

# ENHANCING FLEXIBILITY IN CLINICAL TRIALS USING WORKFLOW MANAGEMENT SYSTEMS

Albert Murungi, Josephine Nabukenya

Information Systems Department, School of Computing and Informatics Technology  
Makerere University Kampala  
Kampala, Uganda

murucumus@gmail.com, josephine@cit.ac.ug

**Abstract .** In the medical environment, clinical research has been predominantly paper-based. The Case Report Form (CRF) is an example of paper-based tools used in documenting clinical trials. A clinical trial can be understood to be a business process. This also means that automating the clinical trial can be referred to as a workflow process. In this research, we seek to automate the clinical trial in order to support the medical practitioners to ably translate their natural language format including CRFs using a workflow system. However we also address the need for flexibility in workflow models in the clinical trials that operate under specific guidelines in form of protocols. To attain the required level of flexibility in clinical trial workflow models, we use the constraint-based modeling language. The results from the validation indicate that the designed workflow modeling framework meets the required level of flexibility with respect to its usefulness and usability.

**Keywords:** workflow modeling, clinical trials, process models, flexibility

## 1 Introduction

The past decade has seen business process modeling become an important means in business process reorganization and management projects [1]. This is because modeling offers a way to capture the implicit process knowledge of an organization and also documents it explicitly in a formal or semi-formal way. However, despite the evolution of business process modeling, highly trained advisors with sufficient domain expertise are still required to perform evaluations of the different business process models [1]. In the same respect, [12] also observes that business process design tasks are still being done by specialized engineers. This alone is a hindrance to the general adoption of business process modeling by other domain knowledge practitioners. This research seeks to resolve the concern of enabling knowledge workers such as medical practitioners to easily translate the volatile and ever dynamic domain specific business processes without entirely changing the existing process and workflow models.

Research on business process application has been targeted at application domain areas such as the banking sector [4], and engineering [12], among others. Substantial research has also been carried out in medical informatics to address some of the business processes within the medical field such as clinical trial (CT) protocol writing [17], computational model-based decision support tools [9, 22], and the use of mark-up languages to transfer existing free text clinical protocols into a computer-interpretable format [20, 19]. CT processes offer a prime area of study for constantly changing and evolving business processes (BP) since they are no exception to unpredicted situations. However, amongst all these processes, flexibility was key aspect and this research has looked at ways of enhancing the same within key business processes. Process flexibility needs to be incorporated into the business process modeling efforts for the CTs mainly for two reasons; first, to enhance exception handling as well as take care of both planned and unforeseen changes. Secondly, to continually identify process related tasks for the respective business so as to identify improvement gaps. Through process flexibility, we steadily aim to ensure that the designed workflow models achieve both stability and enhance efficiency in clinical practice.

Whereas process flexibility can be interpreted as the ability to deal with both foreseen and unforeseen changes, by varying or adapting specific parts of the business process that are affected by them, whilst retaining the essential format of those parts that are not impacted by the variations [15], in this research context flexibility is as much about what should stay the same in a process as what should be allowed to change [21].

To this end, this research seeks to address how best to enhance flexibility in workflow models using an example of CTs. The perceived benefits of this transition include reduced cost and time, enhancement of flexibility, as well as improved support for design-time and run-time process modeling activities in workflow modeling and particularly in CTs. We comprehensively discuss the concept of flexibility in relation to the CT process in section 2. To address our research question, we first identified the process-related tasks in CTs, from which we derived the requirements that were finally translated into workflow models as presented and discussed in sections 3, 4 and 5, respectively. Section 6 provides the case study conducted to test the proposed guidelines, including validating the usability and usefulness of the designed models. Finally we provide directions to future research in section 7.

## **2 CLINICAL TRIALS AND WORKFLOW MODELING**

Amongst the major areas of conduct within CTs is the collection of clinical data from study subjects by physicians, nurses, and any other appropriately trained medical personnel. [12, 8] also emphasizes the fact that, “one of the basic goals of CTs is to obtain data for the safety and/or effectiveness of a drug or a device in order to get that drug or device to market”.

To achieve the above, suitable workflow languages need to be used in designing a workflow system that can be used to address the domain specific tasks. For example [5] have used Business Process Modeling Notation (BPMN) to model some of the CT

phases as models. Furthermore, there is a number of existing approaches to clinical guidelines modeling in form of computer interpretable guidelines (CIGs) such as GuideLine Interchange Format (GLIF), EON, PRODIGY, PROforma amongst others [25, 22, 26]. Despite the benefits of these languages including reduced time and cost of service delivery, they are still unable to address key concerns such as flexibility in CT BPs within the area of healthcare provision. [23] observes that using a model-driven approach facilitates a high degree of flexibility whereby process models can be adapted to fulfill new requirements. The modified process models can then be used to enact the emergent BPs.

## **2.1 Flexibility within Clinical Trial process Models**

Flexibility – involves dealing with both foreseen as well as unforeseen changes to processes and the ability to respond to these changes without necessarily having to change the entire process model [11]. Flexibility is observed as what stays the same not what changes. Indeed, a process can only be considered to be flexible if it is possible to change it without needing to replace it completely [19]. Hence flexibility is effectively a balance between change and stability that ensures that the identity of the process is retained [11]. In the context of this research flexibility is considered as the ability to respond to emerging changes without necessarily changing an entire process model with the aim of ensuring stable process models despite the dynamics in the operating environment.

Generally, we observe that flexibility is significantly critical because the health domain and specifically CTs do operate in a fluid environment although still being guided by well defined protocols. There are a number of foreseen as well as unforeseen changes within the CT processes. This would necessitate the need to vary or adapt those parts of the BPs that are affected by the changes, whilst retaining the essential format of those parts that are not impacted by the variations.

Up to now, there have not been any broadly adopted proposals or standards offering guidance for developing flexible process models able to deal with the different changes experienced within processes. Instead most standards focus on a particular notation (such as XPDL, BPEL, BPMN, among others) and these notations typically abstract from flexibility issues [22]. It is imperative to note that the approaches and mechanisms of implementing flexibility can significantly impact the way workflows are designed. Also, the complexities involved in having to explicitly specify how processes flow and interact with each other needs to be minimized, and hence making it much simpler to design and enact workflow models. Therefore, we aim to simplify the way modeling is done especially by the knowledge workers with limited expertise in working with complicated modeling languages taking into consideration the need to address the identified flexibility concerns.

## **2.2 Enhancing Flexibility in Workflow Modeling using the Declarative Language**

Workflow modeling mainly follows two categories of languages: procedural and constraint-based modeling languages respectively. The traditional approaches tend to use procedural process models (also known as classical modeling languages) to ex-

implicitly specify the execution procedure. Examples are BPEL, YAWL, UML, amongst others [28, 29, 39]. On the other hand, constraint-based models implicitly specify the execution procedure by means of constraints, that is, any execution that does not violate constraints is possible [38, 20]. Examples among others include Logical Temporal Language (LTL), and ConDec [30, 1, 31]. Constraint-based models are considered to be more flexible than traditional models because of their semantics [20, 32]. As such, a new paradigm shift towards a declarative approach introduces declarative models that specify what should be done without specifying how it should be done [19, 31, 1]. The declarative approach is based on constraints, that is to say, anything is possible as long as it is not explicitly forbidden. The Declare framework [33] that provides for multiple constraint-based languages is currently implemented and has been adopted for use in this research. Declare is extendible and without any programming it can be configured to support additional constraint-based languages. Furthermore, instead of explicitly defining the ordering of activities in models, Declare models rely on constraints to implicitly determine the possible ordering of activities, implying that any order that does not violate constraints is allowed [33, 38]. By using such a declarative approach, different types of flexibility become possible [28, 33]. The core of the declare workflow management system consists of the following basic components: Designer, Framework and Work list. The Designer component is used for creating the so-called constraint templates, to design concrete process models, and to verify these models. The Framework enacts instances of process models. Moreover, it also conducts ad-hoc changes of running instances. While the Framework centrally manages the execution of all instances, each user uses his/her Work list component to access active instances.

From the preceding discussion, we observe that the declarative modeling approach presents the opportunity to address the flexibility concerns. The underlying constraints provide the general boundaries around which work is executed. Furthermore, the defined constraints can be declared either as mandatory or optional implying that if an activity breaches a mandatory constraint then all subsequent activities in the workflow cannot be executed. For example, in a CT, assuming the protocol states a specific exclusion criterion for participants, some of the conditions within the exclusion criteria could be tagged as either mandatory or optional. Therefore, for as long as a person presents any of the conditions that fall within the mandatory ones, they are automatically excluded from participating in the study. With a declarative modeling approach, this would be fairly simple to achieve as compared to when using a procedural approach that might cause over specification before achieving a similar result.

### **3 Research Approach**

In order to design the framework for BPs workflow modeling, the design science (DS) method was followed. Design Science (DS) aims at producing artefacts that contribute to the body of knowledge and are relevant to the community [37, 40]. Design science also aims at developing methods and an instance of a system for a given set of user requirements represented as models [40]. Design science therefore seeks to develop new ideas, practices, technical capabilities and products that will enable users to

analyse, design, implement and use information systems effectively and efficiently. It also aims at developing methods and an instance of a system for a given set of user requirements represented as models. Such methods are known as theories that prescribe how to effectively develop an information system [40]. This method consists of three cycles namely, the relevance cycle, the design cycle and the rigor cycle [13, 14, 15, 40].

The ‘Relevance Cycle’ deals with identification a problem or an opportunity in a given application domain. The business environment is explored to determine the business needs, an input to the design cycle, as well as to define an acceptance criteria for testing produced artefacts [40, 41]. In the relevance cycle, we adopted the CT environment and identified processes that can be used to model workflows while factoring in the aspects of both flexibility and support. The designed models were again put back into the CT environment and subjected to test usability and usefulness within the CT community.

The ‘Rigor Cycle’, involves thorough review of past knowledge in terms of foundations and methodologies to identify applicable knowledge; knowledge that can be applied in solving a given problem. It also ensures that the artefacts produced by the research are innovative and add to the existing knowledge base. The Rigor cycle also provides input to the design cycle in terms of appropriate theories and methods for construction and evaluation of artefacts [41, 40]. The rigor cycle involved an extensive literature review on the current states of practice in the CT environment and its relationship with flexibility in workflow modeling.

The ‘Design Cycle’ is the core of design science. It involves the building and evaluating of artefacts following a set of guidelines defined by Hevner *et al.* [40]. The artefacts produced by design science are: constructs: which are concepts or a ‘language’ (e.g. modelling primitives used in modelling tools) that are used to define and communicate problems and solutions; models: these are descriptions of a real world problem and possible solutions using constructs e.g. process models; methods: these define ‘processes’ that is ‘ways’ of performing a set of activities to achieve a given goal; and instantiations: these are working systems that show that the constructs, models or methods can actually be implemented [40, 37]. In the design cycle, evaluations were carried out using a suitable mix of metrics related to flexibility. We also identified modeling requirements specific to clinical trials and then used these requirements to inform the design of suitable workflow modelling guidelines as well as a modelling framework. We chose the DS method because it provides a well-defined approach to establish ways of enhancing flexibility within workflow models through a structured and iterative process that is self checking through the design cycle. Additionally DS has been successfully used in other related researches [21, 24].

A workflow modeling framework in form of a set of guidelines was designed to guide the modeling process. A case study was conducted to test the proposed guidelines, in addition to validating the designed workflow models using a number of personnel involved within CTs including doctors, data clerks, and a data manager. The sampled category was representative of the key functional roles within the CT process and hence had expert knowledge on the processes they are involved with during the CT. The participants were chosen randomly except for cases where there was only a single person in the role and a total of 15 participants were identified for the study.

## 4 Exploratory Study on CT Business Process Environment

In order to design our workflow modeling framework, an exploratory study was conducted in the CT application domain using a Ugandan Health Institution as the case study. A random sample of 15 respondents comprising of different categories of knowledge workers such as data technicians, medical workers, and nurses among others were contacted with the intention of identifying the process related tasks involved within CTs, the nature of the BPs and how that impacted the way these knowledge workers conducted their work, the challenges faced by the current BPs and how some of the identified challenges could be improved. Questionnaires were applied and responses collected, analysed and presented. Below we discuss our findings and derived requirements:

- Identification of process related tasks in CTs – it was observed that CTs do involve a number of different BPs, tasks and activities with some occurring more frequently than others. Some of the tasks identified included data collection, data analysis, diagnosis, prescription, referrals, and patient selection, amongst others. However, the occurrence of these process-related tasks is entirely dependent on the nature of the CT being conducted. The data management process was one of the commonly occurring BPs. The BPs are composed of tasks and the conditions that determine the order of these tasks.
- Nature of BPs within CTs – the case study results did indicate a high percentage of manual tasks (73%) as compared to the combined automated tasks (27%) within the CTs. This means that commensurately, the BPs associated with these tasks were also of a manual nature. This alone is a challenge because of the associated demerits that come along with having manual processes related to more time, cost and effort required to achieve business goals.

### 4.1 Challenges faced during CTs process related tasks execution:

- i. Flexibility of performing tasks within BPs – there exists a very low level of flexibility (47%) in performing tasks within CT BPs, as compared to 20% of respondents who felt that the BPs were flexible. However, 33% of the respondents were not sure about the flexibility levels within the BPs.
- ii. Ease of designing processes for CTs – it was observed that it was not easy (57%) for knowledge workers to design new BPs as compared to 10% of ease.
- iii. Support within clinical trial BPs – the study revealed a low level (40%) of support within CTs. Based on a high-medium-low index; the results indicated a 33%-27%-40% score respectively. The respondents did attest that there is limited support both at build time and runtime offered to the users. Within the current CT BPs there is limited support for verification, performance analysis, and monitoring of the BPs; therefore it is not easy to enforce correct execution of activities, learn from processes, monitor process instances, and enforce correct changes.
- iv. Reusability of current business processes – the findings also show that reusability of BPs within CTs was low (27%). However 40% indicated otherwise, and 33% were not sure. This is partly explained by the fact that the BPs and their related tasks are mainly manual, inflexible, and with limited support as already observed

above. These factors compounded together make it extremely hard for the knowledge workers to reuse some of the existing BPs.

- v. Ease of modeling workflows for current BPs – the results also reflected a high percentage (67%) for inability to easily model workflows in CTs. 27% were not sure, while only 6% did confer to ease of modeling workflows for BPs within CTs.

The results above do provide an insight into the existent problem within CTs. The results do also reveal that it is not easy for knowledge workers to design workflow models for their BPs. In the following section we discuss the requirements used in designing the workflow modeling framework. The requirements were derived from the preceding challenges.

#### **4.2 Requirements for the Proposed Workflow Modeling Framework**

Based on the findings of the exploratory study, we derived four key requirements for the workflow modeling framework. The requirements identified were primarily as a direct result of the responses received pertaining to the challenges faced during execution of CT process related tasks. The requirements include the following:

- Flexibility enhancement - the ability for knowledge workers to easily make changes without having to entirely change the parts of the BPs that are not impacted by these changes.
- Support features provision - the ability to provide sufficient support tools and mechanisms to detect and mitigate issues that affect performance of workflow models.
- Reusability of process instances - the ability for knowledge workers to make use of existing CT process instances without need to completely redesign them.
- Interpretability of workflow models by workflow engines - the ability of workflow management systems to correctly execute the designed workflow models.

### **5 Designing of the Workflow Modeling Framework**

A workflow modeling framework is an essential aspect of this research because it attempts to address the challenges that were identified in the exploratory study discussed in section 4 above. The framework offers a graphical schematic in form of a flow chart that represents the steps required to be followed by knowledge workers during workflow modeling. With the four key requirements identified above, the framework offers an integrated but easily usable approach for workflow modeling. The framework (figure 1) consists of ten key steps that need to be performed by knowledge workers in order to design their domain specific workflow models. The workflow modeling framework hence offers a graphical guideline to workflow modeling that can be used by these knowledge workers. The framework provides both a set of guidelines as well as a way of achieving flexibility within designed workflow models. The framework empowers knowledge workers to easily model their business

processes based on the provided guidelines. By following the steps (1-10) in the framework in figure 1 and using a modeling tool such as Declare, fragments of flexibility are used to fill the pockets of flexibility [36]. Pockets of flexibility can be defined as the different process fragments that would ideally constitute a workflow but are not executed until runtime when required. A special workflow activity called the build activity provides the rules for concretizing the pocket with a valid composition of workflow fragments [36]. The build activity is embedded within the Declare tool and is basically an abstraction layer of meta-models composed of various rules for selecting flexibility fragments from amongst the available options. It is also this meta-modeling concept that ensures reusability of the workflow models. The concept of meta-models is discussed in more detail by [3, 34, 35] amongst others. The steps in the framework (see figure 1) are explained in more detail here below;

Step 1: Protocol Analysis to identify BPs – Clinical trials usually follow a protocol that defines what needs to be done in order to achieve the goals of the study. Through analysis of the protocols, the different business processes can be identified.

Step 2: Identify the tasks/activities associated with each of the processes – There are some logically related activities associated with each of the identified processes, together with the conditions that guide their order of execution. The conditions form the constraints amongst the tasks. Once the processes have been identified, task identification for each of the processes drills a level deeper into the process to identify what activities constitute the process.

Step 3: Choose a modeling language, for example ConDec, DecSer Flow – Using the declare tool, a modeling language of choice is selected and used to design the workflow model. The modeling is done using the graphical user interface (GUI) of the declare tool. The declare GUI provides an abstraction to the complex LTL which simplifies the task of both process modelling as well workflow modelling. This will eventually be used to translate the natural language text used within the protocols into automated workflow models.

Step 4: Specify the process model using the language – The process model will provide a representation of a collection of logically related activities that are performed to achieve a given clinical trial process outcome such as data analysis, or data verification, etc. The chosen modeling language is used to specify both the process and workflow models.

Step 5: Define the constraints – As discussed earlier in section 2, the nature of declarative modelling is such that any activity can be performed as long as it does not violate the set constraints. Mandatory or optional constraints can be defined to ensure that we are able to achieve the business requirements. *Mandatory constraints* are constraints that cannot be violated and provide a boundary within which the model operates. In the context of this research the constraints can also include semantical constraints that are guided by expert advice from the medical practitioners. These may



also include certain government policies and regulations as well as other regulatory requirements. *Optional constraints* are constraints that will only provide a warning if violated but will not stop the model from executing. In the context of this research optional constraints may include for example using a different substitute drug from the prescribed one in cases of drug shortages.

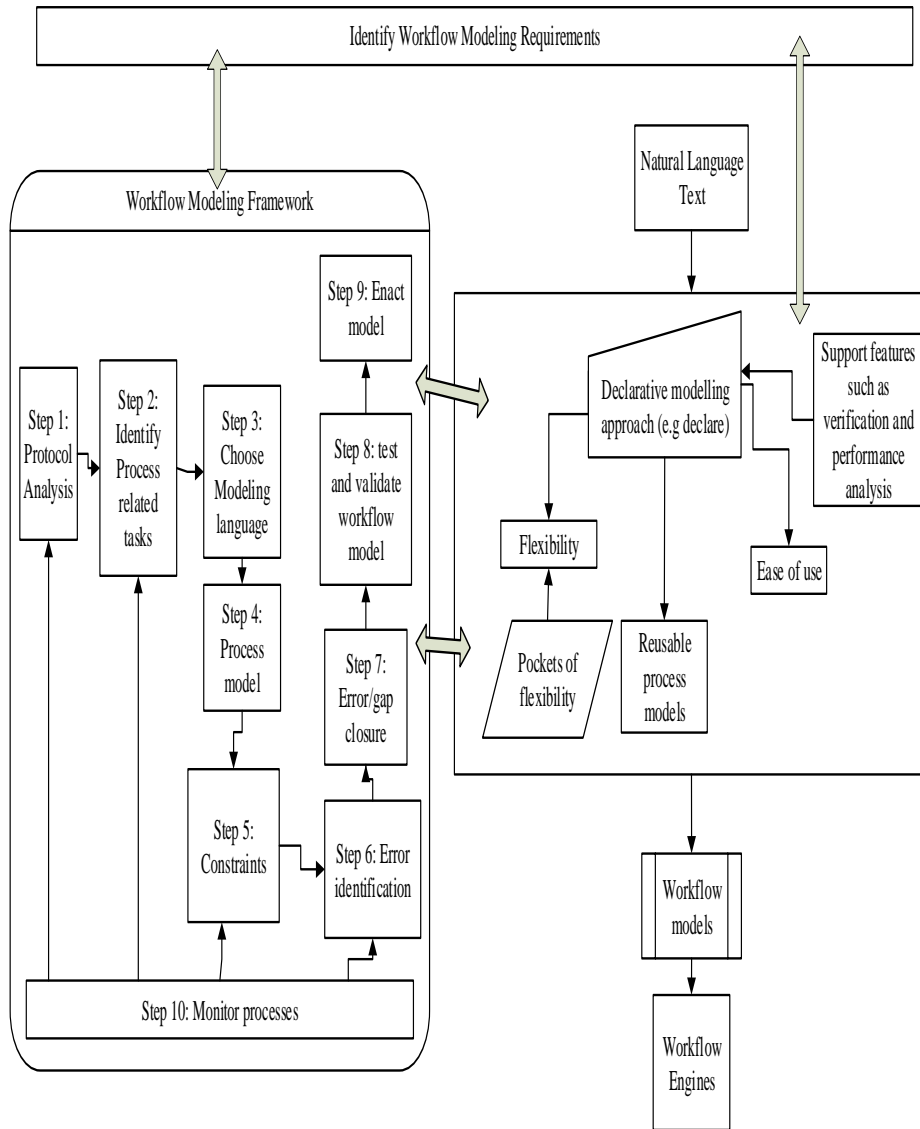
Step 6: Error Detection, for example performs semantical verification or syntactical verification – The declare tool has got embedded features that can be used to perform model verification. During the verification, we can determine whether the designed model has errors with the defined constraints or the control flow of the model.

Step 7: Address any identified gaps - Error correction – In order to correct the identified gaps within designed models and basing on error messages displayed by the declare tool, rightful steps are taken to resolve these gaps and errors either by amending the applied constraints or re-modeling the entire process again. Error correction is necessary to ensure that the designed workflow models achieve their intended purpose.

Step 8: Test and validate process model – Select suitable metrics to validate the designed workflow model. Also, validation criteria can be agreed upon from which different metrics can be identified from time to time. Validation ensures that the designed workflow model is fit for purpose and can be executed by the workflow engines.

Step 9: Enact the model – The designed model at this stage is ready to be executed. This can also be referred to as run time phase of the workflow model. At runtime the designed models should perform the work that they are designed to do. Runtime is achieved by executing the loaded workflow models using the Work list component of the declare tool. Each medical practitioner can access the work list portal using their own individual credentials for purposes of privacy. Also, based on the assigned permissions, access to different workflow models can be restricted.

Step 10: Monitor processes – This is an ongoing process that constantly checks the processes to ensure that they are still relevant and accurate. It is during this step that we monitor to see if changes can be introduced into the existing models and the impact of these changes to the models. The monitoring process may also include reviewing of process logs which can subsequently aid process mining using process mining tools.



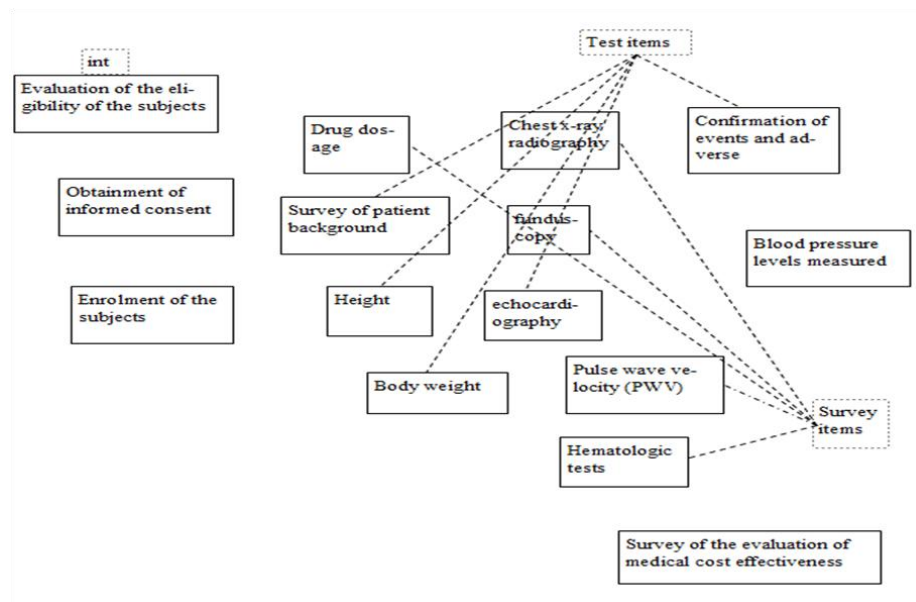
**Fig. 1.** Workflow modeling framework

### 5.1 Case study: Modeling the CT Natural Language Text to a workflow model

In figure 2, we demonstrate a piece of natural language text modelled to a workflow model of the same process based on the hematologic tests activity of a CT process. The hematologic tests activity is composed of several activities that are conducted during the execution of the CTs. Some of the activities within this extract are made of sub-processes where the actual execution occurs. The process includes activities to

determine eligibility into the CT, acquiring consent, enrolment of subjects, study drug dosage information, survey activities about the patients including height, weight, blood pressure levels, x-ray results, as well as test results including urinalysis, and blood tests amongst others.

Using the graphical user interface of the Declare tool, the activities mentioned above are modeled and subsequently both mandatory and optional constraints are applied to the resulting model. After modeling the process, it then becomes the core process (open instance). All subsequent process instances are then based on this core process model. Only the process fragments required for a given process instance are activated at runtime. As these fragments are added, they form pockets of flexibility and it is through these that flexibility is achieved. However, because of the nature of constraint-based modeling, all activities are modeled in no specific order and it is the constraints that determine the ordering of activities. This implies that it is possible to run the activities in any order as long as the constraints are not violated and it is for this reason that modeled activities appear to be simply scattered on the modeling interface in no particular order. We verify the designed workflow model using the in-built feature within the Declare tool. In case of errors in the designed model, these are flagged by the verification module within the Declare tool.



**Fig. 2.** A sample workflow model based on the hematologic tests process of CT process

## 5.2 Demonstrating flexibility in a workflow model using the declarative modeling approach

Core Process – Open instance of a sample CT procedure; the core process can also be called an open instance because it represents a generic process model from which multiple instances can be derived from. It consists of pre-defined workflow activities and pockets of flexibility within the process. The pockets of flexibility have an associated set of workflow fragments (a workflow pack may consist of a single activity or a sub-process) and a special workflow activity called the build activity that provides the rules for concretizing the pocket with a valid composition of workflow fragments.

To demonstrate flexibility in the workflow models, a generic diagnostic process model derived from the ‘hematologic tests’ activity is presented and thereafter used to guide the modeling of three other subsequent process instances. The tasks in the generic process included receive patient, create file, retrieve file, examine patient, build activities (blood test, ultra sound, and second opinion activities), make diagnosis, and receive payment was used.

In each of the derived process instances, different pockets of flexibility are introduced to illustrate flexibility in the process, that is, the degree of stability in the process even as changes are introduced. We observed that a certain number of the core process activities remained stable even as the instances changed (see fig. 4, 5 and 6). The specification of instances initially consists of a copy of the core process (see figure 3). As an instance proceeds with execution, the build activities provide the means of customizing the core process for that particular instance.

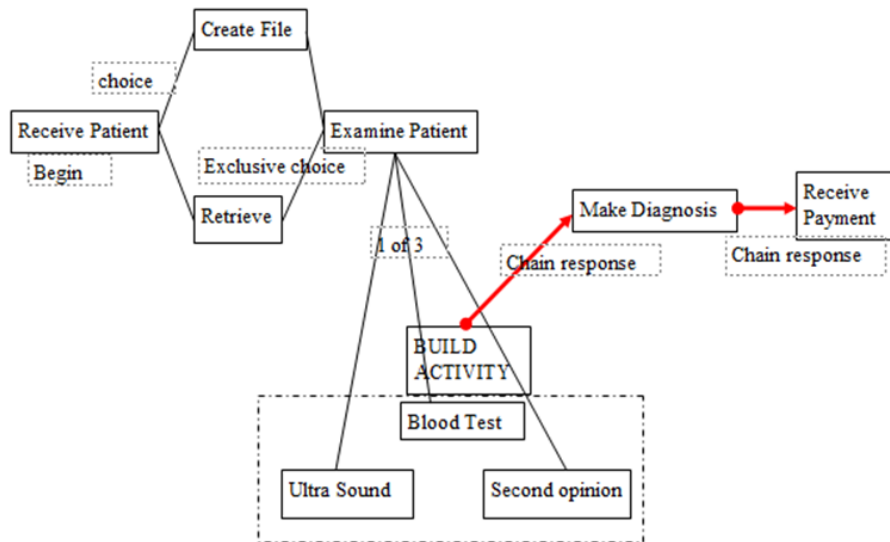
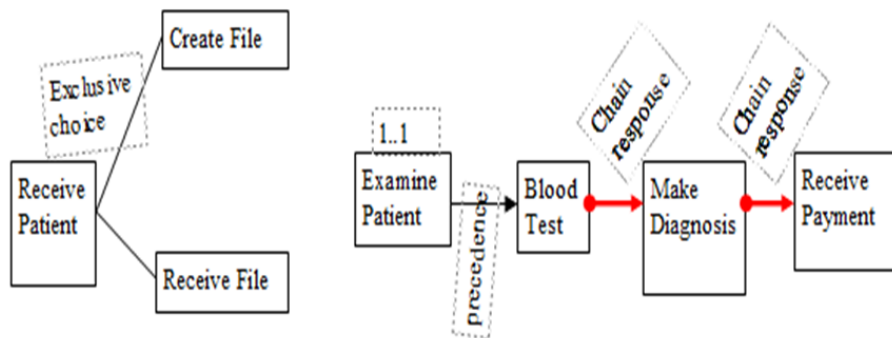


Fig. 3. Open instance of the diagnosis process

In figure 4, we illustrate an example of a flexible workflow representing a typical diagnostic process for investigation of HIV. A new patient coming into the health centre will first be entered into the system, or the file of an old patient will be retrieved. All patients then consult with an attending physician, who will determine what tests need to be performed, on a case-by-case basis. This forms the pocket of flexibility for the process. After the tests, the patient is called again by the attending physician who explains the results of the tests and makes a diagnosis. The patient is then required to report to accounts to make the required payments before leaving the health centre.

Process instance 1 - in figure 4 for instance, the ‘chain response’, ‘precedence’, and ‘choice’ constraints help coordinate the execution of the tasks in that process instance at runtime. Also, in this instance six tasks (‘receive patient’, ‘create file’, ‘retrieve file’, ‘examine patient’, ‘make diagnosis’, and ‘receive payment’) remain unchanged and it is only the ‘blood test’ task that is activated from the available build activities. Nevertheless, given the fact that template building is progressive, the core process may contain several pockets of flexibility which will be concretized through the associated build activities as they are encountered at runtime. As such, the template remains open until the last build activity has completed execution.



**Fig. 4.** Process Instance 1

Process Instance 2 - In fig. 5 only the build activities required for this particular instance are selected and executed at run time excluding any other process fragments that are not relevant and thus ensuring stability of the core process model. In this instance, only activities ‘ultrasound’ and ‘second opinion’ are activated within the build activity amongst all the other available options. Also, in this instance the ‘succession’ constraint introduced between the two active build activities will determine the order in which they are executed. Activity ‘second opinion’ will only happen after activity ‘ultra sound’ has occurred. There is therefore no need to remodel the entire process

but rather to only include the process fragment relevant for a specific instance as indicated below. The build activity is critical since it provides a set of fragments to choose from either automatically or manually.

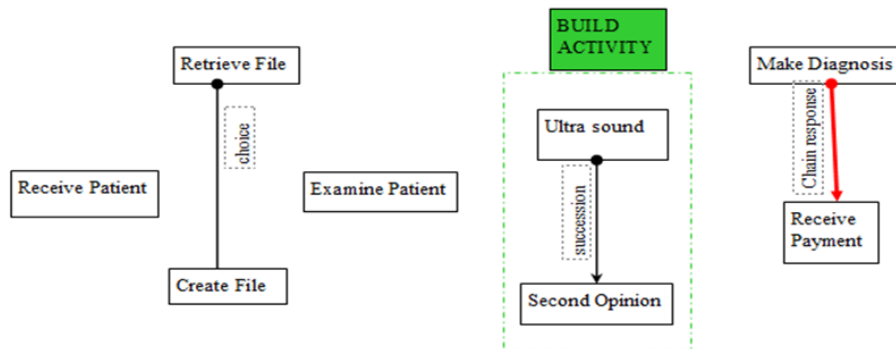


Fig. 5. Process Instance 2

Process Instance 3 – In this instance, new activities including ‘blood test for typhoid’, ‘HIV sero-status’ and ‘BS for MPS (Malaria parasites)’ are introduced into the process and the subsequent process model is modeled into a workflow. After the ‘examine patient’ activity, the multiple choice constraint enables the selection of at least one of the activities from the pockets of flexibility indicated by the dotted boundary in figure 7.

From fig. 4, 5, and 6, we demonstrate how flexibility is achieved in the designed workflow models through the use of pockets of flexibility composed of flexibility fragments. The core process of the workflow model later (fig. 3) forms an instance template from which the subsequent process instances are derived.

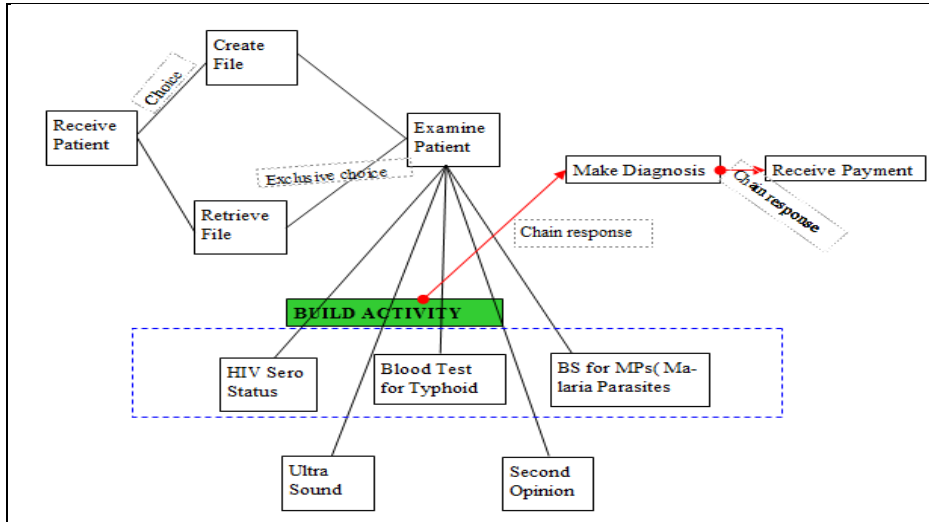


Fig. 6. Process Instance 3

## 6 Framework Validation

The framework validation provided us with a formal procedure to determine whether the designed framework could be applied by knowledge practitioners, specifically medical practitioners involved with CTs.

### 6.1 Validation criteria

To validate the designed framework, we developed three broad criteria (usability, usefulness and flexibility) based on the requirements derived in section 5 and also following the DS method from [9] as summarized below.

1. Flexibility – the ability for knowledge workers to easily make changes without having to entirely change the parts of the BPs that are not impacted by these changes. This was measured using stability and efficiency of the workflow model designed following the framework guidelines.
  - i). stability – the ability of the workflow model to appropriately adapt to the changing business processes (the variations in process instances). It was measured using the number of tasks remaining unchanged in the different process instances, and the number of errors observed upon introduction of changes.
  - ii). efficiency – the degree to which the time and effort resources are saved during workflow modeling. This was measured in terms of; the time to design workflows and correct errors, and the level of output produced during the modeling process.

2. Usability – the ability for knowledge workers to model following the framework guidelines with limited cognitive effort. Usability was measured using: i). learnability - the ability to follow a given number of steps in the framework to model a workflow; ii). satisfaction - the ability to arouse happiness in achieving the goal for which the workflow model was intended for; iii). interactivity - the relationships and inter relations amongst the steps in the guidelines; and iv). understandability - the ability for knowledge workers to comprehend the guidelines and use them correctly (clarity of the: steps that ease interpretation and comprehension; definition of terms; and the nature of the language used).
3. Usefulness – the degree to which the workflow models enhance knowledge workers’ effectiveness through the benefits offered by process automation. This is measured using i). Practitioner perceptions towards the framework ii). Recommendability for usage of the modeling framework to other users.

## **6.2 Validation environment**

In validating the framework, we followed the action research cycle and its steps (planning, action, monitoring and reflection) to engage the medical practitioners first in workshop sessions with the intention of introducing a change to the existing form of practice [2] and secondly evaluating or collecting evidence using questionnaires regards the change that we introduced. During the workshop sessions, the Declare tool was installed on the participants’ computers and also printed guidelines distributed. Participants were guided on how to model their workflows. After three workshop sessions, questionnaires were administered to the participants to seek their feedback on the validation criteria. The responses received were collated and respective response percentages also computed. A sample population of 15 participants was selected from our case organisation. The selection was based on the purposive sampling method of Shajahan [25], because we needed knowledgeable participants about the CT BPs. The sample population was composed of doctors, data managers and clinical officers.

We worked with a clinical study team involved with the ‘datafax study’ in the case organization. The ‘data fax’ study has both manual as well as automated processes which include the following; patient registration, clinical diagnosis, data management, and reporting. We chose the patient registration, data collection during patient diagnosis using the case report forms (CRFs), and data validation processes. These provided an opportunity for our research to analyze several activities and sub processes embedded within the BPs.

## **6.3 Validation results**

Usability – Four parameters were used to measure the degree of usability of the Workflow Modeling Framework as shown in table 1.



**Table 1.** Validation results when using the patient registration process at case organization

Question	Yes (%)	No (%)
Framework learnability	73.3	26.7
Framework understandability	80	20
Satisfaction	93.3	6.7
Framework interactivity	86.7	13.3
Average percentage	80	20

- a) Learnability – the framework was found to be highly learnable (73.3%). This was owed to the logical and systematic arrangement of the modeling guidelines. It was also attributed to the ability of the participants to follow the steps in the provided framework guidelines to model their workflows.
- b) Understandability – this scored highly (80%) as the participants felt that the use of simple language, clear definition of terms, and lack of ambiguous confusing jargons made it easy for them to comprehend what the framework entailed. However, a section of the participants (20%) required more effort in trying to understand the framework, an aspect that could have been attributed to their traditional way of working.
- c) Satisfaction – the framework also scored highly (93.3%) at user satisfaction by the participants. This was attributed to the simplicity of the workflow modeling framework, reusability of process models, and the ability to easily verify and analyse workflow model performance issues. Only 6.7% of the participants posted a differing opinion and we could not quite readily establish the reason for that although we postulated it could have been a direct effect of resistance to change towards the new work processes.
- d) Interactivity – this scored 86.7%. The participants found the framework to be highly interactive due to the inter-linking and inter-related steps in the guidelines.
- e) Usefulness – this was measured using the practitioners’ perceptions towards the framework. The responses recorded in this respect included ‘I’m impressed that I can see which constraints are being violated in the workflow model’, ‘I am able to save my process instance model and reuse it later’, ‘I do not have to be a programming expert to do these models’ among others. These positive remarks indicated that the framework enabled the practitioners to effectively model workflows using the different components and recommended modeling tools. Additionally, we also measured the usefulness of the framework through recommendability – From the results in table 2, it was observed that 80% of the respondents were affirmative in regard to whether they would recommend the framework or not for use by other clinical trials. The respondents attributed this to the benefits they observed in using the framework including reduction in effort, cost and time in performing activities. They felt empowered by the framework because it provided a set of guidelines that sufficiently guided workflow modeling on their part.

**Table 2.** Framework recommendability

Validation criteria	Yes (%)	No (%)
Framework recommendability	80	20

- f) Flexibility – this was measured using efficiency and stability of the workflow models.

**Table 3.** Analysis of flexibility in the designed workflow models

	Mean	Std Deviation	Std Error of mean
Time to model Open Instance	3.20	0.447	0.200
No. of process tasks in open instance	2.20	0.447	0.200
Execution time (process instance 1)	2.40	0.548	0.245
Execution time (process instance 2)	2.20	0.447	0.200
Execution time (process instance 3)	1.60	0.548	0.245
No. of process tasks remaining unchanged (process instance 1)	1.20	0.447	0.200
No. of process tasks remaining unchanged (process instance 2)	1.20	0.447	0.200
No. of process tasks remaining unchanged (process instance 3)	1.80	0.447	0.200
No. of errors in process instance 1	1.00	0.000	0.000
No. of errors in process instance 1	1.00	0.000	0.000
No. of errors in process instance 1	1.00	0.000	0.000
Time to resolve errors in process instance 1	1.00	0.000	0.000
Time to resolve errors in process instance 2	1.00	0.000	0.000
Time to resolve errors in process instance 3	1.00	0.000	0.000

- a) Efficiency – this was measured based on the time taken to complete modeling a workflow, and the time taken to resolve errors as well as the level of productivity over a specific period of time versus the number of process tasks modeled within that time.
- i. Time – we further observed that less time was taken to model process instances of the work packages within CTs compared to using other traditional methods. This is illustrated from the measures of central tendency, where on average the modeling time was between 15-20 minutes with a standard deviation of 0.447 and a standard error of 0.2. This indicated that all responses were close to the average modeling time noted therein. Without use of workflow modeling systems, it was observed to take an average of more than 20 minutes to complete execution of a similar process scenario. With regard to error resolution, it did not take a very long time to resolve any identified errors within the workflow models. The observed error resolution time was on average less than 1 hour.
  - ii. Productivity – the number of tasks modeled as part of the open instance within the timescales mentioned above (15-20 minutes) indicated that only a reasonable amount of effort was required for workflow modeling by medical practitioners. In this case 17 tasks were modeled on average by the respondents. The results do indicate a standard deviation of only 0.447 with a standard error of 0.2. We thus observe that less effort was required to carry out work despite the changing process instances.
- b) Stability – we measured stability using; the number of tasks that remain unchanged with the addition, removal or modification of the designed workflow models for varying process instances, and the number of errors identified in an individual process instance model. From table 3, we also observed a large number of unchanged tasks in the different process instances. This is directly representative of the stability that is maintained within the processes irrespective of the changing process instances as highlighted in the results for process instances 1, 2 and 3. A mean of 1.2 that is representative of an average of 7 tasks is observed for the first and second instances with a standard deviation and standard error of 0.447 and 0.2 respectively. The third instance had on average 12 tasks that remained unchanged from the original open instance with a standard deviation of 0.477 and a standard error of 0.2. The results do indicate that flexibility was achieved in the designed workflow models for the different process instances based on the numbers of tasks that remained unchanged within the changing process instance models. With regard to the errors observed during the execution of the process instances, the results in tables 3 indicated no deviation from the mean value of the error related variables (number of errors encountered during execution of process instance 1, process instance 2, and process instance 3 as well as the time taken to resolve the errors in these three instances). A standard deviation and standard error of 0 was recorded for all the above mentioned variables. On average less than 5 errors were observed within all the 3 process instances. These results thus further indicate that the workflow models were able to sustain a high

degree of stability despite the changing process instances with very few errors introduced into the workflow models as a direct consequence of the changes occurring in the different process instances.

## 7 Conclusion and Future Research

The research was aimed at enhancing process flexibility in the designing of workflow models that could be easily interpreted by workflow engines. The workflow modeling framework was addressing the gaps (see section 1) that come along with the use of natural language text involved with CTs used by medical practitioners. As such, we identified the process-related tasks for BPs in CTs and the challenges faced by the medical practitioners while using the natural language text involved with CTs. Several requirements were derived though with specific attention to process flexibility that were used in the designing of the workflow models and related process instances in the modeling framework. The workflow modeling framework provides a set of guidelines that could be used by knowledge workers during the workflow modeling process. Using the framework we demonstrated an example of a designed workflow model and corresponding model process instances. The framework was validated using various criteria including but not limited to flexibility, usability, usefulness.

Notwithstanding the promising results observed from the workflow modeling framework, more work still needs to be done. First, the research was limited to enhancing workflow modeling for CT practitioners using flexible approaches but did not dwell much on the adoption and subsequent usage of the modeling framework. We therefore suggest to further study the qualitative and quantitative effects of the application and usage of the workflow modeling framework in CT settings. Further research could also be conducted on the identification of ways of better handling documentation processes especially within CTs with a possibility of integrating a documentation module in workflow modeling tools. The module would hence serve as a knowledgebase that can be used as a point for future reference. Lastly, the workflow modeling framework could further be subjected to more empirical validations in the CTs and other health related domains to determine its usefulness and usage in enhancing the way of working.

## References

1. Aalst, W. M. P., Pesic, M., Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. *Computer Science-Research and Development*, 23(2), 99-113. URL <http://dx.doi.org/10.1007/s00450-009-0057-9>.
2. Elliott, J. (1991). *Action research for educational change* (Vol. 49). Buckingham: Open University Press.
3. Becker, J., Rosemann, M., & Von Uthmann, C. (2000). Guidelines of business process modeling. *Business Process Management*, 241-262.

4. Becker, J., Weiss, B., & Winkelmann, A. (2009, January). Developing a business process modeling language for the banking sector—a design science approach. In Proceedings of the 15th Americas Conference on Information Systems (AMCIS 2009).
5. Capretz, M., Toledo, M. (2009). Web technologies in a collaborative platform for clinical trials. *RECIIS*, 3(4), 209-223.
6. Friedman, L., Furberg, C., DeMets, D.L. 1998. *Fundamentals of Clinical Trials*. Springer.
7. Fowler, M. (2005). Language Workbenches: The Killer-App for Domain Specific Languages? Retrieved from <http://martinfowler.com/articles/languageWorkbench.html>
8. Headlee, D. (2004). The paper trail: CRFs, Source Documents and Data collection tools. National Institutes of Health National Cancer Institute Center for Cancer Research Medical Oncology Clinical Research Unit
9. Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K., & Teschke, M. (1999). A comprehensive approach to flexibility in workflow management systems. *ACM SIGSOFT Software Engineering Notes*, 24(2), 79-88.
10. Khan, S. A., Kukafka, R., Bigger, J. T., Johnson S.B. (2008). Re-engineering Opportunities in Clinical Research using Workflow Analysis in Community Practice Settings. *AMIA 2008 Symposium Proceedings*. Pp.363-367
11. Lindsay, A., Downs, D., & Lunn, K. (2003). Business processes—attempts to find a definition. *Information and Software Technology*, 45(15), 1015-1019.
12. Marks, R.G. (2004). Validating electronic source data in clinical trials. Retrieved from [www.sciencedirect.com](http://www.sciencedirect.com)
13. March, S.T., and Smith, G.F. (1995) *Design and natural science research on information technology*. Decision Support Systems.
14. Peleg, M., Boxwala, A. A., Bernstam, E., Tu, S., Greenes, R. A., & Shortliffe, E. H. (2001). Sharable representation of clinical guidelines in GLIF: relationship to the Arden Syntax. *Journal of biomedical informatics*, 34(3), 170-181.
15. Regev, G., & Wegmann, A. (2005). A regulation-based view on business process and supporting system flexibility. In Proceedings of the CAiSE (Vol. 5, pp. 91-98).
16. Reijers, H. A. (2006). Implementing BPM systems: the role of process orientation. *Business Process Management Journal*, 12(4), 389-409.
17. Robson, C. (1993). *Real world research*. Blackwell: Oxford
18. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M, van Dongen, B.F., Alves de Medeiros, A.K., Song, M., Verbeek, (1995). A class of Petrinets for modeling and analyzing business processes. Technical Report 26, Eindhoven University of Technology.
19. Shahar, Y., Young, O., Shalom, E., Mayaffit, A., Moskovitch, R., Hessian, A., & Galperin, M. (2003). DEGEL: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines. *Artificial Intelligence in Medicine*, 122-131.
20. Shiffman, R. N., Karras, B. T., Agrawal, A., Chen, R., Marenco, L., & Nath, S. (2000). GEM: a proposal for a more comprehensive guideline document model using XML. *Journal of the American Medical Informatics Association*, 7(5), 488-498.
21. Schonenberg, H., Weber, B., van Dongen, B.F., van der Aalst, W.M.P. (2008). Supporting flexible processes through recommendations based on history. In: Dumas, M., Reichert, M., Shan, M.C. (eds) *International Conference on Business Process Management (BPM 2008)*, vol 5240 of *Lecture Notes in Computer Science*, pp 51–66, Springer-Verlag, Berlin.
22. Tu, S. W., & Musen, M. A. (2001). Modeling data and knowledge in the EON guideline architecture. *Studies in health technology and informatics*, (1), 280-284.
23. Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag Berlin Heidelberg.

24. Workflow Management Coalition (WfMC). (2005). Workflow Process Definition Interface - XML Process Definition Language (XPDL). Document Number WfMC-TC-1025.
25. Shajahan, S. (2007). Research methods for management. 6th Edition, Jaico Impression.
26. Mendling, J., Reijers, H. A., & van der Aalst, W. M. (2010). Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52(2), 127-136.
27. Huff, S. M., McClurec, R., Parkerc, C., Rochac, R., Abarbaneld, R., Beardd, N., ... & Zillf, D. A. (2004). Modeling guidelines for integration into clinical workflow. In *Medinfo 2004: Proceedings of the 11th World Conference on Medical Informatics*, [San Francisco, September 7-11, 2004] (Vol. 107, p.174)
28. Pesic, M., Schonenberg, H., & van Der Aalst, W. M. (2009). DECLARE Demo: A Constraint-based Workflow Management System. *BPM (Demos)*, 9.
29. Han, Z., Zhang, L., Ling, J., & Huang, S. (2012). Control-Flow Pattern Based Transformation from UML Activity Diagram to YAWL. In *Enterprise Interoperability* (pp. 129-145). Springer Berlin Heidelberg.
30. Pesic, M., Schonenberg, M. H., Sidorova, N., & van der Aalst, W. M. (2007). Constraint-based workflow models: Change made easy. In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS* (pp. 77-94). Springer Berlin Heidelberg.
31. Pesic, M., & van der Aalst, W. M. (2006, January). A declarative approach for flexible business processes management. In *Business Process Management Workshops* (pp. 169-180). Springer Berlin Heidelberg.
32. Redding, G., Dumas, M., ter Hofstede, A. H., & Iordachescu, A. (2010). A flexible, object-centric approach for business process modeling. *Service Oriented Computing and Applications*, 4(3), 191-201.
33. Pesic, M., Schonenberg, H., & van der Aalst, W. (2010). The Declare service. In *Modern Business Process Automation* (pp. 327-343). Springer Berlin Heidelberg.
34. Chiu, D. K. Li, Q. and Karlapalem, K. (1999). A meta modeling approach to workflow management systems supporting exception handling. *Information Systems*, 24(2):159 – 184.
35. Han, L. Yao, Z. Liu, S. and Zheng, Z. (2007). Application and research of ontology in workflow knowledge metamodel. In *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on*, pages 675 –680
36. Sadiq, S., Sadiq, W., Orłowska, M. (2001). Pockets of Flexibility in Workflow Specification. In *Lecture Notes In Computer Science: Proceedings of the Twentieth International Conference on Conceptual Modeling* (pp 513-526), Springer-Verlag, Berlin, Germany.
37. Winter, R. (2008). Design science research in Europe. *European Journal of Information Systems*, 17(5), 470-475.
38. Goedertier, S., Vanthienen, J., & Caron, F. (2013). Declarative business process modeling: Principles and modelling languages. *Enterprise Information Systems*, 1-25.
39. Pichler, P., Weber, B., Zugel, S., Pinggera, J., Mendling, J., & Reijers, H. (2012). Imperative versus declarative process modeling languages: An empirical investigation. In F. Daniel, K. Barkaoui & S. Dustdar (Eds.), (pp. 383-394) Springer Berlin Heidelberg.
40. Hevner, R. A., March, T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Q.* 28, 1 (March 2004), 75-105.
41. Hevner, R. A. (2007). A three cycle view of Design science. *Scandinavian Journal of Information Systems*. Jarrar, Y. 2000. ERP Implementation and Critical Success Factors, The Role and Impact of Business Process Management”, of The 2000 IEEE International Conference on Management of Innovation and Technology Conference proceedings, Singapore.