Modeling and Configuration of Process Variants for On-boarding Customers to IT Outsourcing

Wen Yao, Sujoy Basu, Jun Li, Bryan Stephenson

HP Laboratories
HPL-2012-157

Keyword(s):
process management, process variants, ontology, semantic rules, customer on-boarding, IT outsourcing

Abstract:
An essential part of IT outsourcing is to move the customer's IT environment into the service provider's mode of operation, which is known as customer on-boarding. It covers every aspect of transition and transformation, from the time the customer signs the contract to the time the provider can deliver steady-state IT services. In order to improve the repeatability and enforce adoption of best practices, a standard set of processes should be established to direct, control, and measure on-boarding activities for each customer. However, this process is very complex and often gets modified according to customer environments and requirements. It is very difficult to incorporate customizations needed for diverse scenarios into a single on-boarding process model. These process variations may lead to undesirable consequences, such as projects delays or even discontinuation of contract. In this paper, we propose an ontology and rule based approach to model and configure process variants for on-boarding customers to IT outsourcing. In this approach, the standard on-boarding process and business context that characterize various scenarios are captured in an ontological framework. Further, semantic rules model adaptation policies and help generate a customized process variant schema on the fly. Based on this framework, we developed a prototype to support process variant configuration. Further, we show that our approach can facilitate the configuration of process variants based on various customer needs and the management of process variants in a large repository.
Modeling and Configuration of Process Variants for On-boarding Customers to IT Outsourcing

Wen Yao*
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802, USA
wxy119@psu.edu

Sujoy Basu, Jun Li, Bryan Stephenson
Services and Solutions Research Lab
Hewlett-Packard Laboratories
Palo Alto, CA 94304, USA
{sujoy.basu, jun.li, bryan.stephenson}@hp.com

Abstract—An essential part of IT outsourcing is to move the customer’s IT environment into the service provider’s mode of operation, which is known as customer on-boarding. It covers every aspect of transition and transformation, from the time the customer signs the contract to the time the provider can deliver steady-state IT services. In order to improve the repeatability and enforce adoption of best practices, a standard set of processes should be established to direct, control, and measure on-boarding activities for each customer. However, this process is very complex and often gets adapted according to customer environments and requirements. It is very difficult to incorporate process variants needed for diverse scenarios into a single on-boarding process model, so that they can be reused. In this paper, we propose an approach based on ontology and rules to model the standard on-boarding process and configure process variants based on the business context that characterizes various scenarios. Further, semantic rules model adaptation policies and help generate a customized process variant schema on the fly. Based on this framework, we have developed a prototype to support process variant configuration. We also discuss the flexibility of our approach, and present its cost-benefit analysis.

Keywords—process management, process variants, ontology, semantic rules, customer on-boarding, IT outsourcing

I. INTRODUCTION

Information Technology (IT) outsourcing enables businesses to improve competitiveness by focusing on their core competencies. IT service providers offer standardized services by combining best practices such as ITIL with their specialized technical offerings and their expertise in different industry verticals. These standardized services are repeatable, and ensure that the services are delivered cost-effectively at high quality. The goal is to address the majority of the customer’s requirements with the standard processes, and innovate only where custom-crafted processes will have a significant impact on the customer’s business. It is observed that in many cases, the standard processes and technology components can be used, but must be adapted to meet specific customer requirements.

IT outsourcing is usually comprised of three phases, including (1) the solution design phase, in which the scope of service delivery is negotiated and a contract is signed, (2) the on-boarding phase, in which the customer’s IT services are moved from the current mode of operation to future mode of operation, managed by the service provider, and (3) the steady-state operation phase, in which the service provider delivers stable services as requested in the contract.

The on-boarding phase is critical since it gives customers the first impression of the provider’s services. This greatly influences the chances of the customer buying additional services from the same provider. The entire on-boarding phase is managed by the service provider as a standard process, similar to the many processes that the service provider follows during the steady-state operation phase. The standard process needs to be adapted due to varying requirements for different customers, industry verticals, geographic regions, scale of IT operations, etc. For example, the following presents three scenarios for variations that are captured along with the context for their occurrence:

(A) Customer A requires tracking of maintenance contracts and lease agreements on IT assets. As a result, the service provider needs to notify customer A about additional client-specific data attributes. After that, required data attributes are collected manually or automatically and then loaded into the data repository.

(B) Customer B is a large organization located in non-English speaking country with more than 10 service providers. As a result, the primary service provider needs to setup a translation service and a service exchange that allows bi-directional routing of service cases among the 10 providers.

(C) Customer C paid a premium for expediting the on-boarding process. Thus, the service provider needs to assign extra human resources and parallelize tasks where possible.

Although variations occur frequently in practice, it is very challenging to document, track and reuse the process variants that deviate from the standard process. In summary, we identified the following challenges:

• The standard on-boarding process is huge, hierarchical, and human oriented. In practice, variations can occur in different sub-processes of on-boarding, and they can be documented by the people responsible for that sub-process. Usually, these changes must be reviewed by the process owner for every new context. However, this review process takes time as input may be requested from other people involved in this and related sub-processes. Thus, it is difficult to integrate a large number of variations all together in the on-boarding process.
• **Mapping from the context of the outsourcing projects to the available concrete sub-processes and possible adaptations is difficult.** Current practice usually involves manual work to understand the project’s context and identify required adaptations of the standard process. Furthermore, there is need for manual configuration of the parameters for every identified process variant. Such manual configuration has to be repeated if the identical process variant occurs in another outsourcing deal. In order to reduce the steps involved in tedious manual configuration, we need a formal framework to capture the cause-effect relationship between the business context and the variations in the standard on-boarding process.

• **Variations in past cases are managed in an ad hoc manner.** After an on-boarding process is completed, significant amount of interesting data related to the process variations remains in the text fields in project management databases, Microsoft Project files, emails and other text documents exchanged among members of the on-boarding team. There is no efficient way to compare the variations from a standard process in a large repository containing these documents. Hence, continual improvement of the on-boarding process is burdensome.

In [1], we explored these challenges in details and offered a set of recommendations for a governance framework that can manage complex outsourcing deals. In this paper, we propose a process variant modeling framework to capture the standard on-boarding process and the business context in a common knowledge base and use rules to configure a process variant on the fly which is tailored for the business context of a specific customer. With respect to the technical challenges that we identified above, our framework shows:

• **We are able to use an ontology-based approach to model the standard on-boarding process that is large, complex, hierarchical and human centered.** It facilitates a shared understanding of this on-boarding process among different people that are involved, and provides a common vocabulary for people and software agents. It is the first step towards addressing the problem of identifying what adaptations are needed for a specific customer context.

• **We are able to dynamically introduce adaptations into the standard process using semantic rules.** This allows the service provider to design flexible yet constrained process variants based on customer requirements. The resulting variant is an ontological process model that can be further converted into executable process such as BPEL models [2] for process tracking and monitoring.

• **We can facilitate search and management of process variants in a large repository.** Since each process variant is contextual, we can retrieve a small set of desired variants using context variables according to the user query.

This paper is structured as follows. Section II describes related work. After that, we discuss our approach in Section III. Section IV introduces the ontological knowledge base that unifies the modeling of the on-boarding process and business context. Then we introduce a variant configuration algorithm using semantic rules in Section V. Section VI presents implementation and examples. Finally, we conclude the paper in Section VII along with the plans for future work.

## II. RELATED WORK

### A. Modeling IT Service Management Processes

Practitioners from the business world have been actively involved in establishing standards by consolidating their knowledge to define generally accepted rules and processes. Various best practice frameworks have been developed for IT, among which ITIL [3] and COBIT [4] are the most popular standards. Since these frameworks are usually in the form of plain text documents, a practitioner needs to look up the documents manually and then apply the best practice.

A number of studies realized the necessity and advantages of building an ontological meta-model against the existing linguistic meta-model. For example, Geoken and Alter [5] represented IT governance frameworks as Entity-Relationship models in the context of the COBIT standard. Other governance models for IT outsourcing have been studied by researchers in [6, 7]. However, none of these models is actionable or executable. Ayachitula et al. [8] present a hybrid approach to integrate human-centric and automatic-centric workflows in IT service management automation. They showed that this approach can reduce the complexity in IT process management.

Despite a number of linguistic meta-models, there is only a limited amount of research in modeling and automating processes in IT outsourcing. As a result, process variations are hardly be treated in these approaches.

### B. Process Variation Management

Techniques for supporting dynamic changes during process execution have been extensively studied. For example, a rule-based approach has been proposed to modify a running instance in response to exceptions, using a number of change operations like task insertion, deletion, etc. [9, 10]. Hwang and Tang [11] proposed to consult past exceptions to facilitate workflow exception handling. The focus of these studies is on modifying a running process instance when exceptions occur due to failed tasks, erroneous information, etc.

Recent years have witnessed a growing interest in enabling process variant configuration tailored to the business context. Reference process models are proposed to handle the large number of processes with variations [12]. Domain experts are allowed to configure the variation points according to their own needs. For example, a questionnaire-based guidance component [13] is developed to lead an end user to configure process variants based on the formal conceptualization of domain knowledge. Other studies include a template and rule based approach [14, 15], multi-layered approach [16], and version control approach [17, 18]. Further, Schnieders and Puhlmann [19] proposed a software development approach to allow for rapid and cost-
effective development and deployment of customer tailored process oriented systems.

Although many studies have contributed towards the configuration of process variants, they usually assume the base or template process model have been predefined in a workflow execution language such as BPEL [2]. When the base process is huge and complex, there might exist thousands of variation points. Thus, the applicability of existing approaches is doubtful.

C. Variant Search and Discovery

There is a considerable amount of research on designing flexible processes and configurable process models. However, there is only a limited amount of research in actually managing a large number of variants in the repository. Most studies focus on configuration of flexible workflows with only limited coverage of querying issues, such as [14-16]. The perspective taken in these works is to find variants that are "similar" to a given variant or to a given query pattern. This gives rise to the need for various metrics of similarity. For example, Dijkmana et al. [20] present three similarity metrics used to answer queries on a large process repository, including node matching, structural, and behavioral similarity. Indexing techniques are proposed for quickly querying process graphs based on similarity [21, 22].

Retrieving process variants based on structural similarity may return irrelevant or inaccurate results, because the onboarding process is highly dependent on the business context of customer on-boarding. There is a need to develop novel approaches that take the business semantics into consideration for retrieving desired process variants in a large repository.

III. OUR APPROACH

In this paper, we propose an approach using semantic web technologies to model and configure process variants, with the following main objectives:

- Formalize on-boarding related process, knowledge, and its business context in a common, actionable framework.
- Allow configuration of process variants tailored for the business context of a specific customer.
- Facilitate search and management of desired process variants within a large repository.

A. Design Methodology

Figure 1 shows our design methodology in three phases. The data collection phase (step 1-3) includes case studies in on-boarding projects and interviews with on-boarding managers and team members. We used ITIL-based process guides developed in HP and also collected process logs of several past projects in Microsoft Project files (step 1-2). We analyzed deviations in these process logs with the help of domain experts in order to link them with individual steps in the standard process guides (step 3). The modeling phase (step 4-7) transforms collected data into a formalized framework. Process designers and knowledge engineers are involved in this phase. The main focus of this paper is on this phase. The system architecture and technical details are discussed in the following sections. Finally, the deployment phase (step 8-10) allows our prototype to be used by the on-boarding team and global process owners. We will use an example to illustrate how the system works in Section VI.

B. System Architecture

We implemented a proof-of-concept system as a result of the modeling phase. Figure 2 depicts the system architecture. Our ontological knowledge framework includes two parts: the Semantic Business Process Model (SBPO) that captures the important concepts and their relationships in our body of knowledge on customer ON-Boarding (ONBRD), and the business context model that captures the sources of process variations. Semantic rules are developed to model the cause-effect relationships and used to generate process variants. The system allows an end user (e.g., the on-boarding manager) to create process variants on the fly, use them for tracking the on-boarding process in practice, and retrieve the desired variants from the repository. The important components are discussed as follows:

**Ontological knowledge framework:** We use an open source ontology editor, Protégé 3.4 [33], to encode two ontologies: SBPO and ONBRD, which are both formalized in Web Ontology Language (OWL) [23].

**Jena reasoning engine:** The execution of semantic rules requires the availability of a rule engine, which performs reasoning using a set of rules and a set of facts as input. Any new facts that are inferred are added to the set of existing facts. They will potentially fire more rules (i.e., forward chaining). In our system, we use the Jena rule engine [24] to enable rule-based reasoning on the Protégé data set.

**Variant configurator:** This component is developed in Java to generate customized on-boarding process variants on the fly. It acquires case data from an end user and feeds it to the context knowledge base, so that semantic rules can be triggered to produce a variant automatically. It extracts all context variables from the business context model ONBRD and presents them to the end user in a user friendly format. Thus, the user can easily select a value and assign it to that.
context variable. After a variant is generated, it will be presented to the user in the graphical model but is also available in OWL.

**Variant retriever:** This component searches process variants in the repository with a given query from the end user. Since all the process variants are configured with the rule-based approach and stored in the repository, they can be retrieved using the context variables.

---

**C. Our Contributions**

Recent studies have initiated an effort to formally model and automate IT service management processes. However, they hardly consider and integrate frequent variations into those models and reuse them in future cases. On the other hand, related work on process variant configuration and management cannot be applied in this context because they fail to address the complex, distributed, and human centered nature of the customer on-boarding process.

In this study, we aim to bridge this research gap. We use a formal design methodology to develop a semantic process management model that allows automatic customization of the customer on-boarding process. Unlike other semantic process models, e.g., OWL-S [25], which support a core set of constructs to define Web services, our model emphasizes high level business objectives and organizational responsibilities, recognizing that many tasks are human oriented and cannot be automated by Web services. It also enables low-level workflow configuration triggered by the customer context. Our approach can be easily used by non-technical people for variant configuration and management.

---

**IV. ONTLOGICAL KNOWLEDGE BASE**

An ontology is a formal representation of knowledge as a set of concepts within a domain and the relationships among these concepts. It can facilitate knowledge sharing, logical inference, and knowledge reuse. In this study, we use the OWL [23] to encode two ontologies so that they can be shared across various people, organizations and software agents. The first ontology is SBPO. It addresses the challenges of modeling the on-boarding process, which is complex, huge, and mostly executed by people rather than computers. The second ontology is ONBRD. It satisfies the need for a domain-specific model for the business context of on-boarding. Since OWL uses first-order logic, the models and description of data in these models can be formally validated.

**A. The On-boarding Process Model**

Ontology-enabled semantic process modeling provides a truly unified view on the process space of an organization in a machine-readable form [26]. It facilitates process modeling down to the operations level from the business perspective. The actual execution of these operations may be done either on IT systems, or manually by human beings (i.e., depending on the task type). Thus, it is suitable for use in a collaborative and complex environment where people from different functional units are involved.

---

We use two major resources to extract domain knowledge: the standard process guides based on ITIL and interviews with on-boarding managers. Figure 3 shows a partial view of our SBPO model that has been built following the standard ontology development methodology [27]. It supports the modeling of the on-boarding process, in terms of domain concepts, such as Business Goal, Business Process, Role, Resource, etc. as well as their semantic relationships. These concepts are divided into two levels concerning the business and the technical perspectives of a process respectively. The business process space deals with high level activities and their compositions, roles and responsibilities, business goals, etc. The workflow space deals with low level task execution and coordination. For example, a workflow model specifies whether a task is executed by Web services or human beings; whether two tasks are executed in parallel or in sequence. Note that the on-boarding process is constrained in some way by the capability of ontology modeling, but it is still flexible because no strictly defined control flow relationship is enforced among tasks. All the processes and entities from the on-boarding process are captured in the ontological database. For example, figure 4 shows partial representation of the composite process “IT Asset Management (ITAM) delivery model”, which is composed of other atomic or composite processes.
The proposed ontological knowledge framework captures the knowledge of the standard on-boarding process in terms of important concepts and their semantic dependencies. Their contextual variables are also captured. In this section, we will introduce semantic rules and propose an algorithm for process variant configuration.

### A. Semantic Rules

Rules are usually represented in If-Then statements. Thus they can be used to capture the cause-effect relationship between the business context and variations in the on-boarding process. In this study, rules were developed in consultation with domain professionals from the on-boarding teams. They were first expressed in a natural language and then re-written in Jena [24], the formal language for encoding semantic rules.

Our semantic rules address the two semantic spaces and their intersection. We categorize them accordingly as business rules, variant configuration rules, and workflow adaptation rules. Figure 6 shows where each category is applied. Triggering of certain rules can add inferred information to the captured business context that in turn triggers other rules, based on forward chaining.

**Business Rules (BR)** formalize business policies and address the business context space of customer on-boarding (ONBRD). For example, organizations may have different policies to determine the deal size and discount rates for different deals. Rule BR_1 is for an organization where the deal is defined to be large if the deal amount is larger than 50 million:

\[
\text{BR}_1: (\text{contract onbrd:hasDealAmount } ?\text{amount}), \text{greaterThan} (\text{?amount}, 50,000,000) \rightarrow (\text{?contract, onbrd:hasDealSize "large"})
\]

**Variant Configuration Rules (VCR)** are triggered by the business context space (ONBRD), produce change operations, and apply them on the standard process template from the on-boarding process space (SBPO). Hence they address the intersection of the two semantic spaces. The change operations or adaptation patterns cover five different perspectives of a business process model, including organizational, functional, behavioral, data, and business goal. Organizational perspective involves roles and responsibilities in the process; functional perspective captures what activities are performed; behavioral perspective describes the execution sequence of activities; data perspective records the data flow among activities; finally, business goals, KPIs, and metrics are available in the goal perspective. For example, the following rule VCR_1...
shows an additional role for subject-matter expertise (ProcessSME) is required to be responsible for a certain process when dealing with large deals. Rules that reflect other perspectives are formalized in a similar way.

VCR_1: (?contract onbrd:hasDealSize "large"), (?contract, onbrd:handledBy ?bp), (?bp isComposedOf ?bp2), (?bp2 sbpo:hasOutput sbpo:IT_process_transformation_solution) \rightarrow (?bp2 hasResponsibleRole sbpo:ProcessSME)

**Workflow Adaptation Rules (WAR)** enable self-adaptation of a workflow model, and address the on-boarding process space (SBPO). They are necessary because data inconsistencies and wrong task execution order can be the outcome after adaptation rules from BR and VAR have been applied. The adaptation rules from WAR perform consistency checks to ensure that the control-flow and data dependencies are satisfied, and restore structural correctness to the resulting process variant. For example, if a task \( t_1 \) is removed by variant configuration rules, all the tasks dependent on it should be removed as well. In rule WAR_1, which is based on Jena notation, ‘remove(1)’ indicates that the second triple in the condition part, i.e. (?t2 sbpo:belongsTo ?bp), will be removed. This implies that the relationship between task \( t_2 \) and business process \( bp \) will be removed, since \( t_2 \) is dependent on removed task \( t_1 \).

WAR_1: (?t2 sbpo:dependentOn ?t1), (?t2 sbpo:belongsTo ?bp), noValue (?t1 sbpo:belongsTo ?bp) \rightarrow remove (1)

These three categories of adaptation rules allow separation of business policies, process adaptation policies, and workflow adaptation strategies. The rules in each part will be managed by a person with the corresponding perspective. For example, when business policies change, only impacted business rules need to be updated. Variant configuration rules will be updated only when the policy for IT operations based on business requirements changes.

**B. Variant Configuration Algorithm**

Figure 7 shows the variant configuration algorithm. It takes case data which characterize an outsourcing deal as input and generates a process variant schema that is tailored to customer needs. If no business rule or variant configuration rule is triggered (i.e., no customization required for this case), then the output variant is the process template since no change has been applied (line 4-7); if any variant configuration rule is triggered, it may trigger workflow adaptation rules as well for self-adaptation (line 8-9). The details of how these rules work and their semantic space can be found in the previous section. The resulting workflow model with adaptations will be the ultimate process variant. It can be further transformed into a BPEL process. In the next section, we will use an example to illustrate how this algorithm works.

Algorithm Process_variant_configuration
---
Input: case data CD
Output: process variant PV

1: PV = SBPO //the default variant is the process template
2: Semantic rule set SR = \{BR, VCR, WAR\}; //BR denotes business rules, VCR denotes configuration rules, WAR denotes workflow adaptation rules
3: Update ONBRD context base with case data CD
4: if no rule in BR is triggered
5: do nothing; //if no rule is triggered, then output the standard process template
6: else if any rule in BR is triggered && no rule in VCR is triggered
7: do nothing;
8: else if any rule in VCR is triggered, customization is needed
9: Apply all rules in WAR for workflow self-adaptation //this leads to self adaptation
10: End if
11: Output PV;

---

**Figure 7. Algorithm for process variant configuration**

### VI. IMPLEMENTATION AND EXAMPLES

**A. Implementation**

Based on the techniques presented in Section IV and V, we have developed a proof-of-concept prototype that follows our system design in Figure 2. The SBPO model includes 20 classes, 35 object properties, 22 datatype properties, and 41 restrictions. To build the knowledge base, we have encoded the data obtained from HP process guides. We found more than 200 composite processes, and 50 atomic processes that comprise a large number of tasks. Five organizations, 23 roles, and 83 types of resources are involved. In total, more than 400 instances are inserted in the SBPO knowledge base. The ONBRD model includes 6 classes, 8 object properties, 8 datatype properties, and 15 restrictions. It can be extended further as we identify more context variables in future. Both SBPO and ONBRD ontologies have been validated in terms of their logical consistency using the Pellet reasoner, which is a Protégé plug-in. Next, we use two examples to show how a process variant can be configured and retrieved using our prototype.

**B. Example 1 – Process Variant Configuration**

David is an on-boarding manager and he is responsible for on-boarding customer ABC Company to IT outsourcing. ABC Company is a large corporation in the telecom industry, and is located in Europe. This deal amount is $120M and they have some requirements in the contract, such as “need to track maintenance contract”. Figure 8 (a) shows how this case can be handled using our variant configurator. David is provided with a number of questions that are derived from our context model and he is guided with choices, while filling out the customer data. Since we have formalized the standard processes in the ontological knowledge base and captured the variations by semantic rules, our algorithm (see Figure 7) allows automatic configuration of this on-boarding process. The resulting variant is shown in Figure 8 (b).

The left panel shows high level activities (or processes) and their compositions in a hierarchical structure. The upper right panel shows different perspectives of the selected process in the left panel, including roles and responsibilities, data, resources, etc. The user can click each tab to learn the
details. The lower right panel depicts the workflow model graphically for that selected process. Required tasks and their coordination can be easily understood from the graph. This variant is loaded from an ontology model (encoded in OWL) that captures various aspects of a business process. It can be further transformed into a standard BPEL process.

(a) Configuring a new customer on-bording process

(b) Representation of a generated process variant

Figure 8. Process variant configuration

C. Example 2 – Process Variant Retrieval

Process owners may want to search for desired process variants in the repository. As the repository becomes very large, say, with hundreds of thousands of process variants, search becomes a critical problem. Search based on structural similarity often returns a large result set. Even worse, most of the retrieved variants might be irrelevant to the user’s business context. Our approach addresses this issue, since each variant is configured based on the context model. Figure 9 shows all the past process variants for on-boarding projects with medium deal size, where customer is located in America and is from the fashion industry. This query is derived by reading the values of the three context variables provided by the user (shown in the left panel). By supporting queries on context variables, we are able to find the small set of desired variants. The selectivity of the query can be increased by increasing the number of context variables. We have conducted simulations to verify the advantage of our ontology and rule-based approach over retrieval based on structural similarity. For a repository with 100,000 process variants, increasing the number of context variables from 1 to 3 decreased the number of retrieved variants from over 3000 to below 30. The latter is a reasonable number for a process owner to browse, and select the most appropriate variant.

Figure 9. Variants retrieved from the process repository

D. Discussion and Implications

Challenges addressed: Our approach makes several contributions in terms of the above mentioned challenges. First, the standard on-boarding process and the business context that characterizes different scenarios are formalized in a common and actionable framework. It facilitates a shared understanding of the on-boarding process among different people, organizations and software agents that are involved. Second, this approach allows the service provider to generate flexible process variants based on customer environments and requirements. The resulting variant can be further converted into an executable process model used for process tracking and monitoring. In case the policy changes, only certain rules need to be modified while the process template remains the same. Managing rules is much easier than maintaining thousands of variation points as described in contemporary methods. Finally, this approach can facilitate search and management of process variants in a large repository. Since each variant has contextual metadata, we can retrieve a small set of desired variants using context variables according to the user query. This approach can expedite the on-boarding process and reduce costs in IT outsourcing practice.

Cost-benefit analysis: A limitation of our approach is the complexity and cost involved in the development of the base process template and the repository of rules. Initial activities that contribute to this initial setup cost includes: (1) interview on-boarding team members and capture all process variants from previous project logs and map them with their context variables in cause-effect relationships, and (2) have the knowledge engineer encode the standard process from process guides as well as the information captured from the previous step in the proposed system, and (3) validate the semantics of the resulting model, and (4) ask an on-boarding manager to test the usability with various scenarios. While this initial setup cost can be significant, the incremental cost
of adding new process variants diminishes rapidly as the adaptation rules cover an increasing number of scenarios. For comparison, we consider manual configuration where there is no initial setup cost, and a constant cost for configuring each variant. As shown in Figure 11, there is an inflection point beyond which our approach is more cost-effective. Based on our interviews of the on-boarding team members, this inflection point could be reached in as few as a hundred variants, if the initial setup cost of our approach is managed well.

Figure 11. Cost-benefit analysis

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we developed a modeling framework that is based on ontology and semantic rules to create and manage process variants in IT outsourcing. Although our focus is on the customer on-boarding phase, this methodology can be easily applied to other phases during IT outsourcing. We show that this approach can facilitate configuration of process variants and search in a large repository, especially when the query is precisely specified. Then we discuss the flexibility of our approach and present its cost-benefit analysis.

We are gathering feedback from domain practitioners. We intend to refine our prototype with their feedback. Future work includes a more user friendly interface, so that workflow participants can input any variations that they introduce in their running process instances. Data on such ad-hoc variations can be aggregated, analyzed and used for the purpose of incorporating new process variations as standardized elements in the repository.

ACKNOWLEDGMENT

We thank our colleagues and the reviewers for their insightful comments on the earlier draft of this paper.

REFERENCES