Semantic Model Based Approach for Knowledge Intensive Processes

Madhushi Bandara[1](✉), Fethi A. Rabhi[1], and Rouzbeh Meymandpour[2]

[1] University of New South Wales, Sydney, Australia
{k.bandara,f.rabhi}@unsw.edu.au
[2] Capsifi, Sydney, Australia
rmeymandpour@capsifi.com

Abstract. Many business processes present in modern enterprises are loosely defined, highly interactive, involve frequent human interventions. They are coupled with a multitude of abstract entities defined within an enterprise architecture. Further, they demand agility and responsiveness to address the frequently changing business requirements. Traditional process modelling and knowledge management technologies are not adequate to represent and support those processes. In this paper, we discuss how a process management system based on semantic models can be used to address the needs of non-traditional and knowledge intensive processes. The modelling capabilities of the framework are demonstrated via a case study and evaluated using set requirements that KIP supporting process management system should have. Finally, we discuss how this semantic model based solution can be improved further to cater for the management and execution of knowledge-intensive business processes in a broader context.

Keywords: Knowledge intensive processes · Semantic modelling Ontology

1 Introduction

Knowledge-intensive Processes (KIPs) are processes whose conduct and execution are heavily dependent on knowledge workers performing various interconnected knowledge intensive decision making tasks. KIPs are knowledge, information and data centric in nature and require substantial flexibility at design- and run-time [18]. They have to be understood in the knowledge dimension of the processes and considering the role of human-centred knowledge [9]. To support KIPs the knowledge and collaboration dimensions need to be integrated with the traditional control flow/data dimensions and consider them as a whole by possibly reshaping the process life cycle [11]. The processes such as the diagnostic and treatment process in the medical domain, emergency management process, and artful processes conducted by engineers, researchers or managers can be identified as some examples of KIPs [9].

I. Stamelos et al. (Eds.): SPICE 2018, CCIS 918, pp. 215–229, 2018.
https://doi.org/10.1007/978-3-030-00623-5_15

To address the ad-hoc and frequently changing nature of KIPs, related techniques and tools should support process agility. One obstacle in supporting agile process re-engineering is the gap between organizational level process models and the models built for execution [8,10]. The models built for execution capture the current state of the organizational goals, strategies, and structures, but do not explicitly define them and create the associations between high-level concepts and the execution models. As a result, once the high-level concepts such as strategies and goals change the mapping exercise corresponding to the whole analysis process should be repeated.

In Bandara et al. [3], we proposed a construct to capture processes called "Digital Interaction" (DI), which is defined as part of an enterprise architecture model. It aims to support the dynamic composition of concrete services, a set of interactions and underlying knowledge and information concepts that deliver value to the customer. This composition can capture complex interactions involving humans, events or programming entities such as web services. The basis of the proposed framework is an ontology-based knowledge repository. Embedding DIs in an architectural framework facilitates organizations to manage associations between high level and execution concepts with less effort, as well as to re-engineer and deploy them rapidly in response to business changes.

In this paper, we demonstrate the capabilities of the DI framework for process modeling, and evaluate how such semantic model based process management system can support for KIPs with the aid of the detailed set of requirement defined by Ciccio et al. [9]. We will discuss the background of process modelling approaches followed by a brief introduction to KIP components and requirements in Sect. 2. Section 3 describes the DI framework. In Sect. 4 we demonstrate how DI can be used for KIP modelling, using a case study involving data analytic processes. Section 5 presents the evaluation of the framework and the paper concludes in Sect. 5.

2 Related Work

2.1 Semantic Modeling for KIP

There are two main approaches to process modelling [1] - graphical modes and rule specifications popular in workflow coordination. These modeling approaches limit their focus on specific features or capabilities of a process [13]. Yet the dynamic nature of unconventional business processes is not sufficiently addressed in these approaches [1]. Integrating service-oriented architecture provides a certain flexibility for process modelling and links the execution models to the business level process models. Yet research efforts that focus on the composition of business processes with services such as Cauvet et al. [5] are limited in their contribution to a static description of an executable process.

There are studies that address challenges related to non-traditional business processes such as SmartPM [14] which offers a certain flexibility via run-time adaptation of processes with BPMN 2.0 based modeling schema. ArtiFact-GSM [6] proposes an event-driven, declarative and data-centric approach for business process modelling and highlights the importance of information models as business artifacts to address change management.

Ontologies are proposed for business process management in multiple research works such as Hepp and Roman [12], and Weber et al. [19]. Approaches such as PROMPTUM [7] aim to integrate domain ontologies with business processes to provide semantic quality and traceability between domain knowledge and process models. Rao et al. [16] propose to use ontology-based knowledge maps for process re-engineering, demonstrating the level of traceability achieved by an ontology. Yet they provide limited support for KIP management and can be improved by formalizing knowledge representation around KIPs and linking that knowledge to execution-level process model to provide agility in KIP management and execution.

To address these limitations we employed our experience in studying data analytic process engineering [2,4,15] to design Digital Interaction framework [3] that supports flexible process modeling and management incorporating knowledge representation.

2.2 Components and Requirements for KIP Supporting Systems

Based on scientific literature and real-world application scenarios, Ciccio et al. [9] define a set of formal characteristics of KIPs and six fundamental components (Fig. 1). They define *Data and Knowledge Elements* and the *Knowledge Action* tightly coupled with each other. *Rules and Constraints* define intra- and interdependencies between Data and Knowledge Elements and the Knowledge Action. *Goals* are defined by *Knowledge Workers* and achieved via Knowledge Action. *Environment* is the context of process and impacts all aspects.

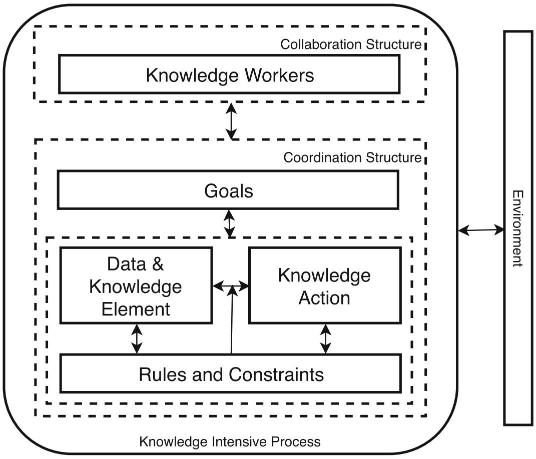

Fig. 1. Components of KIP [9]

Ciccio et al. [9] extend their work by presenting 25 requirements a system should fulfill to support KIPs. The list of requirements is listed in the Table 1. Each requirement is related to one component in the Fig. 1. We employ these components and requirements designed by Ciccio et al. to place the proposed DI framework among existing KIP supporting systems and evaluate its contribution.

3 Digital Interaction Framework

This section provides a brief introduction to the DI framework proposed in [3]. It is based on a construct called "Digital Interaction" defined as a dynamic composition of concrete services, set of interactions and underlying information concepts which can be easily converted into execution level code that deliver value to the stakeholders. This was developed as an extension to the CAPSICUM framework, which is an integrated semantic meta-model for representing different layers of business architecture such as strategies, value streams and high-level processes [17].

The digital interaction construct we propose consists of four parts: service, information, interaction and digital interaction. Main components of the meta-model are Information, Service, Interaction and Digital Interaction, as illustrated by four ovals in Fig. 2. Within each oval, we represent the ontological concepts related to each component and relationships among them. The prefixes "capsi" and "di" are used with concept names to differentiate concepts predefined in CAPSICUM framework and what is proposed in the DI meta-model respectively.

The objective of the information meta-models is facilitating organizations to represent their business objects. The main concept in the information meta-model is di:Information, which is an extension of capsi:Concept. Any information concept related to an organization can be modelled as a subclass of di:Information concept and extended with related properties of type rdf:Property.

The Interaction meta-model captures mechanisms in which inputs or outputs are exchanged between different entities. Some example interactions are messages or events passed within a computer system or human providing inputs through a user interface such as filling a form. Particularly human interactions are frequent and crucial to drive KIPs. By modelling these interactions, we make them flexible, malleable and interpretable. The Interaction meta-model circled in Fig. 2 models the di:Interaction concept as a subclass of capsi:Interaction. It is further extended to three subclasses: form-based, message-based and event-based interactions. Organizations can extend this further to incorporate other interaction types. di:InteractionField is used to represent parameters used or exchanged in an interaction.

Service meta-model is the main building block which links the user-defined interactions and information into actual execution. The service model has to be self-contained so we can create an executable workflow based on it. Our service model is captured by di:Service concept and have parameters named as di:ServiceField to capture concepts used or exchanged in a service.

The process composition is done via DI meta-model, which is an integration of Interaction and Service concepts, linked together via di:FlowLogic and di:ServiceInteractionFieldMapping as shown in Fig. 2. The concept di:ServiceInteractionFieldMapping is used to map inputs from interactions to the service parameters so that a service can be invoked automatically followed by interactions. This composition is created as an instance of the di:DigitalInteraction concept.

The concept di:FlowLogic defines the control flow between different components of the DI. di:FlowLogic is authorized by a service or an interaction which initiates a flow. It contains a set of rules which evaluate a set of InteractionFields or ServiceFields and if they match expected values defined through the information model, respective service, interaction or Digital Interaction is triggered. For example, we can define a Boolean interaction field and create a di:FlowLogic to trigger two services depending on whether the value of the interaction field is true or false.

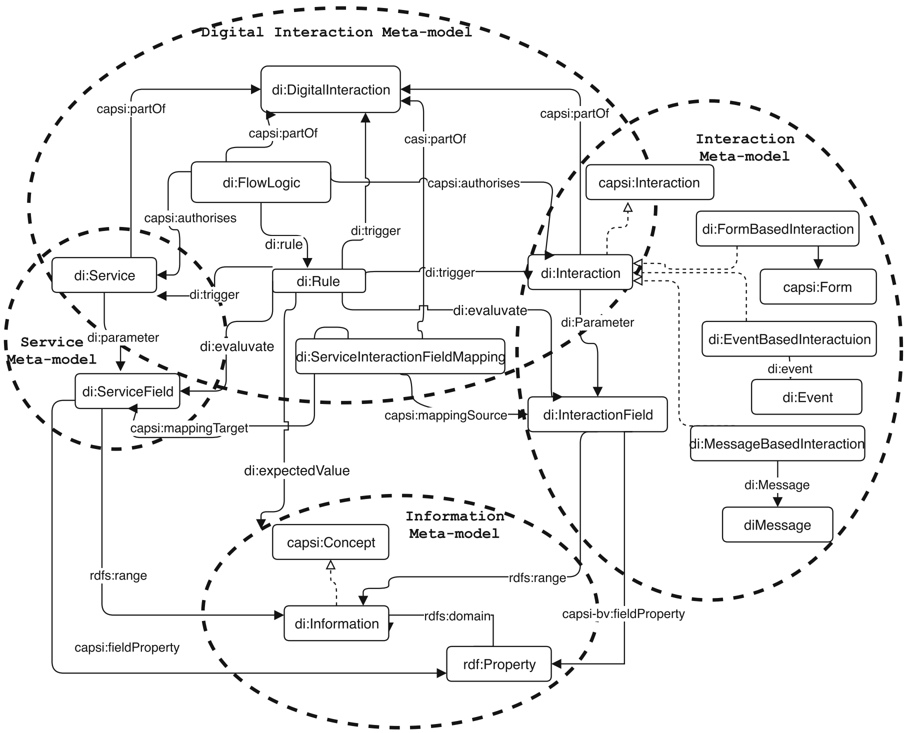

Fig. 2. Main components of the Digital Interaction meta-model with related Information, Service and Interaction model components

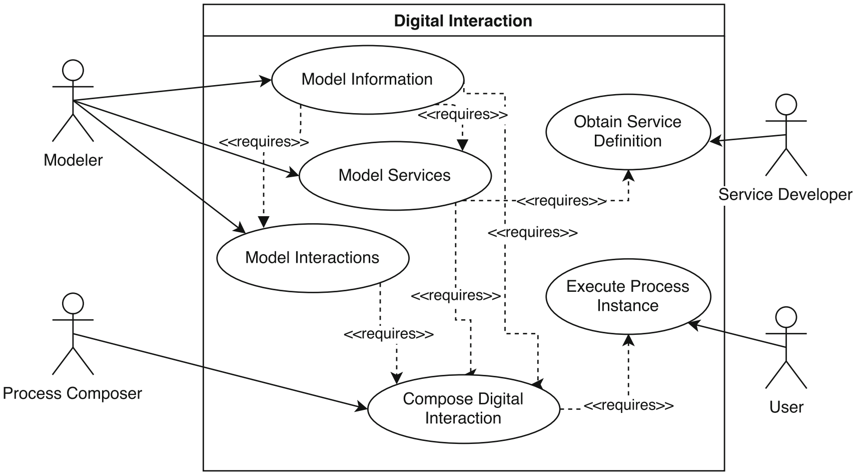

Fig. 3. Use case diagram for Jalapeno-DI extension

The CAPSICUM framework is supported by the Capsifi[1] Jalapeno platform, a cloud-based enterprise architecture modelling platform backed by a triple store for linked data and model management. It allows the definition, analysis, and management of CAPSICUM models as well as exporting the models in machine-readable form (RDF, XSD, JSON) via a GUI. We extended the Capsifi Jalapeno tool and developed Jalapeno-DI extension, a prototype of the DI Framework to demonstrate its capabilities.

Figure 3 shows the use cases that are supported in Jalapeno-DI extension. The first task is the modelling of information, services, and interactions by Modeler. Then the process composer can Compose DI using them. An end user will execute those digital interactions to conduct respective KIPs. Section 4 presents a case study which will elaborate on this further.

4 Case Study

4.1 Dynamic Modelling and Analytics

For our case study, we used a large organization that supports a wide range of data analytics business processes to support the day-to-day decision making. As Fig. 4 illustrates, the organization relies on multiple information repositories arranged into 3 categories: domain-specific knowledge, analytics models and data obtained from different sources. This information changes frequently in response to changes in the external environment and needs to be frequently updated. Within this case study, we select three example KIPs related to predictive analytics as examples to be implemented via the DI framework.

[1] https://www.capsifi.com/.

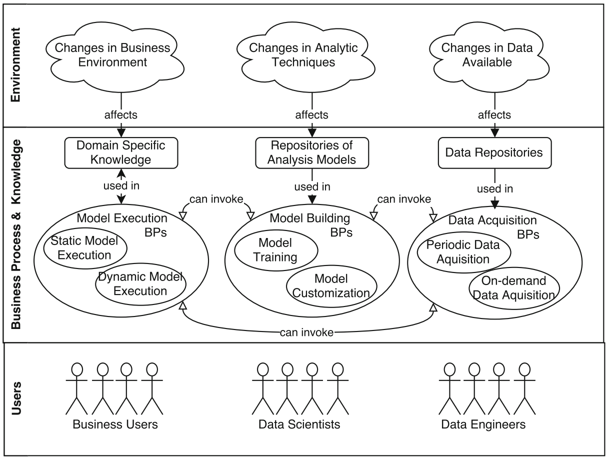

Fig. 4. Overview of case study

Example 1: The first example (DI-1) requires a capability where an analyst can import different datasets, apply predefined prediction models for a specific time period and generate a report. This is realized using three services (REST APIs) that import datasets from given data sources, execute a predictive model and export results. The process and related knowledge are presented in the top rectangle of Fig. 5. The figure illustrates the different stages of this Digital Interaction which uses a mix of form-based interactions and invocation of services.

Example 2: Example 2 (DI-2) is used by analysts to create new prediction models by selecting training and test data and feeding them to an algorithm. Generated prediction models can be included under Model Specification knowledge. The process is captured in the bottom rectangle of Fig. 5.

Example 3: This (DI-3) is an example where agility is needed to respond to changing business requirements. New requirement arises to extend DI-1 and allow users to create new prediction models if a suitable model is not found. This is done via linking Select Prediction Model interaction to DI-2 through two event-based interactions as shown in Fig. 5.

To demonstrate the capability of the proposed framework we modelled the three examples via Jalapeno-DI extension.

4.2 Modelling the Example 1

- **Model Information.** We identified four high-level information concepts related to DI-1: Data Source, Dataset Format, Dataset and Prediction Model and their associated properties. Together they can capture domain knowledge and information sufficient to conduct an analysis. We reused existing ontologies when available. For example, Dataset Format and Dataset concepts are

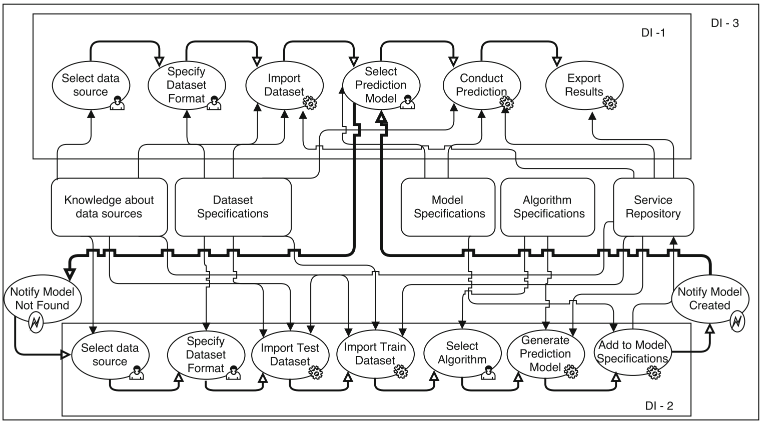

Fig. 5. Example Digital Interaction 3 as a combination of DI-1 and DI-2

adapted from RDF Cube vocabulary[2]. A prediction model was developed using PMML[3] (Predictive Modelling Markup Language) schema.

- **Model Services.** Our example DI-1 implementation leverages three services, modeled within Jalapeno-DI extension. They are 1. Import Dataset 2. Conduct Prediction 3. Export Result.
- **Model Interactions.** We defined 5 interactions: Select data source, Specify dataset format, Import Dataset, Select Prediction Model, Conduct Prediction. Import Dataset and Conduct Prediction interactions provide parameters necessary for respective service executions, while other three aid in the decision making. All interactions are backed by information models to provide suggestions for decision making and to identify parameters user should provide.
- **Compose Digital Interaction.** We model the Digital Interactions, for example, DI-1 that starts with Select data source interaction, followed by Specify dataset format and Import Dataset. Import Dataset interaction triggers the Import Dataset service. Then Select Prediction Model interaction and Conduct Prediction service are linked respectively. Dataset returned from the Import Dataset service is mapped to Execute Model service. Export Results service is triggered immediately after the completion of the Execute Model service to generate a report for the user.

Each link between two components of a DI is captured through di:FlowLogic. To link fields of Interactions with Services, di:ServiceInteraction-FieldMapping concept is used. For example, Data Source returned by Import Dataset service is mapped as the input for Conduct Prediction service.

[2] https://www.w3.org/TR/vocab-data-cube/.
[3] http://dmg.org/pmml/v4-3/GeneralStructure.html.

Once this DI is designed, an execution engine is necessary to create graphical user interfaces from interactions and handle different service calls. Different users can use this DI to conduct individual prediction tasks.

4.3 Modelling the Example 2

We need to define a new DI for the process that creates a new prediction model (DI-2). We reuse information model, Import Dataset service, Select data source, Specify dataset format and Import Dataset interactions. A new information model called Algorithm Specifications is created to capture analytics algorithms such as linear regression. New services were created to Generate Prediction Model and Add them to Model Specifications.

4.4 Modelling the Example Digital Interaction 3

To create DI-3 we extend DI-1. Select Prediction Model interface is updated with a checkbox field (Model Unavailable), which user can tick if a suitable model is not available. Two new event-based interactions are defined, one to capture whether a model is available or not, other to know when a new model is created.

DI-3 is formed by aligning interactions and services as shown in Fig. 6. We included a decision followed by Select Prediction Model interaction to represents a flow-logic for rule-based navigation. This di:FlowLogic instance is defined based on the Model Unavailable field and triggers Notify Model not Found Interaction or Conduct Prediction interaction accordingly. Notify Model Not Found interaction is followed by DI-2, which triggers Notify Model Created interaction at the completion and the DI-1 can continue. Further details of the DI modelling through Capsifi Jalapeno tool can be found at our website[4].

5 Evaluation of Digital Interaction for KIP Support

The proposed DI framework contains all components of a KIP supporting system illustrated in Fig. 1. Environment and Data and Knowledge Elements are captured in the information model. Knowledge Actions are captured in both Services and the interaction models. Rules and constraints are embedded in the di:FlowLogic, di:Rule and di:ServiceInteractionFieldMapping concepts. The process is captured in the Digital Interaction construct. Goals and Knowledge Workers components are not within the scope of the DI framework, but they are handled by CAPSICUM meta-model itself and inertly usable within the DI Framework.

To evaluate how DIs can contribute to uplift KIP management and execution, we selected the set of requirements proposed in literature [9]. Cicco et al. [9] had designed those requirements in order to benchmark the KIP supporting systems in different dimensions. They have published an evaluation of existing KIP

[4] http://adage.cse.unsw.edu.au/Resources/DigitalInteractionCaseStudy.

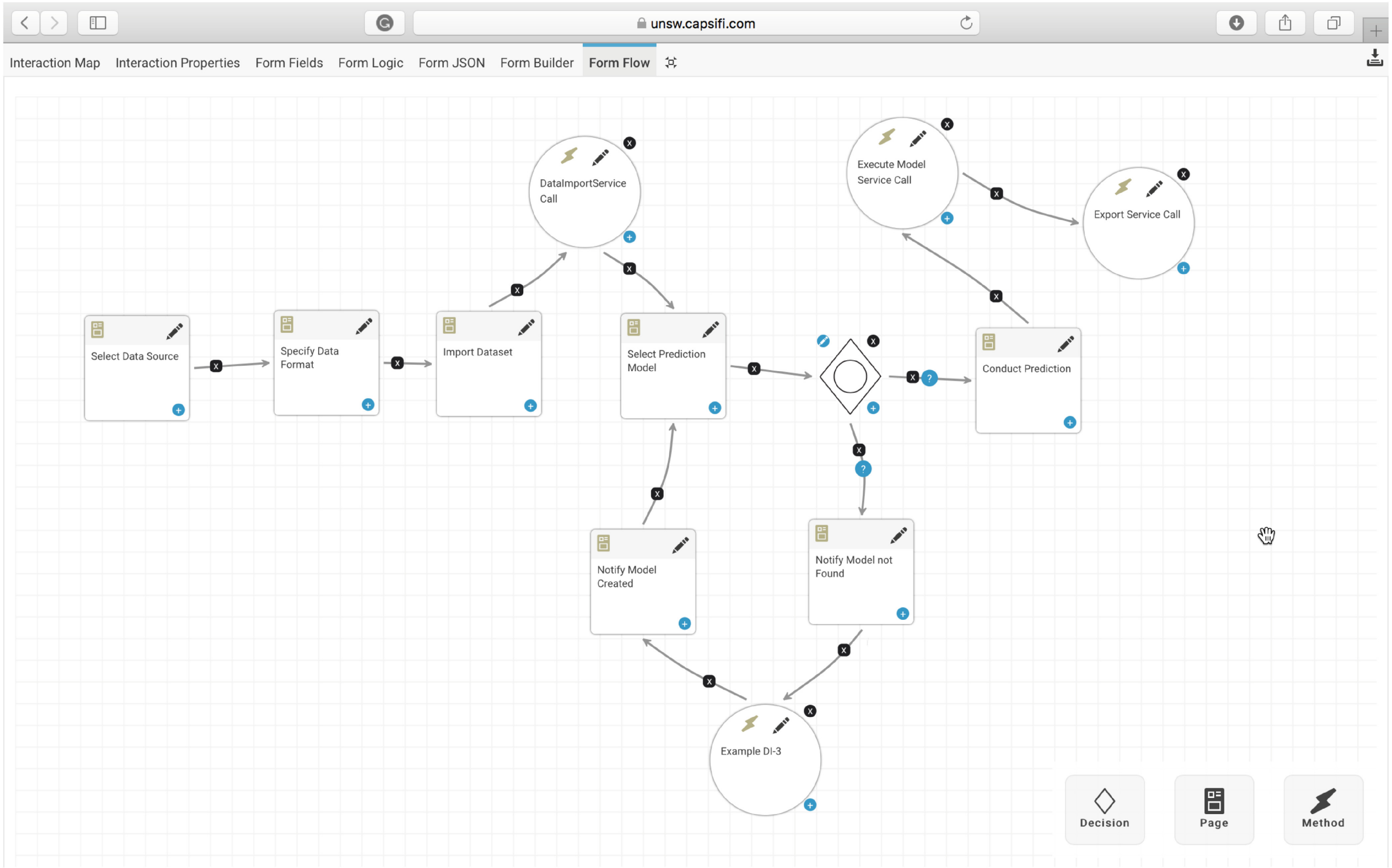

Fig. 6. DI-3 modelled in Jalapeno-DI Extension

Table 1. Compliance of DI to the requirements of KIP Supporting Systems

Component	Requirement	Jalapeno-DI Extension	SmartPM	ArtiFact-GSM
Data	R1 Data modeling	+	+	+
	R2 Late data modeling	+	−	−
	R3 Access to appropriate data	+	−	+
	R4 Synchronized access to shared data	−	−	−
Knowledge actions	R5 Represent data-driven actions	+	~	+
	R6 Late actions modeling	~	~	−
Rules and Constraints	R7 Formalize rules & constraints	+	+	+
	R8 Late constraints formalization	~	−	−
Goals	R9 Goals modeling	Inherent	−	−
	R10 Late goal modeling	~	~	−
Processes	R11 Support for different modeling styles	−	+	+
	R12 Visibility of the process knowledge	+	−	+
	R13 Flexible process execution	+	−	~
	R14 Deal with unanticipated exceptions	−	+	−
	R15 Migration of process instances	+	−	−
	R16 Learning from event logs	−	−	−
	R17 Learning from data sources	+	−	−
Knowledge Workers	R18 Knowledge workers modeling	Inherent	+	+
	R19 Formalize interaction between knowledge workers	Inherent	−	−
	R20 Define knowledge workers' privileges	Inherent	−	+
	R21 Late knowledge workers' model-ing	−	−	−
	R22 Late privileges modeling	−	−	−
	R23 Capture knowledge workers' decisions	+	+	+
Environment	R24 Capture and model external events	+	−	+
	R25 External events late-modeling	−	−	−

supporting systems such as SmartPM and ArtiFact-GSM alongside the requirements. Hence we adapted the same requirement based evaluation procedure to evaluate DI framework and compared the results with what is accomplished in SmartPM and ArtiFact-GSM. Table 1 presents our evaluation results. We use symbols (+), (−) and (∼) to indicate whether each system supports, not support or partially supports the respective requirement. There are some requirements (R9, R18, R19 and R20) marked as *Inherent*, that are not captured by the DI framework, but inherently supported by Capsicum framework.

When looking at the requirements satisfied by the Jalapeno-DI extension, we observed how all data, their properties as well as interrelationships (R1) relevant to a process can be captured through semantic models. Late data modeling (R2) is enabled by linking the information model to the interaction model and enabling new information creation via interactions. Query engine and form-based interaction are used to fetch and show appropriate data (R3). di:InteracionField concept maps the user actions with appropriate information models and enables to constraint actions based on information model (R5). di:Rule and di:FlowLogic closely follows the Event-Condition-Action pattern and enable users to define decision tables (R7). Users can access and visualize all the knowledge captured in the DI meta-model via GUI provided in the Jalapeno tool (R12). With DIs available on a canvas, users have the flexibility to edit and change the process execution (R13). The malleability of semantic models via Digital interaction framework naturally supports the migration of process instances from one information model or set of services to another (R15). The framework provides dynamic support for users based on the historical information and data stored in the information model (R17). All the processes defined via DIs have the capacity to capture knowledge workers' decisions via instantiating the di:FlowLogic (R23). Our information model is flexible and caters for external events coming from the environment and we can associate them with existing components of the information model (R23).

Goal modeling (R9), Knowledge workers and their privilege modeling (R18, R20) as well as defining the interaction between them such as which user/role is responsible for which part of the process is inherently supported by CAPSICUM framework and available for DI framework via capsi:Interaction concept.

We observe that by using a flexible semantic meta-model and supporting dynamic DI composition, the proposed framework fully or partially comply with most requirements supported by SmartPM and ArtiFactGMS with additional requirements such as late data modelling (R2), flexible process execution (R13), learning from data sources (R17) and formalize interaction between knowledge workers (R19). Detailed evaluation of SmartPM and ArtiFactGMS against these requirements can be found at [9].

In terms of limitations, our framework does not support fully for late actions, goal or event modelling, and constraint formalization of the process (R6, R8, R10) at its enactment. User actions at runtime are limited by what is pre-defined at the interaction model. Late knowledge worker modeling (R21) and privilege modeling (R22) are not supported in DI framework. Synchronized access to shared data (R4) and Deal with unanticipated exceptions (R14) are execution platform related requirements our prototype cannot comply with. Currently, we support only a single style of modelling based on actions and hence R11 is not achieved. Yet the semantic models have the capacity to expand and cater to different modelling styles such as data or event centric. Learning from event logs (R16) is another important requirement not supported. Finally, although we are able to capture execution instances of DIs, we do not provide an interface to explore and analyze them (R16).

6 Conclusion and Future Work

In the paper, we proposed a new ontology-based framework called DI to support knowledge-intensive processes management by providing agility and better knowledge representation. The framework is designed as an extension of the CAPSICUM framework [17]. We developed a prototype of the proposed framework on top of Capsifi Jalapeno platform and demonstrate its capabilities via a case study of designing predictive analytic process. The potential of our framework is evaluated through a set of requirements proposed in the literature for KIP support systems.

By implementing DI meta-model on top of CAPSICUM framework we enabled the linking of service concepts, interactions and DIs into the holistic enterprise architecture. We can define DI as a part of a high-level business process or model how different organizational value streams and strategies are linked to these processes. It provides context and traceability for the services, information, and interactions we define. Hence we get one step closer to better alignment between business and IT architecture of the organization.

One limitation of our study is the restriction imposed by extending the CAPSICUM framework. We lose a certain level of flexibility, especially when designing flow logic, as we are building on top of CAPSICUM ontologies. It is a trade-off we made as we believe the value of DI framework is enhanced by extending a framework that is already accepted and used by large-scale organizations.

According to the evaluation, late information modeling - actions, goals, constraints, privileges, actors is a major limitation of our framework. We plan to address in future by extending the architecture to support run-time modeling as well.

To learn from event logs, we plan to extend the information model to capture user insights and performance details for executed DI instances. For example, in the predictive analytics case study, we can extend the Prediction Model concept to record experiences from different users who used a particular model, its performance statistics for different datasets extracted from execution logs etc. Then a user can use that accumulated knowledge in future DI design and execution.

Further, to harness the full potential of DIs we need a good execution platform that can access semantic models and drive different interactions dynamically, fulfilling requirements such as synchronized access to shared data and dealing with unanticipated exceptions.

The main challenge in adapting DIs for an organization is designing a good information model that reflect business objects. This model is unique to an organization and developing it from scratch can be challenging. Our framework is designed to link existing information models (e.g.- RDF Cube to model Dataset used in the case study) easily. Hence designing a repository of abstract information models and guidelines for specific KIP domains such as data analytics, finance or marketing can lift the burden of information modelling and encourage many organizations to adapt DI framework.

We consider this work as a foundation for a new approach to solve challenges related to KIP management and execution using semantic models. As future goals to achieve that objective, we propose to extend DI to contain a knowledge layer that can enable knowledge workers to share their insights and experience, which can supports others in conducting similar KIPs and decision making.

Acknowledgments. We are grateful to Capsifi, especially Dr. Terry Roach, for providing Capsifi Jalapeno platform and sponsoring the research which led to this paper.

References

1. Baghdadi, Y.: Modelling business process with services: towards agile enterprises. Int. J. Bus. Inf. Syst. **15**(4), 410–433 (2014)
2. Bandara, M., Behnaz, A., Rabhi, F.A., Demirors, O.: From requirements to data analytics process: An ontology-based approach. In: Proceedings of the 5th International Workshop on the Interrelations between Requirements Engineering and Business Process Management at BPM 2018. Springer, Cham (2018). (in press)
3. Bandara, M., Rabhi, F.A., Meymandpour, R., Demirors, O.: A digital interaction framework for managing knowledge intensive business processes (manuscript submitted for publication) (2018)
4. Bandara, M., Rabhi, F.A.: Semantic modeling for engineering data analytic solutions. (manuscript submitted for publication) (2018)
5. Cauvet, C., Guzelian, G.: Business process modeling: A service-oriented approach. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, pp. 98–98. IEEE (2008)
6. Cohn, D., Hull, R.: Business artifacts: a data-centric approach to modeling business operations and processes. IEEE Data Eng. Bull. **32**(3), 3–9 (2009)
7. Coşkunçay, A., Gürbüz, Ö., Demirörs, O., Ekinci, E.E.: PROMPTUM toolset: tool support for integrated ontologies and process models. In: Dumas, M., Fantinato, M. (eds.) BPM 2016. LNBIP, vol. 281, pp. 93–105. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58457-7_7
8. Demirors, O., Celik, F.: Process modeling methodologies for improvement and automation. In: IEEE International Conference on Quality and Reliability (ICQR), pp. 312–316. IEEE (2011)
9. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semant. **4**(1), 29–57 (2015)
10. Filiz, C.Y., Demirors, O.: Utilizing process definitions for process automation: a comparative study. In: BPM and Workflow Handbook, Spotlight on Business Intelligence (2010)

11. Gronau, N., Weber, E.: Management of knowledge intensive business processes. In: 2nd International Conference on Business Process Management (2004)
12. Hepp, M., Roman, D.: An ontology framework for semantic business process management. Wirtschaftinformatik Proceedings **2007**, p. 27 (2007)
13. Kumaran, S., Liu, R., Wu, F.Y.: On the duality of information-centric and activity-centric models of business processes. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 32–47. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69534-9_3
14. Marrella, A., Mecella, M., Sardina, S.: SmartPM: an adaptive process management system through situation calculus, indigolog, and classical planning. In: KR (2014)
15. Rabhi, F., Bandara, M., Namvar, A., Demirors, O.: Big data analytics has little to do with analytics. In: Beheshti, A., Hashmi, M., Dong, H., Zhang, W.E. (eds.) ASSRI 2015/2017. LNBIP, vol. 234, pp. 3–17. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76587-7_1
16. Rao, L., Mansingh, G., Osei-Bryson, K.M.: Building ontology based knowledge maps to assist business process re-engineering. Decis. Support Syst. **52**(3), 577–589 (2012)
17. Roach, T.: CAPSICUM - A Semantic Framework for Strategically Aligned Business Architecture. Ph.D. thesis, University of New South Wales (2011)
18. Vaculin, R., Hull, R., Heath, T., Cochran, C., Nigam, A., Sukaviriya, P.: Declarative business artifact centric modeling of decision and knowledge intensive business processes. In: 15th IEEE International Conference on Enterprise Distributed Object Computing (2011)
19. Weber, I., Hoffmann, J., Mendling, J., Nitzsche, J.: Towards a methodology for semantic business process modeling and configuration. In: Di Nitto, E., Ripeanu, M. (eds.) ICSOC 2007. LNCS, vol. 4907, pp. 176–187. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-93851-4_18