Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 113 (2017) 376–383

Procedia Computer Science

www.elsevier.com/locate/procedia

The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2017)

Supporting process execution by interdisciplinary healthcare teams: Middleware design for IBM BPM

Nihan Çatal[a], Daniel Amyot[a,b,*], Wojtek Michalowski[b,c], Mounira Kezadri-Hamiaz[d], Malak Baslyman[a], Szymon Wilk[c,e], Randy Giffen[f]

[a]*School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Ave., Ottawa, ON, K1N 6N5, Canada*
[b]*Institut du savoir Montfort, 202-745A Montreal Road, Ottawa, ON, K1K 0T1, Canada*
[c]*Telfer School of Management, University of Ottawa, 55 Laurier Ave. East, Ottawa, ON, K1N 6N5, Canada*
[d]*College of Computer Science and Engineering (CCSE), Taibah University, PoBox 344, 41477, Al Madinah, Kingdom of Saudi Arabia*
[e]*Institute of Computing Science, Poznan University of Technology, oPiotrowo 2, 60-965 Poznan, Poland*
[f]*IBM Canada, 3755 Riverside Dr, Ottawa, ON, K1V 1B8, Canada*

Abstract

Interdisciplinary healthcare teams (IHTs) are involved in clinical processes composed of tasks requiring specific capabilities from different disciplines, often executed at different times. Although some hospitals use Business Process Management (BPM) suites to support their clinical activities, these tools are often unable to support the dynamic and capability-based allocation of tasks to the most suitable practitioner during the execution of a given process. Extensions, such as our previous work on ontological frameworks, exist that enable reasoning about IHT dynamics and allocate tasks to practitioners on the fly, but they are either unable to interact with commercial BPM suites or they are tightly integrated to one specific BPM suite. This paper contributes an innovative BPM-oriented middleware that enables existing BPM suites to interact with a semantic layer, hence offering more opportunities for deploying such advanced functionality in healthcare organizations. As a proof of concept, the middleware implementation is connected to a specific semantic layer and a specific commercial BPM suite (from IBM). The resulting system is illustrated with an acute stroke management process, hence demonstrating the feasibility of the proposed middleware-based approach. This solution compares advantageously against related work.

Keywords: Business process management; generic interface; IBM BPM; interdisciplinary healthcare teams; middleware; team dynamics

* Corresponding author. Tel.: +1 613-562-5800 ext. 6947; fax: +1 613-562-5664.
E-mail address: damyot@uottawa.ca

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.08.350

1. Introduction

Interdisciplinary healthcare teams (IHTs) are composed of practitioners from different disciplines who collaborate in order to provide optimal care to patients. IHTs enable collaborative and coordinated services in the healthcare system[1] and, in areas such as chronic pain management, raise the effectiveness of patient treatment[5]. However, as care processes (or workflows) can easily span many hours, days, or even months, the composition of IHTs usually changes over time. Another source of change is the real-time availability of practitioners (physicians, nurses, etc.) with specific capabilities required by the tasks of a care process, as these tasks are about to get executed.

The continuous selection of appropriate team leaders and members, and their allocation to tasks in a patient's care process, represent challenges for the automated support of IHT dynamics. This is particularly true in a context where *Business Process Management* (BPM) suites are used to model healthcare clinical processes and manage their execution. Commercial off-the-shelf and open-source BPM suites provide many process definition and execution features useful in a healthcare environment, as well as collaborative features such as mobility support and instant communication. For instance, The Ottawa Hospital (in Ontario, Canada) automated some of their traditional clinical processes with IBM's Business Process Manager suite (IBM BPM)[10] and also added support for mobility and collaboration with its integrated platform[9]. Yet, BPM suites do not necessarily have the functionalities needed to describe and reason about *team dynamics*, where team members and leaders change over time. The problem is that such tools need to be supplemented with additional functionalities to associate and dismiss, dynamically, suitable practitioners with the tasks being part of these clinical processes. Such functionalities can be modelled in many ways, including with a *semantic layer*[17] that defines concepts for dynamic team management while supporting the real-time allocation of process tasks to available and relevant practitioners.

An intermediate middleware, with a well-defined interface, is a potential way to integrate such a semantic layer with a BPM suite in order to capture complex healthcare processes and support dynamic IHT collaboration. This paper contributes such a middleware layer, including a *Generic Engine and Semantics Interface* (GESI), which decouples the semantic layer from underlying BPM suites while enabling the addition of dynamic IHT management features to existing BPM suites (commercial or open source), hence adding value to tools already used in healthcare. We have used Hevner et al.'s Design Science Research approach[8] to develop and assess this middleware artefact[3].

In this paper, Section 2 provides related work on IHT and BPM. Section 3 presents the design of the middleware, including GESI. The middleware's implementation and its interaction with a commercial BPM suite (IBM BPM[10]) are discussed in Section 4. Section 5 demonstrates how the middleware helps support typical IHT scenarios related to an acute stroke management process, and compares our approach to related work. Section 6 presents conclusions.

2. Related Work

IHTs are needed in healthcare to follow patients in a continuous way (e.g., as for chronic diseases) and often to treat an increasing aging population with complex care needs. Many problems occur in relation to a lack of a) sufficient collaborative work of healthcare teams towards common goals such as incomplete specifications of responsibilities, b) continuity in teams working in shifts, c) information about practitioners' capabilities, and d) clarity about work assigned[6]. To address such problems, real-time capability-based assignments can be achieved with an ontological framework combined with BPM[7]. An ontology for supporting teams is useful since team members' capabilities and responsibilities need to be clear in order to optimize the team's efficiency.

In her literature review, Çatal collected many goals and requirements for solutions in that context[3]. The language used to define clinical processes needs to be standard and well supported, with sufficient expressiveness. A common choice here is the *Business Process Model and Notation* (BPMN)[12]. The ontology must support healthcare concepts (patients, practitioner, capabilities, etc.) and team dynamics with role variability and user/patient preferences (especially in a patient-centric context). The team leader, also called *most responsible physician* (MRP), needs to be identified, with possible updates. The solution must allow the assignments of tasks to the most appropriate members based on their availability and capabilities. Both task-level and process-level assignments are needed, and some tasks/processes might be urgent. As healthcare teams care for patients who have a wide diversity of issues not always foreseen by clinical processes, the handling of process exceptions is another important feature. Finally, a BPM solution must integrate user interfaces and collaborative work support.

In terms of existing generic approaches, Papapanagiotou et al. pioneered a framework to support collaborative work of healthcare teams by assigning tasks to the most appropriate healthcare team members based on their capabilities[13]. However, their framework was not tested on workflows. Schmidt and Kunzmann developed a human resource framework to combine competence management and knowledge management[15]. Their ontology provides links to the business processes that are connected to a competency; however, their framework does not support team dynamics and process-level assignments.

There are also studies about the workflow management automation of IHTs. Prinyapol et al. developed a *Dynamic Platform for Workflow Management* (DPWFM)[14] manage requirement workflows dynamically and role variability (including leaders). However, the assignment of the tasks is done manually by the leader. Cabanillas et al.[2] proposed a team composition and allocation approach based on team member capabilities and BPMN processes. The allocations are done at runtime and concepts have support for healthcare, including for roles and delegation hierarchies. The focus of this team composition approach is however limited to the task level and it does not have an implementation based on a BPM engine.

Wilk et al. developed a framework[16] to support IHT dynamics, which includes an ontology that integrates workflow, patient, and IHT concepts and relations, in a *semantic layer* that also includes a rule engine. Practitioners who are members of the team execute tasks assigned to them based on their capabilities, competency levels, and availability. Capabilities are represented as facts (e.g., draw_blood_sample or conduct_invasive_therapy) that are associated with practitioners, and each practitioner can have multiple capabilities. The competency level is a value indicating the expertise of the practitioner (e.g., novice, regular, expert) and is used to assess potential members of the team. The availability of practitioners (available, busy, or unavailable) is also monitored and taken into consideration in the assignment procedure. However, the developed system, called MET4, was implemented as a *multi-agent system*[17], in lieu of a BPM suite, this it did not benefit from a BPM suite's integrated collaborative environment.

In terms of BPM suites, many commercial and open source solutions exist. Most will offer ways to model, instantiate, and monitor processes, define user interfaces for the process tasks accessible through Web browsers or mobile clients, manage access control, and support application programming interfaces (APIs) or web services for externally controlling some of these aspects. IBM BPM[10] is a leading commercial suite in that space, with many additional features, e.g., for social interactions among process participants (via instant messaging). This tool, like its competitors, does not support team dynamics well. Dermler et al.[4] extended IBM BPM to support dynamic teams with a team concept that is pre-defined and with members dynamically allocated to associated process/tasks. Some filters are used in these allocations to obtain a set of members who are capable of executing the process/tasks. Pre-defined teams are selected at design time and the filtering mechanism selects from these sets of users who will perform the tasks at run-time. However, IHTs need more flexibility than this as a particular practitioner can be assigned to any process task if he/she is capable of executing. Moreover, frequently changing leaders should be considered and the practitioners should be able to join/leave teams when needed.

Given this existing body of work, the problem we are tackling here is how to reuse existing IHT ontologies and dynamic team management solutions to augment the capabilities of feature-rich BPM suites. Our objective is to develop a middleware to have these existing tools interact in a decoupled way. As we have experience with one of the leading IHT ontologies (Wilk et al.[17]), this semantic layer is what we use here to manage IHTs. The middleware developed in the next section is independent from the semantic layer and from the underlying BPM suite. However, our solution will be tested with IBM BPM, a commercial BPM suite already used by our many hospital partners.

3. Middleware Design and Generic Interface

A middleware is a software that enables incompatible applications to interact, often via APIs and messaging. In our case, the desired middleware enables an upper, *semantic layer*, to interact with a lower, *execution layer* (Fig. 1). The semantic layer used is the one described by Wilk et al.[17]. It is composed of an IHT ontology with an instance base (collecting facts about processes, practitioners, patients, etc.) and behavioral rules expressed with logic. These semantic components are used by a reasoner (Microsoft Z3 here, https://github.com/Z3Prover/z3) to find solutions to real-time, capability-based allocation problems, which can in turn update the content of the instance base. The execution layer is where the BPM suite resides, together with other hospital information systems (HISs).

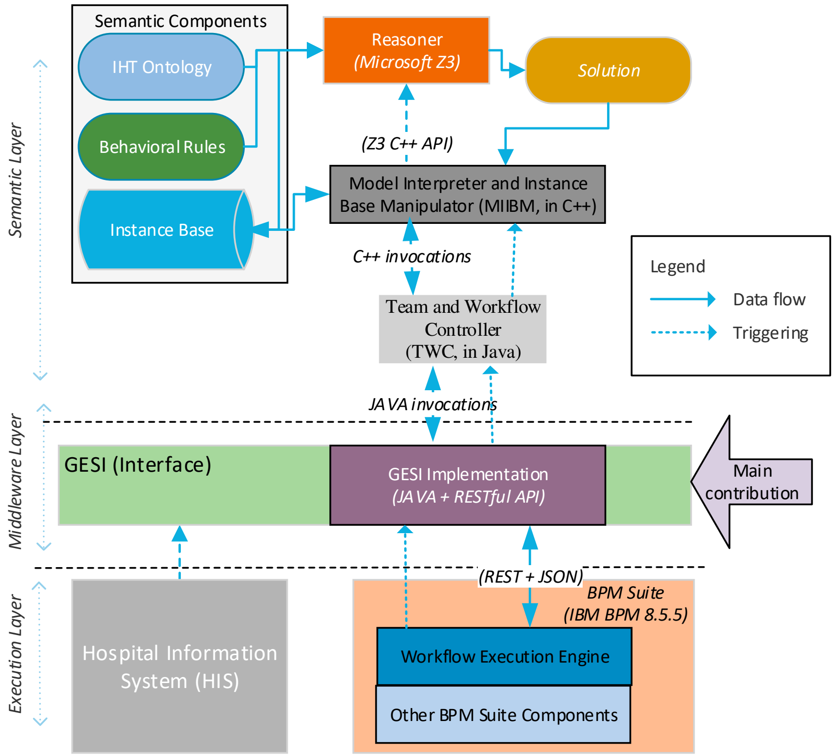

Fig. 1. Architecture overview of the semantic, middleware, and execution layers, with some selected technologies.

Many of the goals and requirements enumerated in the previous section are met by the semantic layer (definition of healthcare concepts and management of team dynamics) and the execution layer (process language support and expressiveness, and collaboration). The other goals are only partially met by these layers. The original contribution of this paper resides in the development of the middleware layer, and in particular of its *Generic Engine and Semantic Interface*, which complements the other layers in addressing the remaining goals (capability-based allocation, task assignments, leaders, exceptions). GESI's documentation is available online (https://goo.gl/CtUuor).

GESI provides seven methods, which rely on two predefined data structures (Fig. 2). GESI's methods are meant to be minimal while being generic enough to map easily to the APIs and services of a wide variety of BPM suites. The middleware's responsibility is to translate and relay the invocations and replies between the layers, and to coordinate the latter. Fig. 3 describes the typical usage of the interface by the other layers.

The method `createPatient` is triggered by the execution layer, e.g., by a HIS, as a patient with a given presentation requires the corresponding care process to be initiated, with a relevant IHT. In turn, the middleware relays that information to the semantic layer that selects a presentation-specific process and creates (initially empty) IHT to manage the given patient. Then, the semantic layer invokes the method `startWorkflow`, and the middleware then calls the corresponding API or service in the BPM suite to instantiate the required care process (the semantic layer is aware of the care processes available through the BPM suite). `getTaskList` is initiated by the semantic layer and causes the middleware to query the BPM engine for the next tasks to be performed for that process instance. As this information is relayed by the middleware back to the semantic layer, `assignMPR` and `assignTask` can be invoked according to the solutions computed by the semantic layer to set the MRP (leader) and to assign the next tasks to the most appropriate practitioners (according to capability matching, availability, patient preferences, continuity of care by previous IHT members, physical distances, etc.). `getWorkflowInstanceDetails` collects the status of the tasks to be performed (e.g., are they completed?). At any time after a process is instantiated for a patient, `reportException` can be invoked by the BPM suite, and the exception is relayed by the middleware to the semantic layer so that the exception can be handled (e.g., by finding another team leader/member, or by cancelling a process).

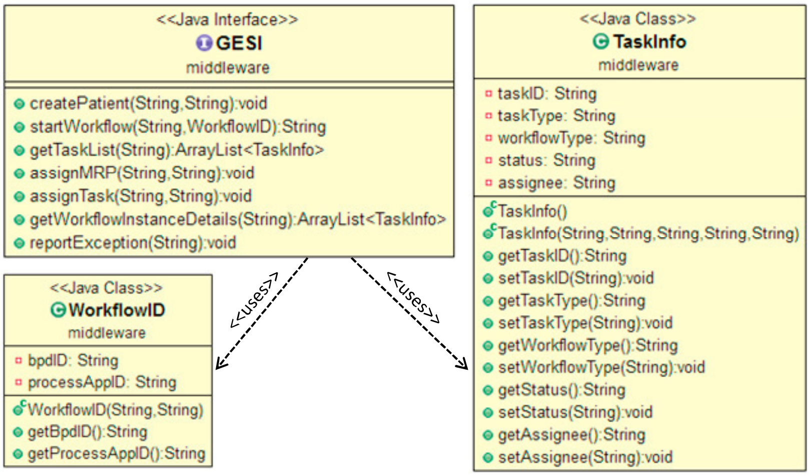

Fig. 2. Generic Engine and Semantic Interface (GESI).

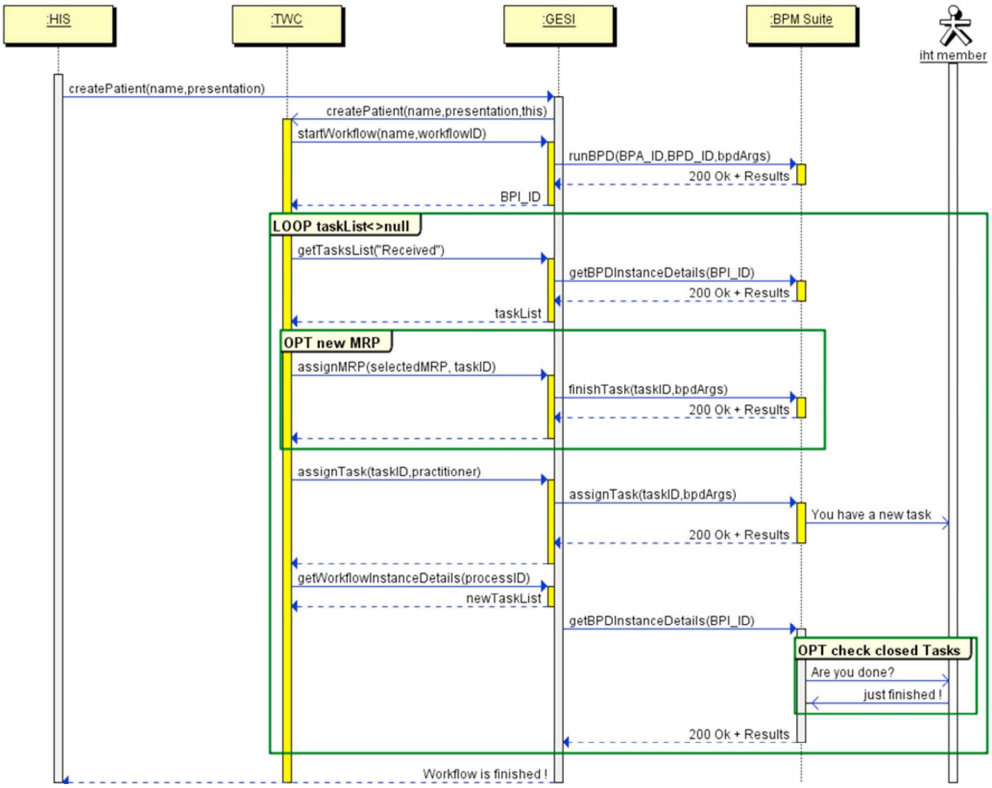

Fig. 3. Sequence diagram describing GESI's typical usage, with IBM BPM as a BPM suite example. "200 Ok" is a confirmation from IBM BPM.

The two classes defined Fig. 2 are `WorkflowID`, used to convey unique identifiers of clinical process definitions and their instances, and `TaskInfo`, which captures identifiers for task definitions and instances, the process to which a task belongs, its status, and who is performing it. These classes provide an intermediate format that enables the semantic and executions layers to exchange essential information via the middleware.

In order to ensure the generality of GESI's methods and data structures, we have studied the APIs of two commercial solutions (IBM BPM and SAP Business Process Management) and four open-source solutions (Bonita

BPM, Camunda, Activiti, and JBPM) to ensure that common scenarios such as the one in Fig. 3 could be implemented. Çatal's thesis provides a tentative mapping between the APIs of these six BPM engines and GESI's[3]. The next section highlights how one particular implementation of the middleware was done.

4. Implementation

Our middleware, implemented in Java, provides predefined business logic handling the coordination between the semantic and execution layers (e.g., as in Fig. 3) and the translation and relaying of information. It also provides an implementation of the *Team and Workflow Controller* (TWC) in the semantic layer (see Fig. 2), a Java wrapper interacting with the existing C++-based *Model Interpreter and Instance Base Manipulator* (MIIBM, Fig. 2), which controls the semantic layer. The middleware can handle multiple processes and multiple process instances.

To use this middleware with a particular BPM suite, the parts of GESI's interface involved in interactions with the execution layer need to be implemented. IBM BPM 8.5.5 was used in our first implementation of GESI, called GESI4IBM. The Java API provided by IBM BPM is accessible using RESTful services. The *Uniform Resource Identifiers* (URIs) in these APIs identify RESTful services that access business processes and task data in the engine in order to interoperate with an external system. These methods exploit *JavaScript Object Notation* (JSON) objects used to pass input parameters and get results. For example, the method from IBM BPM's Java API for assigning a task to a role in the process is: `JSONObject client.assignTask(new Integer(task_ID), role);` which maps to the RESTful service `POST /rest/bpm/wle/v1/task?action= {string}[&taskIDs={string}]`. There is hence a need to convert data objects used in GESI's methods and JSON objects understood by the RESTful services. This required some parsing effort as the JSON objects used in the BPM suites are complex, with extensive lists of fields.

The resulting GESI4IBM implementation is an adapter in the middleware that implements GESI's methods with code that invokes methods from IBM BPM's Java interface, and surrounding code that converts the data types used by GESI to those used by the BPM suite's (for method parameters and return values), many much information returned by IBM BPM is in the form of JSON objects. As GESI is using generic types for its fields (e.g., there are many `String` parameters), there are also many simple type conversions involved. For example, IBM BPM is using integers for many of its identifiers, while such identifiers are stored as strings by GESI and the semantic layer.

5. Proof-of-Concept Demonstration and Comparison

A simplified acute stroke management process, derived from NICE guidelines[11], is used here as an illustrative proof-of-concept scenario aiming to test the middleware implementation and suggest the feasibility of the overall approach. This process in interesting in that it covers sequential, alternative, concurrent, and urgent tasks. The process was remodeled in BPMN using IBM BPM, with task annotations for required capabilities and for urgent tasks. These task definitions (with annotations and identifiers) were stored as facts in the semantic layer, according to the IHT ontology, together with the process definitions (and identifiers) to which they belong. The fact base was also initialized with facts capturing a given context in terms of who are the practitioners (with their specialties, capabilities, and availability status), some existing practitioners and patients in process instances, and the correspondence between patient presentations and clinical processes. These are typical instances of the concepts and relationships covered by the ontology of Wilk et al.[17]

Using this initial environment, we tested the following scenario in order to cover our goals and requirements. John Doe and Mary Major are consecutively registered by the HIS, both with an acute stroke presentation. GESI4IBM gets the name and presentation of each patient and creates an instance of a TWC thread with each patient's information. These two patients are then handled in parallel by the system. TWC selects the process to execute and invokes GESI's `startWorkflow` method, which returns the process instance for each patient. TWC invokes the `getTaskList` method of GESI and gets received `taskList` for each `patient`. *t_GCS_Assesment* is the first task of the workflow and it is an urgent task. TWC selects practitioner MK for John's process instance, and the `assignMRP` method is invoked accordingly. GESI4IBM relays this assignment to IBM BPM. TWC detects there is no more available practitioner to assign for patient Mary. The reasoner detects a practitioner (NC) who has the appropriate competency levels, but who is busy with a normal task (*t_imaging_evaluation*) in some other patient's workflow. The semantic layer releases practitioner NC from the normal task and assigns her to Mary's *t_GCS_Assessment* task. GESI4IBM relays these requests to IBM BPM. When practitioner NC finishes the task (e.g., via a user interface provided by IBM BPM), TWC asks for the new task list from GESI and gets the *t_imaging_evaluation* task. TWC selects a practitioner for the new task (with a preference for a practitioner previously known to the patient, if still available), and invokes the `assignTask` method to apply the assignment in IBM BPM. *t_imaging_evaluation* needs a decision from the practitioner, i.e., a choice of treatment in the process. The practitioner collaborates with his/her colleagues while deciding and then TWC gets this choice information by invoking GESI's `getWorkflowInstanceDetails` method and asks for the new task list. This scenario continues and successfully completes.

Several such scenarios were tested (with multiple processes, multiple process instances multiple team members)[3], which improves our confidence in the suitability of the middleware and of the overall approach.

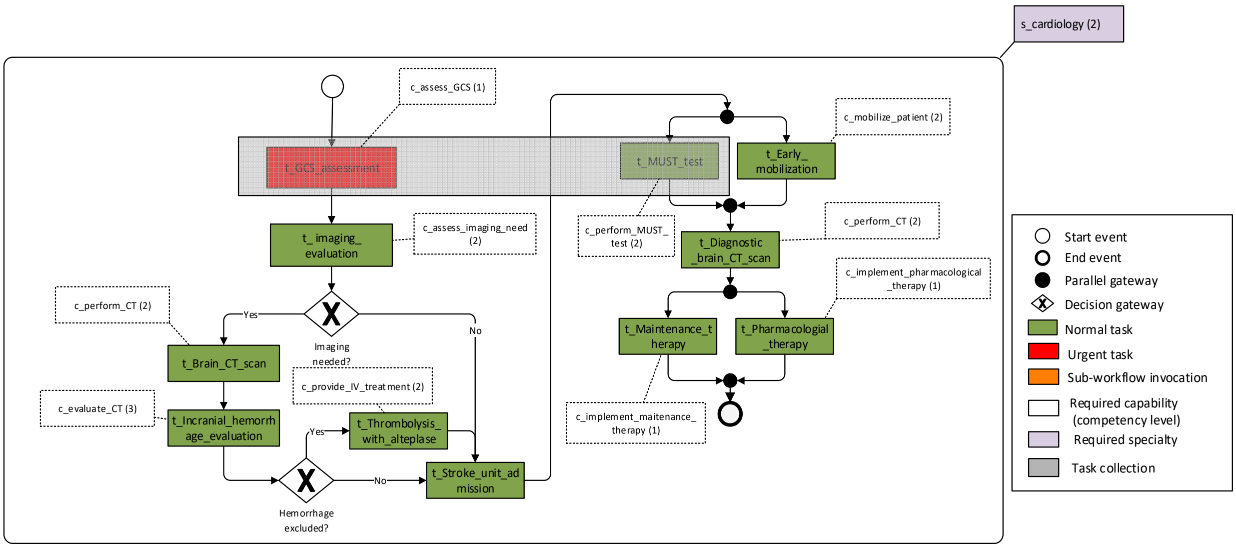

Fig. 4. Simplified acute stroke management process model, with annotations for required capabilities and urgent tasks (GCS= Glasgow Coma Scale; IV=intravenous; MUST= Malnutrition Universal Screening Tool; CT= computerized tomography).

Going back to the goals and existing approaches identified at the beginning of Section 2, Table 1 provides a high-level assessment of our solution compared to closely-related work. The approaches of Cabanillas et al. and Wilk et al. are the ones that score the highest in terms of enabling support of IHT dynamics. Both have a satisfactory support for the first six goals, just like our approach. However, Cabanillas et al. lacks support for the management of frequently changing leaders, exceptions, and collaboration. Wilk et al.'s approach supports leaders, but it only partially supports exception handling and does not support collaboration. Our new approach fills this latter gap as it supports leader selection and automatically inherits collaboration support by integrating with IBM BPM.

Table 1. Comparison of the proposed middleware-based solution with closely-related work.

Approach	**Language Support**	**Language Expressiveness**	**Team Dynamics**	**Capability-Based Alloc.**	**Task/Process Assignments**	**Healthcare Concepts**	**Support for Leaders/MRPs**	**Exceptions**	**Collaboration**
Papapanagiotou et al. (2012)[13]	N	+/-	+/-	N	+/-	Y	Y	Y	N
Schmidt and Kunzmann (2006)[15]	+/-	+/-	Y	Y	Y	Y	Y	N	N
Prinyapol et al. (2009) (DPWFM)[14]	+/-	+/-	+/-	+/-	Y	Y	+/-	N	N
Cabanillas et al. (2015)[2]	Y	Y	Y	Y	Y	Y	N	N	N
Wilk et al. (2016) (MET4) [16,17]	Y	Y	Y	Y	Y	Y	Y	+/-	N
Dermler et al. (2014)[4]	Y	Y	+/-	+/-	Y	N	+/-	+/-	Y
Proposed middleware-based solution	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**+/-**	**Y**

6. Conclusion and Future Work

This paper presented a new middleware and an interface (GESI), which decouple a semantic layer from an underlying execution layer in managing IHT dynamics with a BPM suite. The middleware's logic provides appropriate translations between the concepts, commands, and data structures involved. An implementation of the middleware's interface (GESI4IBM) for the IBM BPM suite, together with the testing of a realistic care process through several scenarios, provide a proof of concept as well as some evidence that the GESI interface is adequate to the task and that the overall approach is feasible, at least from a technical perspective.

There are still many limitations to this study that lead to several future work items. Although we have studied other BPM suites, we still have to support the implementation of other adapters to demonstrate the generality of the middleware. Another threat is the very limited number of processes and scenarios tested so far, and the lack of deployment in a real healthcare environment, with real users and interactions with health information systems. Another research question is whether such process-based approach would scale well for patients with multiple morbidities; knowing where this paradigm breaks is important to better scope the types of care processes that are feasible. Other research concerns include usability, security, privacy, scalability, and adoptability by practitioners.

Acknowledgements

This work was supported by grants from the Mitacs-Accelerate Program (Cluster Internship #IT02932), the NSERC CREATE Program, the NSERC Discovery Grants Program, and the Telfer School of Management Research Fund. We are thankful to IBM Canada for their software license and technical support.

References

1. Andreatta, P. B. (2010). A typology for health care teams. *Health care management review*, 35(4), pp. 345-354.
2. Cabanillas, C., Resinas, M., Mendling, J., and Ruiz-Cortés, A. (2015) Automated Team Selection and Compliance Checking in Business Processes. *2015 International Conference on Software and System Process (ICSSP 2015)*, ACM, pp. 42-51.
3. Çatal, N. (2016). *Supporting Interdisciplinary Healthcare Team Dynamics with Business Process Management*. M.Sc. thesis, Systems Science, University of Ottawa, Canada.
4. Dermler. G., Pfau G., and O'Shea H. *Modeling teams with IBM Process Designer*, Retrieved 15 May 2015, from http://goo.gl/4XdDao.
5. Gatchel, R. J., McGeary, D. D., McGeary, C. A., and Lippe, B. (2014). Interdisciplinary chronic pain management: Past, present, and future. *American Psychologist*, 69(2), pp. 119-130.
6. Grando, A., Peleg, M., Cuggia M., and Glasspool D. (2011). Patterns for collaborative work in health care teams. *Artificial intelligence in medicine*, 53(3), pp. 139-160.
7. Hepp, M., and Roman, D. (2007). An ontology framework for semantic business process management. *Wirtschaftinformatik Proceedings 2007*, AISeL, pp. 27.
8. Hevner, A., March, S., Park, J. and Ram, S. (2004). Design Science Research in Information Systems. *Management Information Systems Quarterly*, 28(1), pp. 75-105.
9. IBM (2013). The Ottawa Hospital Puts Mobile Patients First, Retrieved 29 May 2017, from https://goo.gl/2whUiG
10. IBM (2015). *IBM Business Process Manager*. Retrieved 28 July 2015, from https://goo.gl/XLdbCT
11. NICE (2008). *Stroke and transient ischaemic attack in over 16s: diagnosis and initial management*. Guidelines CG68, UK National Institute for Health and Care Excellence. Retrieved 21 December 2015 from https://www.nice.org.uk/guidance/cg68
12. OMG (2011). *Business Process Model and Notation (BPMN), Version 2.0*. Standard formal/2011-01-03, January 2011.
13. Papapanagiotou, P., Fleuriot, J., and Grando, A. (2012). Rigorous process-based modelling of patterns for collaborative work in healthcare teams. *2012 25th IEEE Int. Symposium on Computer-Based Medical Systems (CBMS).* IEEE CS, pp. 1-6.
14. Prinyapol, N., Fan, J. P., and Lau, S. (2009). A hospital based dynamic platform workflow management. *IAENG International Journal of Computer Science*, 36(2), pp. 192-198
15. Schmidt, A., and Kunzmann, C. (2006). Towards a human resource development ontology for combining competence management and technology-enhanced workplace learning. *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops,* LNCS 4278, Springer Berlin Heidelberg, pp. 1078-1087.
16. Wilk, S., O'Sullivan, D., Kezadri-Hamiaz, M., Kuziemsky, C., Rosu, D., Michalowski W., Fung-Kee-Fung M.: Aligning interdisciplinary healthcare team behavior with workflow execution: an example of a radical prostatectomy workflow. *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2016, pp. 112-117.
17. Wilk, S., Kezadri-Hamiaz, M., Rosu, D., Kuziemsky, C., Michalowski, W., Amyot, D., and Carrier M. (2016). Using semantic components to represent dynamics of an interdisciplinary healthcare team in a multi-agent decision support system. *Journal of Medical Systems*, 40:42, Springer, February 2016, pp. 1-12.