Designing a Testing Environment for the CAPABLE Telemonitoring and Coaching Platform

Giordano Lanzola, Francesca Polce
Dept. Electrical, Computer and Biomedical Engineering
University of Pavia
27100 Pavia, Italy

Valentina Tibollo
Lab. of Systems Science and Clinical Research
Istituti Clinici Scientifici Maugeri
27100 Pavia, Italy

Silvana Quaglini
Dept. Electrical, Computer and Biomedical Engineering
University of Pavia
27100 Pavia, Italy

Szymon Wilk
Institute of Computing Science
Poznan Univ. of Technology
Poznan, Poland

***Abstract*—*Introduction*: The CAPABLE project has been funded by the European Union to develop a telemonitoring and coaching platform improving the quality of life for cancer patients. The platform, based on a multi agent blackboard architecture, is classified as a medical device according to the current EU regulation. Thus it needs extensive tests before being put into service for the planned clinical trials, which calls for a dedicated simulating and testing environment. *Materials and Methods*: Coordination in CAPABLE is achieved through the Case Manager, a component able to generate and notify events to other interested agent components. For representing and exchanging health care information we have adopted HL7 FHIR as a semantic interoperability standard. *Results*: FHIR has been exploited to design a structured history of a real patient affected by renal cell carcinoma. A simulator has been developed for automating the whole testing process represented by specific scenarios of the patient's history. *Conclusions*: The simulator relies on the events produced by the Case Manager for coordinating the agents. This proved to be effective in checking that the agents reactions to new data showing up on the blackboard comply with the expected behavior.**

***Index Terms*—Health technology assessment, distributed system testing, patient coaching, mHealth**

I. Introduction and Background

Cancer patients need to undergo several treatments sometimes extending throughout their lives [1] and often causing side effects. Delays or failures in detecting those side effects may compromise the situation and impose a temporary stop to the treatment that reduces or spoils its effectiveness. Moreover, some treatments require actions to be accomplished by the patient such as regularly taking drugs and medications, or simply being compliant with an appropriate and healthier lifestyle. If the patient does not receive an adequate support he/she may forget to adhere to those requirements [2].

Telemedicine helps in bridging the gap between the patient and the clinic staff allowing the regular submission of Patient Reported Outcomes (PROs) and Patient Reported Experiences (PREs) as soon as they are noticed [3]. Additional data useful for properly managing the patient may also come through smart devices or sensors. Those may automatically send data concerning the environment or even acquire it directly on the patient, possibly even playing an active role in therapy [4]. Also patients seem to appreciate those interventions by proactively using them since they feel being better cared of by the clinic staff [5].

Based on the above mentioned issues, the CAPABLE (CAncer PAtients Better Life Experience) project has been funded by the European Union within the Horizon 2020 programme for the years 2020-2023. The CAPABLE primary goal is to improve the quality of life of home cancer patients supporting them in addressing all the above mentioned issues. This is achieved setting up at the patient's clinic a GDPR (General Data Protection Regulation) compliant platform adhering to the same privacy and security policies of the clinical information system. CAPABLE makes available services such as reminders or even actions for coaching and motivating patients with the aim of improving their well being and enhancing the adherence to the treatment, thereby increasing the outcomes.

CAPABLE receives data from multiple sources: the electronic Health Record (EHR); patient questionnaires administered through smartphones; smartwatches automatically acquiring physical activity and vital signs; environmental sensors networks providing air quality data. For processing all the data acquired, CAPABLE foresees the combined support of knowledge-driven and data-driven methods. The knowledge-driven support is provided in form of a computational environment for enacting Computer Interpretable Guidelines (CIGs) that encapsulate the most recent medical knowledge for therapeutic plans. The data-driven method leverages instead techniques involving big data analytics. For example, data collected by wearable sensors (smartwatches) are used to develop and refine personalized models to predict most appropriate timing for well-being interventions [6].

From the above description it emerges that CAPABLE qualifies as a decision support system, which poses challenges related to its development and testing. As a matter of fact, according to current EU regulation, it is classified in risk class IIa [7] being a *"software intended to provide information*

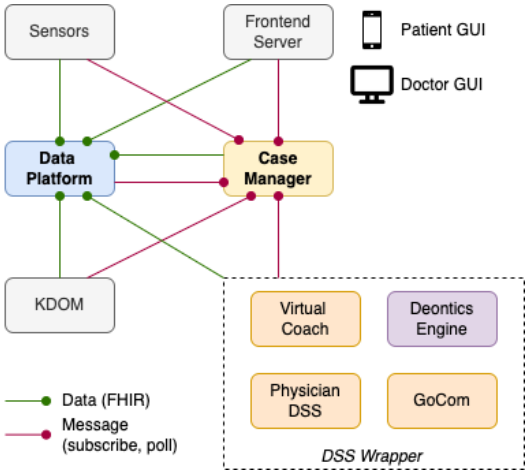

Fig. 1. The block diagram illustrating the overall architecture of the CAPABLE system including several components behaving as autonomous agents.

which is used to take decisions with diagnosis or therapeutic purposes". The regulation also mentions the need for a formal validation (conformity assessment) before putting the device into service. Thus we needed a way to check its performance throughout its development path. This paper describes the approach we are following for testing the CAPABLE system behavior which represents a prerequisite for the conformity assessment.

II. Materials and Methods

A. The Case Manager as an Event Notification System

The overall architecture chosen for CAPABLE is illustrated in Fig. 1 [8]. The block diagram shows that it encompasses a high number of loosely coupled components, behaving as autonomous agents. At its heart the Data Platform (DP) is located, playing the role of a central blackboard that makes patient data available to all the other components through the green links. The *DSS Wrapper* includes all components that are involved in providing comprehensive decision support to various groups of users. Specifically, the *Physician DSS* (PhDSS) focuses on physicians and the *Virtual Coach* (VC) aims at patients. They both rely on services provided by other components in the DSS Wrapper box – *GoCom* mitigates adverse interactions between already prescribed actions and possible recommendations while *Deontics Engine* executes CIGs in the PROforma language. Functionality of the PhDSS and VC may be accessed through the *Patient GUI* and *Doctor GUI*. The *Sensors* regularly feed the DP with pre-processed data acquired from wearable devices (smartwatches). Pre-processing involves extraction of relevant features and abstraction of temporal data. Finally, the *Knowledge-Data Ontology Mapper* (KDOM) is a component performing clinical abstractions on raw data and PROs concerning the patient.

Keeping all those components separate helped in managing the complexity of the CAPABLE system and simplified its software development. This approach is in line with the current practices pushing toward the separation of concerns among the components and the transition to distributed systems. In CAPABLE it has given more freedom to the developers in using specific tools, libraries and already existing solutions. Moreover, it facilitates the provision of comprehensive support timely delivered to patients and their caregivers, as advocated by the "five rights" of clinical decision support [9].

As a consequence of this architectural choice, the key problem we faced was how to coordinate the behavior of all its components. For example, since the DP is also implemented as a separate component, there was no easy way for the others to tell when new data became available to make progress in the management of a case. To address this issue an additional dedicated component was designed and implemented, named the Case Manager (CM) [10]. The CM, shown right of the DP in Fig. 1, acts as a notification service addressing the components involved in provisioning decision support and those serving as GUIs, as pointed out by the red links.

The CM is modeled after the *inversion of control* paradigm [11] in that it is provided without any prior knowledge about the notifications to be issued and therefore on the specific behavior of the components using it. Instead, a language for its dynamic configuration has been devised, so that the CAPABLE components may set up the combination of facts that need to be checked on the DP. Those patterns are encapsulated into *Event Rules*, that are the formalism made available by the language, and represent the *events of interest* for each component. Once they are discovered on the DP, the CM generates a notification. Thus the CM is the key element responsible for implementing the opportunistic reasoning process foreseen by CAPABLE that is achieved by continuously monitoring the occurrence of the events of interest on the DP and eventually forwarding the notifications about those to the components.

B. Sharing Patient Data through FHIR

CAPABLE collects information from different sources to provide decision support to physicians and coaching to patients. This information is integrated on the DP and is accessed by the various CAPABLE component agents that all run inside the network of the clinic where the patient is treated to avoid privacy/security issues. Thus, a key issue requires addressing interoperability to ensure that all the components consistently interpret that information. Moreover, all the components must share the same terminologies and use them in the same contexts. To achieve this goal in CAPABLE we adopted HL7 FHIR (Fast Healthcare Interoperability Resources) that proposes as a semantic interoperability standard for exchanging health care information among software applications [12].

FHIR foresees the representation of health care data in terms of modular elements called *Resources*. Each resource defines a specific information concerning the health care context including all the relevant attributes and constraints required to properly convey its meaning. The Resources foreseen by FHIR represent the most commonly used concepts (e.g. Patient, Observation, Medication, etc.) and convey a small amount

of information when taken separately. However, they have the ability to cross-reference each other thus creating an arbitrarily complex semantic network of concepts. Moreover, from a technical viewpoint all the resources share the same implementation model which is formally specified by the standard starting from the simplest data types and proceeding hierarchically to the highest levels. This greatly simplifies their processing and exchange between different applications and several libraries have been developed for that purpose.

In CAPABLE we use the HAPI FHIR library (https://hapifhir.io) which is available in public domain for the Java platform. Moreover, since we had to choose the resource set fitting the scope of the project, we started this selection process analyzing some guidelines prepared by the European Society for Medical Oncology (ESMO) [13]. The process has been subsequently refined implementing several scenarios addressing a real patient which have been used to provide demonstrations of the project during the periodic reviews. Presently, those scenarios are being used for the simulation tests representing the subject of this paper, as discussed in Section III-A, that are mandatory for its conformity assessment in sight of the foreseen pilot trials. The following FHIR resources have been already modeled in CAPABLE: *Patient*, *Observation*, *Communication*, *MedicationRequest*, *Goal*, *List* and *QuestionnaireResponse* with some more expected to follow soon.

III. Results

A. Implementation of a Structured Patient History

For the purposes of implementing, testing and demonstrating CAPABLE we have reconstructed the clinical history of a patient currently treated at the *Istituti Clinici Scientifici Maugeri* (ICSM) which is a major research hospital located in Pavia, Italy. The patient is suffering from renal cell carcinoma and the history spans from the initial diagnosis to the last follow-up over a period of about 13 months. Data was manually collected from the following different sources:

- The Agenda: a section of the Hospital Information System (HIS) with schedules for visits and treatments;
- The Renal Cell Carcinoma (RCC) Registry: an eCRF developed using the REDCap software, reporting the tumor characteristics at the onset, information about oncological therapy lines, metastases, toxicities and their impact on the treatment.
- Outpatient visit reports including follow up information. Usually during an outpatient visit, patients also receive the oncological treatment.

The first two sources contain structured data, while the last one is a free-text source. After merging all the information and organizing it in chronological order, the history was represented as a flat timeline, as shown in Fig. 2.

Note that the enrollment in the CAPABLE system is supposed to have occurred during the first oncological visit of the patient at the ICSM hospital (01/10/2020) in line #10. By that time past events are also reported with their actual dates, because they are important for the decision support components – PhDSS and VC. Those events include for example the duration of a comorbidity period or information about the radiotherapy the patient underwent some days before.

After preparing the patient timeline, all the events of interest for any decision support component available in CAPABLE have been identified and selected. Finally, they have been manually translated into FHIR resources. Those are the events to which the different components are subscribed with the CM and whose occurrences generate one or more recommendations or alerts or reminders, targeting patients, doctors, or both. For example, VC enacts rules encapsulating knowledge represented in clinical guidelines in order to generate relevant recommendations for patients. A sample rule for patients undergoing radiotherapy (RT) is:

```
IF RT is ongoing OR
      finished less than 6 months ago
THEN recommend to avoid sun exposure in
  hottest hours AND
  recommend to protect irradiated body
  areas with silk/cotton clothes/foulards
```

As agreed with clinicians and patients participating in the process of establishing requirements for the CAPABLE system, recommendations mentioned in the above rule should be delivered once every 30-days to avoid overloading a patient with information (we rely on a similar consensus in all other cases when timing has not been specified explicitly). This can be achieved by specifying an appropriate expiration period for a CM event triggering rule.

Reminders and notifications are also generated by the VC for new treatments being prescribed or existing treatments being revised (see lines #11 or #14 in Fig. 2). Their generation is triggered by a CM event rule that checks for new *MedicationRequest* resources. The CAPABLE system also offers alerts that are issued at specific times of the day to remind the patient to take a drug (this exact timing is stored in *MedicationRequest* resources). This closely recalls the functionality of an alarm clock and is better managed by the *Patient GUI* via *snooze* and *dismiss* buttons.

Another reminder type is related to entries such as the one at line #12 in Fig. 2 when the patient must be reminded in advance to go to the hospital. In this case the reminder is triggered by a CM rule that checks for *Appointment* resources (imported to DP from the HIS Agenda). Following the consensus, it was decided to have this rule look ahead for appointments scheduled in the next 24 hours so that the reminder is generated only once the day before.

B. The Simulator

In the CAPABLE multi agent architecture, each component reacts by saving new information to the DP based on the results previously added by other components. Thus, we soon needed a tool for generating the preconditions for the intervention of each component in support of the integration process.

	Patient #11	M, 01/01/1958	Event
1	2019	Hypercholesterolemia	Comorbidity
2	2019	Atorvastatin 10 mg, 1 pill 3 times a week	Therapy 1st prescription
3	26/08/2020	Pantoprazole (es. Pantorc) 20 mg, 1 pill in the morning	Therapy 1st prescription
4	26/08/2020	Initial diagnosis of metastatic disease for renal cell carcinoma	Diagnosis
5	26/08/2020	metastasis in lungs	Diagnosis
6	03/09/2020	Pregabalin 50 mg, 1 pill in the morning	Therapy 1st prescription
7	03/09/2020	oxycodone/paracetamol (es. Depalgos) 5/325 mg, 2 pill/die	Therapy 1st prescription
8	16/09/2020	Hyper fractioned radiotherapy start (bones, 5 number of fractions, 20 Grey)	Therapy 1st prescription
9	22/09/2020	Hyper fractioned radiotherapy stop	Therapy stop
10	01/10/2020	FIRST ONCOLOGICALVISIT in ICSM	Visit
11	02/10/2020	oxycodone/paracetamol 5/325 mg, 3 pill/die	Therapy change
12	05/10/2020	BONE SCINTIGRAPHY	Diagnostic test
13	06/10/2020	ONCOLOGICAL VISIT	Visit
14	06/10/2020	administered paracetamol 1000 mg in 100 cc of physiological solution	Therapy admin. at the hospital
15	06/10/2020	abdomen CT scan, brain CT scan, chest CT scan, blood tests, urinalysis	Diagnostic test
16	07/10/2020	ONCOLOGICAL VISIT	Visit
17	07/10/2020	Zoledronic Acid (4 mg in 100 cc of physiological solution in intravenous hydration)	Therapy admin. at the hospital
18	13/10/2020	blood test	Diagnostic test
19	14/10/2020	Non-occlusive thrombosis of the deep femoral vein on the left leg	Diagnosis
20	14/10/2020	enoxaparina 6.000 U.I.aXa/0.6 ml subcutis, every 12 hours, for at least 3 months	Therapy 1st prescription

Fig. 2. An excerpt of the clinical history of a renal cancer patient currently treated at *Istituti Clinici Scientifici Maugeri* (data have been pseudonymized by adding some *noise* to dates and numerical values).

Time plays a crucial part in the reasoning process since medical data is time-stamped and components use that information in applying their knowledge. Nevertheless, since the components run based on the ”wall clock time”, simulating the acquisition of data in the past produced spurious results. Even the CM had problems since it had to notify that the events noticed on the DP occurred in the past.

We first addressed those issues changing the ”wall clock time” for the components. However dynamically changing the time at the operating system level proved to be impossible, due to delays before changes were acknowledged by the software platforms where each component was running. Moreover, the components run on different machines over which we had limited or no control at all. We then introduced specific events signalling time change occurrences but this approach showed technical and conceptual drawbacks. Situations arose when a component processing an event had not yet switched to the new time while the one generating the preconditions already did, meaning that consequences could be stamped with a time predating their own preconditions.

Relying on the central role played by the CM in generating and dispatching events, we finally designed a simulator meant to test the component reactions to their events of interest. This approach seemed the most appropriate based on the patient history discussed in Section III-A as we asked the developers to intersperse that history with the FHIR resources that their components were supposed to add. Segmenting the patient history and considering each resource added by a component as a *checkpoint* to be asserted greatly simplified the simulation process based on the following assumptions:

- Stateless behavior. Components must not internally cache any information required to undertake any action in their reasoning process.
- Mediated interaction. Components must only interact writing resources on the DP, even if they just need to exchange private messages which are represented through Communication resources.
- Timely reaction. The CAPABLE system is not strictly speaking a real time one, but the CM generates and dispatches events as soon as their preconditions show up on the DP. Components are also required to promptly process events and write back new Resources to the DP.

Failing to comply with those requirements would spoil the component ability of reacting to the events generated by the CM or the possibility of promptly observing their behavior. With those assumptions the behavior of the simulator is structured on the following workflow:

- The patient history augmented with checkpoints is provided as input to the simulator;
- The simulator automatically segments the history into several legs, each one starting from the very beginning of the history and ending with a checkpoint;
- The simulator iterates on each leg emptying the DP and loading the partial history referring to the leg. Event generation by the CM is suppressed and resource times are shifted in the past so that the ending checkpoints appear to be positioned at the current ”wall clock time”.
- The last portion of resources, supposed to trigger the components reactions represented by the checkpoints, are loaded last, after enabling event generation by the CM;
- The simulator stops for a while for *asserting the checkpoint* (i.e. testing if the resource represented by the checkpoint actually shows up on the DP);
- The process is repeated for the next leg, properly recording the results of the current one.

Summarizing, the simulator has been conceived after the

paradigm of *Unit Testing* which is widely adopted in software development. As such it is not provided with a graphical user interface. It is fed instead, as input, with the scenarios that are designed externally, in terms of FHIR resources, augmented with the checkpoints. It then produces, as output, a report including all the failed checkpoint assertions. Thus, whenever any checkpoint fails to be asserted, it becomes trivial tracking down the problem to the single component and context causing it. The simulator completely automates the whole process and the only effort deals with preparing the patient histories interspersed with checkpoints. Loading all the resources in the past avoids the need to tamper with time since each leg always ends at the current "wall clock time" when checkpoints are tested. Since component behavior is stateless, they only react to the events generated by the last portion of Resources loaded. However, once invoked, they have full visibility on the past patient history stored on the DP.

C. Simulation Example

Below we present an example of how the proposed simulator is applied to check for relevant recommendations, notifications and reminders that should be generated for the patient history discussed in Fig. 2 (later in the text we refer to specific lines of that history). For the sake of consistency we focus on VC, however, the same testing procedure is applicable to other components in the CAPABLE system. Specifically, we consider a simulation scenario for feedback provided by VC that covers the following checkpoints:

- checkpoint 1: a recommendation related to a recent RT (see the example rule) that should be generated after the first visit to ICSM (line 10) when the patient has been enrolled and their data have been entered into the CAPABLE system;
- checkpoint 2: a notification about a revised pharmacological treatment that should be generated after increasing the dose of oxycodone/paracetamol (line 11);
- checkpoint 3: a reminder about an upcoming visit to ICSM in order to perform bone scintigraphy (line 12) that should be generated 24 hours in advance.

In the simulation scenario we use the FHIR resources listed in Table I. Most of these resources represent information provided in the history shown in Figure 2 (the corresponding line is indicated in the history columns). The list also includes additional resources that are not explicitly captured in the history: *r01* – the *Patient* resource, and *r12*, *r14* and *r16* – *Communication* resources with the feedback generated by VC. The latter three resources also constitute checkpoints to be asserted by the simulator. For simplification, we assume the history starts on 31-12-2019 and the number given in the offset column represents a time offset (in days) between the date of a specific resource and the patient history start. For example, the offset for *r11* that captures the first visit to ICSM is 275, as this is the time span between 31-12-2019 (the history start) and 1-10-2020 (the visit date). These offsets are used to shift resources in the past as explained in the previous section. Here we need to explain that in practice offsets are more fine-grained (they can be specified in hours and minutes) – we only round them to days here to simplify the presentation. Moreover, since the DP does not support the *Procedure* resource, we represent RT as *Observation*.

As per the workflow presented in Section III-B, the simulator starts with checkpoint 1 (see Fig. 3a). It empties DP and loads resources *r01 – r10*. The times of specific resources are adjusted according to their offsets (see Table I) in reference to the "wall clock" time denoted as *t0*. These resources are loaded in a special event-suppression mode so that no events are raised by CM. Then DP loads *r11* in a regular mode, that triggers a CM event that is captured by VC. In response VC executes the rule for RT and generates an RT-related recommendation (the patient underwent RT in the last 60 days, as indicated by *r09* and *r10*, and no such recommendation has been provided). This recommendation is stored as a *Communication* resource. The simulator asserts checkpoint 1 verifying whether the DP contains the resource *r12* associated with this checkpoint and logs the result.

Then, the simulator proceeds with checkpoint 2 (see Fig. 3b). It clears DP, and loads resources *r1 – r12* in the event-suppression mode – their times are moved by 1 day further in the past to account for passing time. Note that resources loaded at this stage also include *r12* that constituted the previous checkpoint. Then, the simulator loads *r13* that represents a modified prescription. This triggers a CM event and in response VC based on its rules creates and stores *r14* with appropriate notification. Although RT (*r10*) is still in the 60-day window, no new RT-related recommendation is generated by VC, as the last one was issued on the previous day. The simulator asserts checkpoint 2 verifying on the DP the presence of the associated resource (*r14*) and logs the result.

Finally, the simulator processes checkpoint 3 (see Fig. 3c). It first loads (with events suppressed) resources *r01 – r14*. Then it stores *r15* that represents an upcoming visit for bone scintigraphy that is scheduled for the next day. This triggers a CM event and in response VC generates a visit reminder and stores it in the DP as *r16*. As previously, VC creates no RT-related recommendation, since the previous one (*r12*) was provided 3 days earlier. Once again, the simulator asserts checkpoint 3 verifying if the DP contains the associated resource (*r16*) and logs the result.

IV. Conclusions and Future Work

This paper describes the approach we are following for performing a thorough conformity assessment of the CAPABLE system in sight of the pilot trials that are expected to take place during the year 2024 at two clinical settings. Given the time frame available the work on the simulator is still in progress.

We presented a single simulation scenario focused on VC, but the same approach is applicable to other components. Extending the scenario to cover the entire CAPABLE system will require introducing additional FHIR resources and defining relevant checkpoints (e.g., corresponding to recommendations generated by PhDSS). At the same time, no other changes to

TABLE I
RESOURCES USED IN THE SIMULATION EXAMPLE.

	Offset	Resource type	History
r01	0	Patient	
r02	0	Observation	1
r03	0	MedicationRequest	2
r04	239	MedicationRequest	3
r05	239	Observation	4
r06	239	Observation	5
r07	247	MedicationRequest	6
r08	247	MedicationRequest	7
r09	260	Observation	8
r10	266	Observation	9
r11	275	Appointment	10
r12	275	Communication	
r13	276	MedicationRequest	11
r14	276	Communication	
r15	279	Appointment	12
r16	278	Communication	

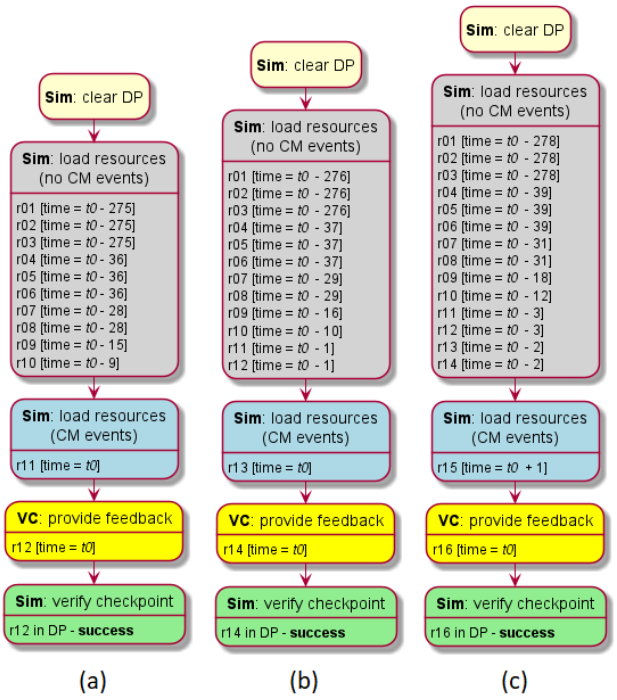

Fig. 3. Workflow in the simulation example (Sim = simulator): (a) checkpoint 1 - recommendation related to a recent RT; (b) checkpoint 2 - notification about a revised prescription; (c) checkpoint 3 - reminder about an upcoming visit.

the simulator will be necessary, as it is isolated from other components and relies solely on CM and DP.

We are aware that the simulator in its present form is affected by some limitations. For example, it is driven only by CM events and is unable to check any component reaction due to the plain time flow. Some support is already available in the CM for generating events caused by the lack of any incoming information over time. However this works on a time scale of a day or more, and it is meant to capture situations usually found in guidelines such as *if symptom X is not observed after taking drug Y then .....* It certainly does not fit scenarios where reminders or alarms should be reissued in a short time period. A possible mitigation would envision sending out special time flow events only after any information has been added to the DP and the regular simulation based on events has ended. Clearly, separating the event-driven and the time-based simulation, with the latter coming last, allows to avoid most of the criticalities that were initially experienced. Moreover, since in production no time flow events will ever be generated, their use does not contradict in principle our design requirement of testing components in their final shape.

ACKNOWLEDGMENT

This work was funded by the EU Horizon 2020 Program for Research and Innovation over the years 2020-2023 under grant agreement No. 875052 to the CAPABLE project.

REFERENCES

[1] J. L. Phillips and D. C. Currow, "Cancer as a chronic disease," *Collegian*, vol. 17, no. 2, pp. 47–50, 2010.

[2] H. Gönderen Çakmak and S. Kapucu, "The effect of educational follow-up with the motivational interview technique on self-efficacy and drug adherence in cancer patients using oral chemotherapy treatment: A randomized controlled trial," *Semin Oncol Nurs*, vol. 37, p. 151140, 2021.

[3] N. Nixon, E. Spackman, F. Clement, S. Verma, and B. Manns, "Cost-effectiveness of symptom monitoring with patient-reported outcomes during routine cancer treatment," *J Cancer Policy*, vol. 15, pp. 32–36, 2018.

[4] G. Lanzola, D. Capozzi, N. Serina, L. Magni, and R. Bellazzi, "Bringing the artificial pancreas home: Telemedicine aspects," *J Diabetes Sci Technol*, vol. 5, no. 6, pp. 1381–1386, 2011.

[5] E. Losiouk, G. Lanzola, S. Del Favero, F. Boscari, M. Messori, I. Rabbone, R. Bonfanti, A. Sabbion, D. Iafusco, R. Schiaffini, R. Visentin, S. Galasso, F. Di Palma, D. Chernavvsky, L. Magni, C. Cobelli, D. Bruttomesso, and S. Quaglini, "Parental evaluation of a telemonitoring service for children with type 1 diabetes," *J Telemed Telecare*, 2017.

[6] A. Lisowska, S. Wilk, and M. Peleg, "Catching patient's attention at the right time to help them undergo behavioural change: Stress classification experiment from blood volume pulse," in *Proceedings of the 19th International Conference on Artificial Intelligence in Medicine, AIME 2021*. Springer, 2021, pp. 72–82.

[7] The European Parliament, "Regulation (eu) 2017/745 of the european parliament and of the council of 5 april 2017 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," http://data.europa.eu/eli/reg/2017/745/2020-04-24, 2017, last accessed Jan 2022.

[8] E. Parimbelli, M. Gabetta, G. Lanzola, F. Polce, S. Wilk, D. Glasspool, A. Kogan, R. Leizer, V. Gisko, N. Veggiotti, S. Panzarasa, R. de Groot, M. Ottaviano, L. Sacchi, R. Cornet, M. Peleg, and S. Quaglini, "CAncer PAtients Better Life Experience (CAPABLE) first proof-of-concept demonstration," in *Proceedings of the 19th International Conference on Artificial Intelligence in Medicine, AIME 2021*, 2021, pp. 298–303.

[9] R. Campbell, "The five "rights" of clinical decision support," *J AHIMA*, vol. 84, pp. 42–47, 2013.

[10] F. Polce, G. Lanzola, M. Gabetta, E. Parimbelli, S. Wilk, D. Glasspool, R. Leizer, A. Kogan, and S. Quaglini, "The Case Manager: Driving medical reasoning in a distributed environment for home patient monitoring," in *Public Health and Informatics: Proceedings of the Medical Informatics Europe Conference*, 2021, pp. 610–614.

[11] M. Fowler, "Inversion of control containers and the dependency injection pattern," https://martinfowler.com/articles/injection.html, 2004, last accessed Jan 2022.

[12] R. Saripalle, C. Runyan, and M. Russell, "Using HL7 FHIR to achieve interoperability in patient health record," *J Biomed Inform*, vol. 94, p. 103188, 2019.

[13] P. Bossi, A. Antonuzzo, N. Cherny, O. Rosengarten, S. Pernot, F. Trippa, U. Schuler, A. Snegovoy, K. Jordan, and C. Ripamonti, "Diarrhoea in adult cancer patients: ESMO clinical practice guidelines," *Annals of Oncology*, vol. 29, pp. iv126–iv142, 2018.