and doctors. It immediately

became clear that enabling communication and coordination among agents required:

- A common data representation shared by all agents.
- A mechanism for operationalizing agent communication and coordinating their activities.<sup>35</sup>

The first requirement was satisfied by introducing FHIR across the whole system. To address the second one, we did not want to set up separate communication channels for all pairs of interacting agents since this approach had major limitations and lacked scalability. Instead, our solution relied on the DP for data sharing and the CM for encapsulating and operationalizing the knowledge required to coordinate the agents. This approach is more comprehensive and unifying, much easier to manage, and consistent with the efforts of layering knowledge into separate levels, adopting different representation formalisms, which is a recurring issue in complex clinical AI applications.<sup>36</sup>

The CM was conceived based on our previous experiences with MASs.<sup>37</sup> Since we adopted FHIR, we designed the CM to support interoperability among loosely coupled agents leveraging it as a common semantic framework. This helped in enforcing a standard representation of medical information and ensuring proper interpretation thereof.<sup>38</sup> The most distinguishing feature of the CM is that, instead of encapsulating the knowledge that controls the overall behavior of the system,<sup>39</sup> it is completely customizable through a configuration language leveraged by the very same agents that it is expected to drive. This reduces coupling among agents allowing the integration of new ones.

In this context the CM shares some similarity with Arden Syntax (AS) rules.<sup>40</sup> Specifically, both allow defining events that trigger specific actions. AS rules also specify action implementations in their *action slots*. However, they are limited to performing basic computations, displaying messages, or storing data in the patient record. Another well-known limitation of AS is the lack of an underlying data representation, which requires defining mappings to a specific data repository. Adopting FHIR, the CM overcomes this limitation making the coordination knowledge independent of an underlying data repository.

Another language for formally representing clinical rules is GDL (Guideline Definition Language) leveraging the openEHR Reference Model and Archetype Model.<sup>41,42</sup> However, actions in GDL rules are limited to setting values of variables and data times, so it is less powerful than AS. While there are certain similarities between GDL and CM rules they have a different purpose. GDL rules specify both activation conditions and actions and are meant to be executed by a single agent. CM rules instead only specify the activation conditions to trigger a specific agent and the corresponding actions are chosen by the triggered agent. In other words, CM rules separate coordination knowledge from domain knowledge offsetting the responsibility to execute triggered actions onto specific agents, thus becoming orchestration rules for the distributed environment. Since CAPABLE uses FHIR, the PROforma formalism for representing CIGs seemed to be a better choice than GDL.

The CM is seen as an enabling technology introducing *Dependency Injection* and making available the *Inversion of Control* principle in a standard and declarative way as opposed to approaches based on proprietary solutions.<sup>43,44</sup> Leveraging the semantic layer provided by FHIR, the CM supports a consistent expressive language that the various agents use to define their events of interest. Those definitions are dynamically sent by the agents to the CM that interprets them and starts monitoring the DP. Thus, the CM is eventually able to drive agent activation without any prior knowledge about their functionality. Finally, upon generating events that result in the activation of a target agent, the CM also hands over the evidence causing those activations, i.e., involved FHIR resources, for further inspection.

We believe that the proposed solution may be useful in other scenarios as well, as it supports integration of legacy clinical systems through agent-like wrappers relying on the services provided by DP and CM. One of the fields that may benefit from our proposal is telemedicine. Despite positive outcomes of pilot studies, telemedical solutions often fail to enter routine practice. Among the reasons, besides the possibly disrupting impact on the organizational workflow,<sup>45</sup> there is an inherent difficulty in bringing telemedicine into practice due to the multitude of different proprietary and possibly incompatible standards that are always used in relevant research projects.<sup>46</sup> Current technology in clinical applications still focuses on data silos offering very limited interoperability among each other.<sup>47,48</sup> Solving this issue could result instead in a wealth of large-scale services built on top of already existing modules for which there is already a growing interest.<sup>49</sup>

The proposed architecture may be proficiently exploited for the distributed management of comorbidities that are increasingly found as the populations grow older.<sup>50,51</sup> In particular, multi-morbidity conditions often generate contradictory requirements that are difficult to address.<sup>52</sup> C3-cloud is another interesting project addressing comorbidity, even though in that case interaction mitigation is done manually by physicians that agree on a particular treatment.<sup>53</sup> Within CAPABLE, GoCom mitigates interactions automatically considering a large number of possible interactions without modeling them ahead of time.

Despite its potential, our work has limitations since the architecture has been implemented on a very small scale. Assembling telemedicine services from existing components and rendering them as autonomous independent agents requires additional middleware, particularly for supporting shared care. Brokers are needed to help the CM in accessing patient data shared across several institutions using standardized APIs as well as for allowing specialized components to access it.<sup>54</sup> As a design choice, the CM did not leverage FHIR Subscriptions which trigger events checking resources individually. Doing so would have enhanced the CM's integration with FHIR. However, this seemed too much of a stretch in the design since CM triggers not only events upon the acquisition of new resources, but also upon missing data or by periodically re-evaluating rules. Nevertheless, the need to match the FHIR standard imposes a limitation on the syntax for specifying the

conditions in rules. Within the CM we enhanced that syntax using JavaScript code, which is a procedural language, while a declarative approach would probably be a better choice. Finally, data availability is a challenge for CIG-based systems whose authors often shift the responsibility of checking their presence onto users.<sup>50</sup> In CAPABLE this is mitigated by the fact that the dynamics of patients' evolution is much slower than the system response. Moreover, CM rules are always used in "isolation" and do not constitute reasoning chains. When a rule is activated, it generates an event triggering an agent. Thus, we offset the responsibility from the CM onto the specific agent involved, while handling it at the infrastructural level could have resulted in a more general solution.

As a final remark, telemedicine systems may harm the patient. Thus, they must comply with the regulations of the country where they are used. In the EU, this is accomplished through the Medical Device Regulation 201/745 (MDR) of the EU Parliament.<sup>55</sup> According to the MDR (Annex VIII, Chapter III, Point 6, Rule 11), CAPABLE falls into risk class IIa. This had an impact on the activities of the project, especially regarding the clinical study taking place during its final year. Our goal is not to commercialize the system, which would require obtaining a CE marking, but just to assess its effectiveness in the pilot study. According to MDR Art. 62, a clinical investigation demonstrating the benefits of a device (without the CE marking) requires, besides other activities, preparation of a risk assessment according to ISO 14971, receiving a nonnegative opinion by the ethics committee of the institution responsible for the investigation, and authorization by the government of the member states involved. Thus, after receiving the approval of the local ethics committees of ICSM and NKI, we submitted notifications to the notified bodies of the government of the two countries involved.

## Clinical Relevance Statement

In CAPABLE we inherently needed to share structured information among several agents to implement the Case Manager as an orchestrator. Thus, sharing data is essential and was facilitated adopting FHIR, enhancing it with suitable terminology mappings. We accomplished this task using well-established standards such as SNOMED CT, LOINC, and OMOP-CDM. Besides simplifying data exchange, we believe that this approach enforces a higher quality of the collected data and helps in producing more reliable coaching and decision recommendations. Finally, using OMOP-CDM allows clinical researchers to share and reuse large libraries of statistical tools already developed by the scientific community.

## Multiple Choice Questions

1. Which architecture was chosen by the authors for the implementation of an advanced clinical decision support system?
  - a. Standalone monolithic architecture
  - b. Decision support embedded into the electronic health record

- c. Multi-agent system architecture
- d. Service-oriented architecture

**Correct Answer:** The correct answer is option c. A multi-agent system allowed partitioning the competencies among the various agents. For each agent the most appropriate representation formalisms and development tools were adopted, reusing many software libraries already developed by the researchers participating in the project. Coordinating the agents was also simpler through the implementation of the Case Manager.

2. If a complex system needs to be assembled reusing  $n$  existing components, what approach would minimize the development efforts required to enable information exchange among those?
  - a. Devising a specific format for data representation and rewriting the internals of every component so that they all eventually conform to that single format.
  - b. Coupling each of the components with as many services as needed to independently perform the conversions to/from any other one, possibly resulting in up to  $(n*(n-1))/2$  separate services.
  - c. Choosing components that only use the same information model (concept and terminologies).
  - d. Defining a common information model for the whole system, then mapping the requirements of each component to that common model using mapping services, resulting in up to  $n$  services.

**Correct Answer:** The correct answer is option d. Each component of our system uses its own information model that possibly differs from that adopted by others. This allows greater flexibility in selecting the most appropriate software tools for processing information. Mapping services may be adopted in case a conversion among different standards is required.

3. Which of the following approaches to coordination has been used by the authors as being the most flexible (in terms of decoupling the agents) and effective (in terms of the impact on a shared data repository)?
  - a. Hard coding the coordination strategies within each agent, so that they are also implicitly aware of the strategies of all the remaining ones
  - b. Designing a separate component to which agents dynamically declare their activation conditions and by which they are activated whenever those conditions are met
  - c. Designing a separate component that hard codes all the coordination strategies and drives the activation of the agents as needed
  - d. Adopting a shared data repository that is queried by the agents searching themselves for their activation conditions

**Correct Answer:** The correct answer is option b. Even though the development of a fully customizable

component is not trivial, it is the approach that maximizes decoupling of the agents. In our system, the Case Manager follows the *Inversion of Control* paradigm and can be configured directly by the agents that it is supposed to notify.

### Protection of Human and Animal Subjects

The system described in this paper will be used in two pilot studies performed in compliance with the World Medical Association Declaration of Helsinki on Ethical Principles for Medical Research Involving Human Subjects. The study was reviewed by the Institutional Review Boards of the two hospitals involved. For ICSM the study was approved on May, 11<sup>th</sup> 2022 with protocol number 2640CE. For NKI it was approved on Dec, 28<sup>th</sup> 2022 with protocol number NL81970.000.22 22-981/H-O.

According to the Regulations on Medical Devices (MDR) CAPABLE falls in Risk Class IIa as a software intended to provide information which is used to take decisions with diagnosis or therapeutic purposes. Thus, it requires regular assessment by a notified body. For this reason, two notifications to the Ministry of Health in Italy and to the Central Committee on Research Involving Human Subjects in The Netherlands have been filed.

### Funding

The work described in this article received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation over the years 2020–2024 under grant agreement no. 875052 to the CAPABLE project (<https://capable-project.eu>).

### Conflict of Interest

D.G. is an employee of and shareholder in Deontics Ltd., which is the company producing the PROforma CIG authoring tool and the execution engine. The remaining authors have no conflict of interest of any kind to declare.

### References

- 1 Hanlon P, Daines L, Campbell C, McKinstry B, Weller D, Pinnock H. Telehealth interventions to support self-management of long-term conditions: a systematic metareview of diabetes, heart failure, asthma, chronic obstructive pulmonary disease, and cancer. *J Med Internet Res* 2017;19(05):e172
- 2 Lanzola G, Losiouk E, Del Favero S, et al. Remote blood glucose monitoring in mHealth scenarios: a review. *Sensors (Basel)* 2016; 16(12):1983
- 3 McCorkle R, Ercolano E, Lazenby M, et al. Self-management: enabling and empowering patients living with cancer as a chronic illness. *CA Cancer J Clin* 2011;61(01):50–62
- 4 Penedo FJ, Oswald LB, Kronenfeld JP, Garcia SF, Cella D, Yanez B. The increasing value of eHealth in the delivery of patient-centred cancer care. *Lancet Oncol* 2020;21(05):e240–e251
- 5 Guo Y, Hao Z, Zhao S, Gong J, Yang F. Artificial intelligence in health care: bibliometric analysis. *J Med Internet Res* 2020;22(07): e18228
- 6 Balaji PG, Srinivasan D. An introduction to multi-agent systems. In: Srinivasan D, Jain LC, eds. *Innovations in MultiAgent Systems*



- and Applications. Vol. 310. Berlin, Heidelberg: Springer; 2010: 1–27
- 7 Sayyad Shirabad J, Wilk S, Michalowski W, Farion K. Implementing an integrative multi-agent clinical decision support system with open source software. *J Med Syst* 2012;36(01):123–137
  - 8 Li S, Mackaness WA. A multi-agent-based, semantic-driven system for decision support in epidemic management. *Health Informatics J* 2015;21(03):195–208
  - 9 Safdari R, Shoshtarian Malak J, Mohammadzadeh N, Danesh Shahraki A. A multi agent based approach for prehospital emergency management. *Bull Emerg Trauma* 2017;5(03):171–178
  - 10 Isern D, Moreno A. A systematic literature review of agents applied in healthcare. *J Med Syst* 2016;40(02):43
  - 11 Corkill D. Blackboard systems. *AI Expert* 1991;6(09):40–47
  - 12 Liang D, Yuan S. Structural health monitoring system based on multi-agent coordination and fusion for large structure. *Adv Eng Softw* 2015;86:1–12
  - 13 Straub J, Reza H. A blackboard-style decision-making system for multi-tier craft control and its evaluation. *J Exp Theor Artif Intell* 2015;27(06):763–777
  - 14 Mesa I, Sanchez E, Diaz J, et al. GoCardio: a novel approach for mobility in cardiac monitoring. *Impact: The Journal of Innovation Impact* 2013;1(06):110–120
  - 15 Falconelli N, Sernani P, Brugués A, et al. Indexing the event calculus: towards practical human-readable personal health systems. *Artif Intell Med* 2019;96:154–166
  - 16 HL7 International. Accessed February 2023 at: <https://www.hl7.org/>
  - 17 HL7 International. FHIR Release 4. Accessed February 2023 at: <https://www.hl7.org/fhir/>
  - 18 Mandel JC, Kreda DA, Mandl KD, Kohane IS, Ramoni RB. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *J Am Med Inform Assoc* 2016;23(05): 899–908
  - 19 Garcia SJ, Zayas-Cabán T, Freimuth RR. Sync for genes: making clinical genomics available for precision medicine at the point-of-care. *Appl Clin Inform* 2020;11(02):295–302
  - 20 Gordon WJ, Baronas J, Lane WJ. A FHIR human leukocyte antigen (HLA) interface for platelet transfusion support. *Appl Clin Inform* 2017;8(02):603–611
  - 21 Dorr DA, D'Autremont C, Pizzimenti C, et al. Assessing data adequacy for high blood pressure clinical decision support: a quantitative analysis. *Appl Clin Inform* 2021;12(04):710–720
  - 22 SNOMED International. SNOMED CT Browser. Accessed February 2023 at: <https://browser.ihtsdotools.org/>
  - 23 The National Library of Medicine. RxNorm. Accessed February 2023 at: <https://www.nlm.nih.gov/research/umls/rxnorm/index.html>
  - 24 The Regenstrief Institute. LOINC. Accessed February 2023 at: <https://loinc.org>
  - 25 Stram M, Seheult J, Sinard JH, et al; Members of the Informatics Committee, College of American Pathologists. A survey of LOINC code selection practices among participants of the College of American Pathologists Coagulation (CGL) and Cardiac Markers (CRT) proficiency testing programs. *Arch Pathol Lab Med* 2020; 144(05):586–596
  - 26 Gershenov M. The ICD family of classifications. *Methods Inf Med* 1995;34(1–2):172–175
  - 27 Wright A, Sittig DF. A four-phase model of the evolution of clinical decision support architectures. *Int J Med Inform* 2008;77(10): 641–649
  - 28 Shalom E, Goldstein A, Ariel E, et al. Distributed application of guideline-based decision support through mobile devices: Implementation and evaluation. *Artif Intell Med* 2022;129:102324
  - 29 Lisowska A, Wilk S, Peleg M. From personalized timely notification to healthy habit formation: a feasibility study of reinforcement learning approaches on synthetic data. In: 2021. SMARTERCARE Workshop Proceedings CEUR-WS. Vol. 360. 2021:7–18
  - 30 Karni L. Health Information Technology for Empowering Elderly Patients with Chronic Diseases. PhD Dissertation. Örebro University School of Business, Sweden; 2022. Accessed February 2023 at: <http://urn.kb.se/resolve?urn=urn:nbn:se:oru:diva-99556>
  - 31 Sutton DR, Fox J. The syntax and semantics of the PROforma guideline modeling language. *J Am Med Inform Assoc* 2003;10 (05):433–443
  - 32 The CAPABLE Team. Models of the CAPABLE resources. Accessed February 2023 at: <https://simplifier.net/capable-fhir-repository>
  - 33 Georgia Institute of Technology. The OMOP on FHIR Project. Accessed February 2023 at: <http://omoponfhir.org>
  - 34 Lanzola G, Polce F, Tibollo V, et al. Designing a testing environment for the CAPABLE telemonitoring and coaching platform. In: 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON). 2022:1112–1117
  - 35 Matney SA, Heale B, Hasley S, et al. Lessons learned in creating interoperable fast healthcare interoperability resources profiles for large-scale public health programs. *Appl Clin Inform* 2019;10 (01):87–95
  - 36 Long WJ. Medical informatics: reasoning methods. *Artif Intell Med* 2001;23(01):71–87
  - 37 Lanzola G, Gatti L, Falasconi S, Stefanelli M. A framework for building cooperative software agents in medical applications. *Artif Intell Med* 1999;16(03):223–249
  - 38 Gøeg KR, Rasmussen RK, Jensen L, Wollesen CM, Larsen S, Pape-Haugaard LB. A future-proof architecture for telemedicine using loose-coupled modules and HL7 FHIR. *Comput Methods Programs Biomed* 2018;160:95–101
  - 39 Lanzola G, Falasconi S, Stefanelli M. Cooperative software agents for patient management. In: Barahona P, Stefanelli M, Wyatt J, eds. *Procs of the 5th Conference Artificial Intelligence in Medicine Europe (AIME), Lecture Notes on Artificial Intelligence (934)*. Berlin: Springer; 1995:173–184
  - 40 Adlassnig KP, Haug P, Jenders RA. Arden syntax: then, now, and in the future. *Artif Intell Med* 2018;92:1–6
  - 41 openEHR. Open industry technology including specifications, clinical models and software for e-health. Accessed February 2023 at: <https://www.openehr.org/>
  - 42 Anani N, Chen R, Prazeres Moreira T, Koch S. Retrospective checking of compliance with practice guidelines for acute stroke care: a novel experiment using openEHR's Guideline Definition Language. *BMC Med Inform Decis Mak* 2014;14:39
  - 43 Fowler M. Inversion of control containers and the dependency injection pattern. Accessed February 2023 at: <https://martinfowler.com/articles/injection.html>
  - 44 Fowler M. Inversion of control. Accessed February 2023 at: <https://martinfowler.com/bliki/InversionOfControl.html>
  - 45 Scott Kruse C, Kareem P, Shifflett K, Vegi L, Ravi K, Brooks M. Evaluating barriers to adopting telemedicine worldwide: a systematic review. *J Telemed Telecare* 2018;24(01):4–12
  - 46 Stange KC. The problem of fragmentation and the need for integrative solutions. *Ann Fam Med* 2009;7(02):100–103
  - 47 Koch S. Home telehealth—current state and future trends. *Int J Med Inform* 2006;75(08):565–576
  - 48 Zanaboni P, Wootton R. Adoption of telemedicine: from pilot stage to routine delivery. *BMC Med Inform Decis Mak* 2012;12(01):1
  - 49 Norum J, Pedersen S, Størmer J, et al. Prioritisation of telemedicine services for large scale implementation in Norway. *J Telemed Telecare* 2007;13(04):185–192
  - 50 Bottrighi A, Piovesan L, Terenziani P. Supporting physicians in the coordination of distributed execution of CIGs to treat comorbid patients. *Artif Intell Med* 2023;135:102472
  - 51 Wallace E, Salisbury C, Guthrie B, Lewis C, Fahey T, Smith SM. Managing patients with multimorbidity in primary care. *BMJ* 2015;350:h176

- 52 Piette JD, Richardson C, Valenstein M. Addressing the needs of patients with multiple chronic illnesses: the case of diabetes and depression. *Am J Manag Care* 2004;10(2, Pt 2):152–162
- 53 Laleci Erturkmen GB, Yuksel M, Sarigul B, et al. A collaborative platform for management of chronic diseases via guideline-driven individualized care plans. *Comput Struct Biotechnol J* 2019;17:869–885
- 54 Dullabh P, Hovey L, Heaney-Huls K, Rajendran N, Wright A, Sittig DF. Application programming interfaces in health care: findings from a current-state sociotechnical assessment. *Appl Clin Inform* 2020;11(01):59–69
- 55 The EU Parliament. Regulation (EU) 2017/745 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC. Official Journal of the European Union. 2017. Accessed February 2023 at: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32017R0745>