SEVENTH FRAMEWORK PROGRAMME
Challenge 1
Information and Communication Technologies

Trusted Architecture for Securely Shared Services

Document type:	D3.1
Title:	Design of a semantic underpinned, secure & adaptable process management platform (1)
Work Package:	WP3
Deliverable Number:	D3.1
Editor:	Jutta Mülle, University of Karlsruhe
Dissemination Level:	PU
Preparation Date:	31 December 2008
Version:	1.0

Information Society
Technologies

The TAS³ Consortium

Nr	Participant name	Country	Participant short name	Participant role
1	K.U.Leuven	BE	KUL	Coordinator
2	Synergetics nv/sa	BE	SYN	Project Manager
3	University of Kent	UK	KENT	Partner
4	University of Karlsruhe	DE	KARL	Partner
5	Technical University of Eindhoven	NL	TU/e	Partner
6	CNR/ISTI	IT	CNR	Partner
7	University of Koblenz-Landau	DE	UNIKOLD	Partner
8	Vrije Universiteit Brussel	BE	VUB	Partner
9	University of Zaragoza	ES	UNIZAR	Partner
10	University of Nottingham	UK	NOT	Partner
11	SAP research	DE	SAP	Partner
12	Eifel	FR	EIF	Partner
13	Intalio	FR	INT	Partner
14	Risaris	IR	RIS	Partner
15	Kenteq	BE	KETQ	Partner
16	Oracle	UK	ORACLE	Partner
17	Custodix	BE	CUS	Partner
18	Medisoft	NL	MEDI	Partner

Contributors

	Name	**Organisation**
1	Jutta Mülle, Jens Müller	University of Karlsruhe
2	Alex Boisvert, Arnaud Blandin	Intalio
3	Jeroen Hoppenbrouwers	Synergetics
4	Quentin Reul	VUB / Starlab

1 Executive Summary

TAS³ has the goal to provide a next generation trust & security architecture that is

- ready to meet the requirements of complex and highly versatile business processes,
- enables the dynamic user-centric management of policies, and
- ensures end-to-end secure transmission of personal information and user-controlled attributes between heterogeneous context-dependent and continuously changing systems.

The topic of work package three is the support of adaptive, secure business processes in the TAS^3 architecture.

This document describes the conceptual design and basic components of the system architecture for business processes support developed during the first project year in WP3. In the following versions of this deliverable the ongoing research results will be added and the report continued.

The TAS^3 architecture is based on executing business processes based on web services. Therefore, we first provide a stable process modelling and execution framework in a service-oriented application area. Following the analyzed requirements, the conceptual design will provide mechanisms and concepts on one hand for secure, privacy-preserving business processes and on the other hand for allowing to alter and adapt process schemas of running instances to concrete content of the process, e.g. to select appropriate services in respect to security properties, quality properties, and requirements of involved data. In order to support modelling security specification and adaptability of processes, ontology methods will semantically underpin the business processes.

The report concludes with establishing an example process out of the employability scenario as basis of further validation of the research results.

Table of Contents

2 Introduction

Scope and objectives

TAS³ has the goal to provide a next generation trust & security architecture that is

- ready to meet the requirements of complex and highly versatile business processes,
- enables the dynamic user-centric management of policies, and
- ensures end-to-end secure transmission of personal information and user-controlled attributes between heterogeneous context-dependent and continuously changing systems.

The topic of work package three is the support of adaptive, secure business processes in the TAS^3 architecture.

This report starts with setting up the requirements of TAS^3 scenarios to the business process management support and to introduce guidelines of the system architecture for secure adaptable business processes in respect of integration into the overall architecture. Then the conception of a framework of business processes for TAS^3 is described, which has been established during the first project year. In the following versions of this deliverable the ongoing research results will be added and the report continued.

The TAS^3 architecture is based on executing business processes based on web services. Usually, several services and data sources are adequate to support distinct activities in the business process and there should not be the need to determine in advance which activity/service will be used during execution.

Special emphasis holds on processes that compute privacy-relevant data, personal information about health of persons in e-health applications and about education and e-portfolio of persons to allow for employment and training services. The services are provided in a distributed environment and are flexibly offered via web services with many involved participants at distributed sites.

Therefore, we first need to provide a stable process modelling and execution framework in a service-oriented application area. Following the analyzed requirements, the conceptual design then introduces mechanisms and concepts on one hand for secure, privacy-preserving business processes and on the other hand for allowing to alter and adapt process schemas of running instances to concrete content of the process, e.g. to select appropriate services in respect to security properties, quality properties, and requirements of involved data.

In order to support modelling security specification and adaptability of processes, ontology methods will semantically underpin the business processes. Ontology of business processes will be used at the modelling and execution level to add security issues in order to achieve processes with a specific security level or quality and helps to allow for process adaptation.

Finally, an example process out of the employability scenario establishes a basis for further validation of the research results. The process model as result of modelling one of the scenario business processes of the employability application area is described. The goal is to set up a real-world business process for validating the developed concepts and framework for business process support in future. In this report it was set up and used to analyze and refine requirements and to start with a first system architecture and conceptualization of adaptive and secure processes and high-level ontologies.

Document structure

The rest of this document is organized as follows.

Chapter 2 presents the results of requirements analysis applied to the distinct research perspectives in this work package, i.e. to support the life cycle of business processes with BPMN modelling tool, BPEL execution engine and the BPEL4People workflow model, to focus on enhancing processes with security specifications and managing security within processes, to introduce the aspect of adapting processes during execution and so to provide flexible processes in an open heterogeneous and dynamic environment, and to develop a process ontology in order to underpin adaptable secure processes with ontology. This Chapter concludes with an overview about the role of business processes in the TAS^3 system architecture concerning the integration perspective.

Chapter 3 contains the present status of the conceptual design at the end of the first project year. The presentation is organized along the introduced research perspectives and describes the basic approach taken, the challenges and first results.

Aiming at future validation of the conceptual design, in Chapter 4 we present a concrete process model of one of the application scenarios of the TAS^3 project, namely e-employability. In concrete, the accreditation of prior learning process of Kenteq, shortly Kenteq APL process, was taken as a first real-world process and modelled with BPMN (the standardized Business Process Modelling Notation for conceptual process modelling). We will refine the modelling and enhance it in the next year and use it for validation of the conceptual design and its implementation. Chapter 5 contains a conclusion of this report.

3 Requirements and System Architecture

Lifecycle of Business Processes

The purpose of TAS³ is to create a *trusted architecture for securely shared services*. Thus, a crucial aspect is the successful interaction of *(web) services* with different owners. The process of coordinating the sequence and data flow is known as *orchestration*.

Deploying a new or changing an existing business process consists of several distinct steps: First,

we have to create a (formal) description of the process (or change the existing one). Then, this description is possibly translated into a different form and deployed to a server. Based on certain events (like incoming web-service calls), the server creates new instances of the process and executes them.

For the TAS³ project, we follow a standards-based approach. Thus, BPEL (1) is the language of choice for describing executable business processes, as is BPMN (2) for modelling them. In Figure 1 a rough schema describes the levels of business processes and how to use in TAS^3.

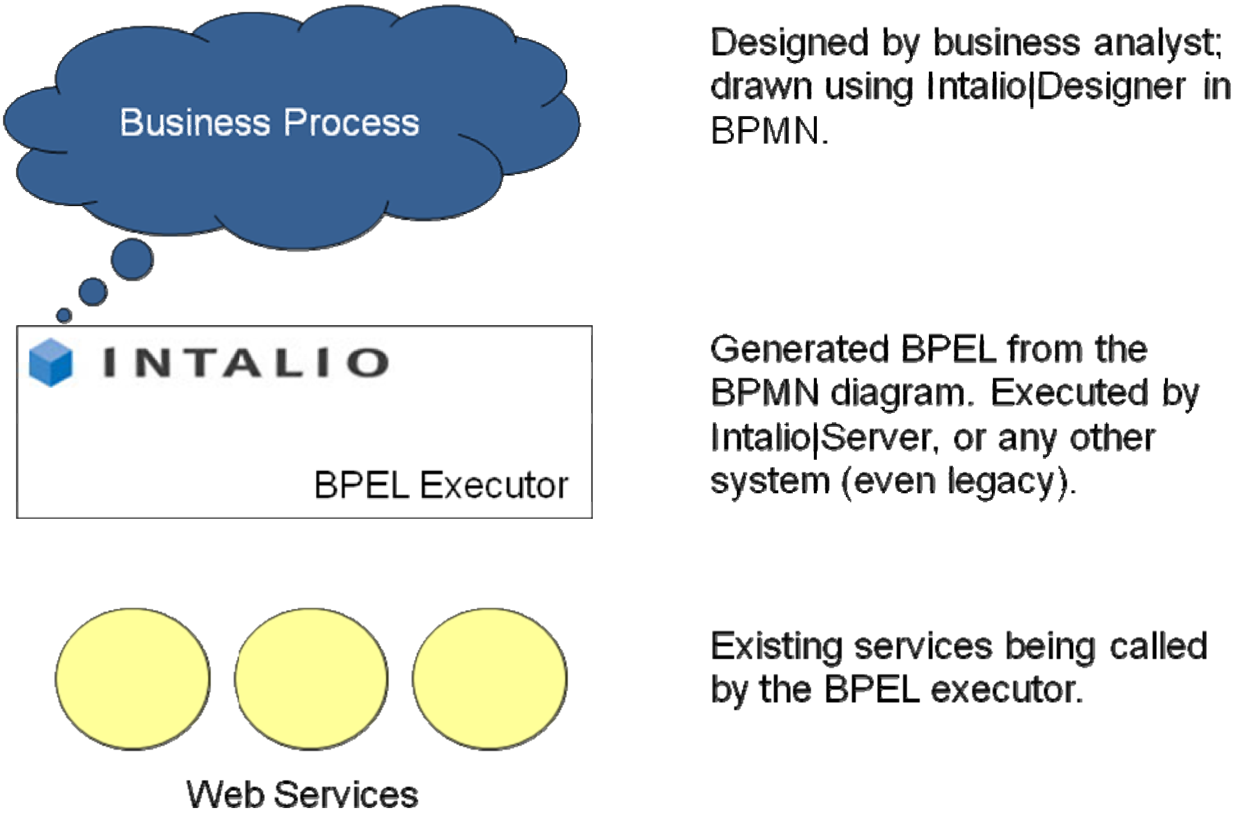

Business Process
Designed by business analyst; drawn using Intalio|Designer in BPMN.
INTALIO
BPEL Executor
Generated BPEL from the BPMN diagram. Executed by Intalio|Server, or any other system (even legacy).
Existing services being called by the BPEL executor.
Web Services

Figure 1: Business Process Layers in TAS^3

Thus, the deployment of a process consists of several phases, with distinct components responsible for each phase:

1. The modelling of the process (as a BPMN model enhanced with annotations to support the execution) is the responsibility of a BPMN modelling tool, such as Intalio|Designer.
2. Another task is the translation of the graphical model into an executable form and the deployment of the resulting BPEL process and XForms onto the BPEL execution and workflow engines, respectively. This is performed by Intalio|Designer, as well.
3. Finally, the resulting process descriptions can be instantiated and executed. In the Intalio|BPMS system, this task is split between the Apache Ode server, which handles the execution of the actual BPEL process, and the Tempo engine, which handles the people interaction with workflow forms.

Any comprehensive approach on the integration of security and adaptability of business processes must address all three phases, and consequently all of the corresponding software components.

BPMN Modelling Tool

BPMN

The Business Process Modelling Notation (BPMN, see (2)) specification provides a graphical notation for expressing business processes in a business process diagram. The objective of BPMN is to support business process management by both technical users and business users by providing a notation that is intuitive to business users yet able to represent complex process semantics. The BPMN specification also provides a mapping between the graphics of the notation to the underlying constructs of execution languages, particularly BPEL.

Intalio|Designer

Intalio|Designer (see (3)) is a modelling tool that supports the BPMN notation and can generate executable BPEL process definitions from business process diagrams and additional information provided by the user, such as data assignments. Intalio|Designer also supports workflow user interactions through integration with the Tempo project, which is an implementation of BPEL4People architecture (see section 2.3 for details) but still lacks compliance with the BPEL4People specification[1] (see (4))

Security Aspects

Security aspects are out of scope of the BPMN specification. While it is possible to model different participants in separate pools or lanes, the semantic meaning of such separation is vague and does not have defined bearing on security aspects.

Intalio|Designer currently supports annotating pools representing people with organizational roles to define the authorization between processes and people tasks. Beyond this, Intalio|Designer does not support natively modelling TAS[3] security aspects.

In Chapter 4 there are several screenshots of the BPMN Designer Tool of Intalio. It would be desirable to extend support for TAS[3] security aspects and separation of duty through annotations to provide declarative means of specifying security policies instead of leaving these to be implemented in an imperative manner by business or technical users.

Further planned steps

Based on the above, our plan with respect to the BPMN Modeling Tool is as follows:

- Explore avenues to extend the BPMN notation and Intalio|Designer with TAS[3] security annotations;
- Enhance Intalio|Designer to generate BPEL definitions using BPEL4People's <b4p:peopleActivity> instead of the BPEL <bpel:invoke> activity;
- Enhance Intalio|Designer to improve support for separation of concern through additional annotations;

[1] As of this writing, the BPEL4People specification is not yet finalized. It is expected that it will be ratified as an OASIS standard.and finalized some time during 2009.

- Enhance Intalio|Designer to support data assignment and mapping between the process' security context, and other participants' security context (BPEL partnerLinks) to support infrastructure-level policy enforcement.

BPEL Execution Engine

BPEL

The Business Process Execution Language (BPEL, see (5)) is a domain-specific language that allows for business process logic to be expressed decoupled from existing software yet tied to external software through service invocation. This reduces and potentially eliminates the need to code business process logic in a traditional programming language, such as Java, C/C++, etc. This clear separation of concern between business processes and software services makes process logic (e.g., workflow management) simpler, more focused and easier to manage.

Apache Ode and Intalio|BPMS

Apache Ode is an open-source implementation of the BPEL 2.0 specification and is governed by the Apache Software Foundation. It is the BPEL engine used within Intalio|BPMS product suite to execute processes and is integrated to provide seamless deployment from Intalio|Designer.

Security Aspects

There are significant challenges related to use of BPEL technology in secure distributed computer systems and web services. Of particular interest to the TAS^3 project are the problems of authorizing users to execute tasks within a workflow while enforcing constraints such as separation of duty on the execution of those tasks, conducting end-to-end secure transactions between multiple participants, and carrying a security context between parties involved in a given process.

While BPEL may be coupled to a web service security infrastructure (i.e., WS-Security) to provide integrity and confidentiality at the transport level, the BPEL language does not provide any support for the specification of either authorization policies or authorization constraints on the execution of activities composing a business process. We believe, therefore, that it is important to couple BPEL with a model for expressing such authorization policies and constraints, and a mechanism to enforce them. It is also important that such authorization model be high-level and expressed in terms of entities that are relevant from a modeling and organizational perspective.

A partial solution to these issues is the BPEL4People extension (as discussed in the next sub-section) which deals with workflow tasks that require people involvement – as opposed to other computer systems. A more complete solution would be to manage security-related authentication and policy information using existing BPEL primitives – through explicit data assignments, explicit conditional logic, explicit invocations to security infrastructure services, etc. While feasible, this solution is not very satisfactory because it clutters the process definition with many imperative statements that could otherwise be specified in a declarative fashion and thereby reducing one of the principal value of BPEL which is to separate process logic from other concerns in order to achieve clearer and simpler process models.

Thus, if BPEL processes are to conduct secure transactions that span not only people but also web services, a security model is required to be integrated with BPEL that is more general than the one offered by BPEL4People.

Further planned steps

We propose the following development plan as related to the BPEL execution engine:

- Explore and define bindings of the TAS[3] security model as extension to the BPEL 2.0 language, where appropriate;
- Implement and integrate these extensions to Apache Ode and Intalio|BPMS.

BPEL4People Workflow Model

Tempo

Tempo (see (5)) is an open-source project to provide a workflow implementation based on the BPEL4People architecture. Tempo is the workflow engine underlying Intalio|BPMS.

As defined by BPEL4People, Tempo supports role-based interaction of people, it provides means of assigning users to generic human roles, it can delegate ownership of a task to a specific person and supports use-cases such as separation of duty, nomination, escalation and chained execution.

BPEL4People Compliance

The Tempo project was started shortly after the original BPEL4People whitepaper (see (4)) was published but before the first BPEL4People specification draft became available. Due to this parallel evolution, there are some differences between the two. Here is a high-level summary of these differences:

- As a conservative approach, Tempo's design was based on what is referred to as "Constellation 4" in the BPEL4People specification. In other words, it relies on the standard BPEL <bpel:invoke> activity instead of the <b4p:peopleActivity> and therefore implements its own protocol between the process engine and the task management services (i.e., WS-HumanTask implementation);
- Tempo uses standard BPEL data assignment activities to manipulate people and role assignments instead of extending the <bpel:assign> activity;
- Tempo does not use process-related human roles such as the <b4p:processInitiator>, <b4p:processStakeHolders> and <b4p:businessAdministrators>;
- The task lifecycle in Tempo is extensible and composed of 4 basic states: ready, completed, failed, claimed. BPEL4People has 5 basic states: running, completed, failed, terminated, and obsolete. In Tempo, the terminated and obsolete states are handled using the failed state and an explicit reason.
- Ad-hoc attachments are associated with the current task instead of being associated with the BPEL process that created the task. Ad-hoc attachments may be propagated through a workflow using standard BPEL data assignment activity;

In addition to standard BPEL4People capabilities, Tempo provides:

- A comprehensive role-based access control (RBAC) security framework that integrates with most Lightweight Directory Access Protocol (LDAP)-compliant directory servers;
- Integration with multiple web-based user-interface technologies such as Xforms and Intalio|RIA;

- A user-friendly task list for workflow participants that displays outstanding tasks, notifications and allows the kickstart of processes using people-initiating forms;
- Integration with Java Portlet technology (JSR-168) to display the task list in an integrated portal such as Liferay Portal.

Further planned steps

Our plan with respect to BPEL4People is as follows:

- Implement the WS-HumanTask (see (6)) specification within Tempo; this implies reconciling the differences mentioned above;
- Extend the Apache Ode engine to implement the BPEL4People specification, most notably the <b4p:peopleActivity>;
- Extend the Tempo security framework to support federated identity integration, which is a requirement of the TAS3 infrastructure;
- Integrate Tempo with one federated identity system, such as OpenID.

Secure Processes

This section contains requirements that are motivated by examples derived from the Kenteq APL process (see also the project deliverable D1.2 and D1.4 for further details). As the description of that process is not yet known in full detail, some changes might become necessary later on.

Distributed identity management

Actors in business processes need to be authenticated. This authentication must be based on a distributed identity management system (federated identity), because business processes in a service-oriented architecture comprise actors from different organisations. It is desirable that they can use a single account to participate in these processes, issued e.g. by their employer or a national registry.

The Kenteq APL process, for example, spans the employability agency, the company of the candidate and the coach and assessor, who might be independent consultants.

The distributed identity management framework itself is outside the scope of WP3. The workflow management engine needs a component that can check presented credentials.

Flexible user/role assignment

The choice of actors in workflows can be based on different, application-dependent criteria. It should thus be possible to explicitly model the assignment of users to workflow tasks and workflow roles.

In the Kenteq APL process, this is necessary for the assignment of the coach and assessor roles: They cannot be chosen by a simple role mapping. Instead, special skills and qualifications need to be taken into account, either by a human or automatically by the process logic.

In Ode and Intalio|Tempo, it is currently possible to flexibly assign persons to tasks at runtime by setting the respective field in the call to the Task Management Service. However, this is only done on a per-task basis, which requires error-prone repetitions when a person is assigned to a workflow role performing several tasks. The next section will explore several possible remedies. As a solution, the introduction of workflow roles is considered in the next chapter.

Role mapping

A simple way of assigning users to a task is the mapping of workflow roles to organisation roles managed in a suitable directory.

This is applicable to the *Manager* role in the Kenteq APL process.

From an architectural point of view, this directory is part of the authentication and authorisation infrastructure. In the workflow engine, only a component interfacing with the directory is needed. Intalio|Tempo already has an extensible architecture regarding authentication mechanisms. Thus, the integration of the interface component will not pose serious problems.

Delegation of authority regarding data sources

In order to fulfill their tasks in a workflow, actors need access to certain data relevant to the workflow. Thus, it must be possible to grant them access to data sources when they are assigned a role or task in the workflow.

In the Kenteq APL process, this applies, e. g., to the coach and assessor. They must be granted access to the candidate's portfolio in order to coach or assess the candidate. This delegation of authority will be caused when they are assigned to their roles, or at some other suitable time during the execution of the process. Likewise, the delegation will be revoked when they do not need access any longer because they have completed their task.

Thus, data repositories must provide an interface (service call) that allows delegation of authority, including the revocation of such delegations. Further, the scope of delegation policies and a proper format for such policies must be defined, and delegation policies must be enforced. Processes executed by the process engine are simply users of this interface.

A possible mechanism to automatically align the assignment of people to roles with the delegation of the corresponding authorisations require further examination.

Delegation of workflow roles and tasks

Actors in workflows might not be able to fulfil their tasks for different reasons, e. g. absence or high workload. In such cases, they should have the possibility to delegate their tasks (and accordingly, the permission to perform the task).

The granularity of such delegation decisions has yet to be determined. For the time being, it is planned to support the delegation of single tasks. This involves changes in the workflow engine and its user interface to perform the delegation (i.e., changing the user to which the task is assigned), and securely logging the delegation action.

Separation of Duties

To avoid conflicts of interest (thus, ultimately for security purposes), it is sometimes required that several people cooperate to achieve a specific result (e.g., requiring to signatures of different persons to cash a cheque). This is usually done by requiring that several tasks that are all needed to achieve the desired outcome may not be performed by the same person and thus the corresponding workflow roles may not be activated by the same person for the same workflow instance (dynamic separation of duties) or that a person may not hold two roles at once (static separation of duties).

Dynamic separation of duties is an issue in the APL process, as well: There is a conflict of interests between coaching the candidate and assessing his performance at the end of the coaching, so these tasks (and consequently, the roles of coach and assessor) should not be performed by the same person.

SSoD is a problem that would have to be addressed in the assignment of roles to users, i.e. in the user directory.

For dynamic separation of duties, in our context we must enforce that conflicting roles are not activated inside one workflow instance (instead of user sessions). This is similar with the view in *Multisession separation of duties* (Chadwick et al.), where separation of duties is guaranteed for business contexts. Workflow instances can be seen as a special case of business contexts.

DSoD, which is limited to the scope of a single workflow, can be implemented in the process definition itself (by explicitly modelling the assignment of people to tasks), but this may be error-prone for large process diagrams or complex SoD constraints, as the constraints themselves are not explicitly visible. Thus, a better option is to explicitly specify the SoD constraints and enforce those explicit constraints during execution. This requires a SoD constraint editor in the modelling tool and an SoD enforcement component in the workflow engine, i.e. Intalio Tempo. For this, the task management in Tempo needs to be enhanced so that it can take the context of a whole process into account. Furthermore, a format for SoD constraints needs to be specified.

Binding of Duties

Binding of duties is, in a sense, complementary to separation of duties: Where separation of duties requires two activities to be performed by different persons, binding of duties requires them to be performed by the *same* person.

For example, the person that enters data and is asked to complete that data if it has been found incomplete by someone else should be the same.

Sticky policies

The TAS³ architecture will define policies that are attached to and travel with data items. Depending on what obligations these policies may impose on data processors in the TAS³ eco-system, there might be a need to integrate checking and enforcement of such policies in the process execution engine.

As the definition of sticky policies is currently not detailed enough, the implications on the design of the process management platform remain to be determined.

Reactions to security violations

TAS³ as a trusted and secure architecture will perform various security checks at runtime. When these checks discover security violations, processes must be able to react and recover. These recovery mechanisms must be specifiable in a flexible and systematic way, in order to be able to react to unforeseen security violations and to keep process definitions concise.

Security violations can happen in all processes, e.g. because of changed permissions. In the Kenteq APL process, this might be the case for operations on the candidate's portfolio.

Automatic security check before process execution

It has been proposed to provide mechanisms that automatically determine immediately before the execution of a process whether security permissions are sufficient to allow the process to complete successfully. However, it is hard to impossible to algorithmically determine whether a specific non-trivial property holds for an arbitrary programme (cf. Rice's theorem). Consequently, such a feature needs to be defined more narrowly. A more specific definition of the security infrastructure is needed for this.

However, some requirements to other components or the general architecture that would be necessary in order to implement a solution have been identified:

During the check phase, for all services that will be called later during the execution it has to be determined whether that (later) request will succeed. This will take the form of a 'test request'. However, at the time the test request is performed, the actual parameters of the call are not yet known. Thus, the parameters of the test request will be a generalisation of actual parameters. Of course, these generalised parameters might not be sufficient to decide finally whether the request will be allowed, so the answer might take the values "Yes", "No" or "Maybe".

Existing research has already partly explored the problem: (1) develop necessary and sufficient conditions for the set of constraints of a workflow authorisation schema such that a sound schema is ensured, i.e. the process for every user or role authorised to perform a task, there is a complete workflow instance where this user or this role performs this task.

Changes of permissions during execution

Processes handle various kinds of data, including personally identifiable information. The "owners" of this information should be able to change permissions on this information at any time, i.e., even while a process instance is running. These changed permissions should then be respected and no further processing of the involved should take place.

In the Kenteq APL process, e.g., the candidate might want to revoke permissions on some documents (e.g. a diploma) that he has already added to his portfolio.

It has been agreed not to pass data, especially personally identifiable information, around through the process management engine. Instead, data should be stored in secure repositories and loaded from there on each access. Thus, access can be securely logged for auditing purposes. Instead of the data itself, vouchers will be passed, which enable the recipient to retrieve the data from a repository. The repository then is the authoritative instance where permissions to the data are managed. Thus, if permissions are revoked or additional permissions are added, these changes immediately take effect for processes accessing the data.

Process Adaptation

In order to enhance adaptability of business processes (workflows) existing solutions, mostly supporting adaptations of process models, are not sufficient in TAS^3 applications (see also the project deliverable D1.1 containing an overview about the state of the art). There are also interesting approaches to handle adaptation of process models in a very flexible way, but basic support is already contained in existing BPM systems or there already exist (frequently only formal) concepts. But there is also significant need for further research, e.g. in order to get more user support or to support cooperation of business processes that is defined by business process choreography (see e.g., (9), (10)).

In TAS3, we need to improve the flexibility of processes especially for already active instances. Adaptability in our context is much more than following alternative paths in a fixed process model.

Distinct levels of flexibility determine the following rough process categories:

- Ad Hoc Workflows: without fully modeled process schema
- Workflows in an open dynamic environment: have to be adaptable to invocation of which web service or other process and to the use of involved data, and to security rules
- Production Workflows, not very flexible but very robust.

Adaptation of processes comprises at least the following issues:

- Change process schema (process model)
- Basic structural changes, e.g. replace tasks like invoked web services or subprocesses, delete tasks, insert tasks
- Change of process flow, e.g. alter branch conditions, use different data, insert human tasks, alter process flow as result of the active role or process data
- Change cooperation with other business processes
- Apply adaptations to process instances, i.e. migration
- Execute adaptation automatically, semi-automatically or manually

In order to achieve adaptability of active process instances, modeling of adaptations is needed in order to formalize process status and define well-formed processes. We need adaptation operators, and have to migrate process instances to a changed process model. Also, we need to identify the influence of adaptations to security specification of the process and manage it

Existing processes like the Kenteq APL process, which we are modeling in chapter 4 as a first process model of one of the project scenarios, is not adequate to analyze requirements for adaptability, because the process is in practical (commercial) use and there do not exist sufficient adaptation power in existing systems. Therefore we take our requirements out of previous analysis which was conducted at University of Karlsruhe (9).

Security specifications of process models provide means to define who can access what model and how can these been adapted and deployed. This knowledge helps to achieve more flexible and dynamic processes, which are required in open networks of information services. Achieving a better automation and/or user support, semantic information about services and processes is relevant. For that, we need adequate ontologies of the application area.

Furthermore, the adaptation process has to observe the underlying security characteristics of sub-processes and services. The activities to reach this result are to define and implement concepts for process adaptation. In a first step we concentrate on structural adaptations and in a further project phase the concept will be enhanced with more elaborated, semantic adaptations. We need to explain the link between the process model to be written and existing specifications and have to define how live instances can be adapted and migrated.

Structural adaptations concern the change of the structure of the business process, i.e. in the simplest cases to replace activities or tasks by more adequate ones depending on the process execution context, or to remove activities. That may be enhanced by allowing to replace activities by subprocesses or even replace some part of a process by a new subprocess. The process adaptation consists of two main steps. First to define the new process model, i.e. to provide an adapted process schema, second to migrate the running process to this new structure. Using web services only the substitution of single web services by other web services will be supported, if the concept of abstract web services is used in the process model. Further adaptation needs new mechanisms and concepts to take care of the actual process status and to investigate if process changes will be allowed.

High-Level Ontology covering Business Processes

Web services are designed to access applications over a network, such as the internet, and execute these applications on their respective hosting servers. The World Wide Web Consortium (W3C) has developed an XML format, called the *Web Services Description Language* (WSDL) (1), to define abstract operations and messages provided by such services.

Although WSDL documents describe the input expected by a service as well as its output, they lack a definition (i.e. a semantic) of what is actually meant by these inputs and outputs. The Semantic Web (2) has been proposed as a solution to this problem by annotating content with entities (i.e. concepts, properties and instances) from ontologies. An *ontology* is commonly defined as: "*a [formal,] explicit specification of a [shared] conceptualization*"[2] (3). More specifically, an ontology explicitly defines a set of entities (e.g. classes, relations and individuals) imposing a structure on the domain that is readable by both humans and machines. A variety of representation schemas have been proposed to encode this ontological conceptualisation, such as CML, CycL, KIF, Loom and the Web Ontology Language (OWL).

A business process defines a collection of tasks performed in a given order to achieve a desired objective. The inputs and outputs can either be documents or information generated by human actors, other Web services, or a combination of both. Thus, a business process ontology describes the relationship between activities and the role of the participants (e.g. actors, data) in these activities in a meaningful manner. However, this ontology does not include knowledge about the sequencing of activities.

[2] Gruber's original version is without the words "formal" and "shared", which nowadays are accepted to describe more precisely the intention of ontologies.

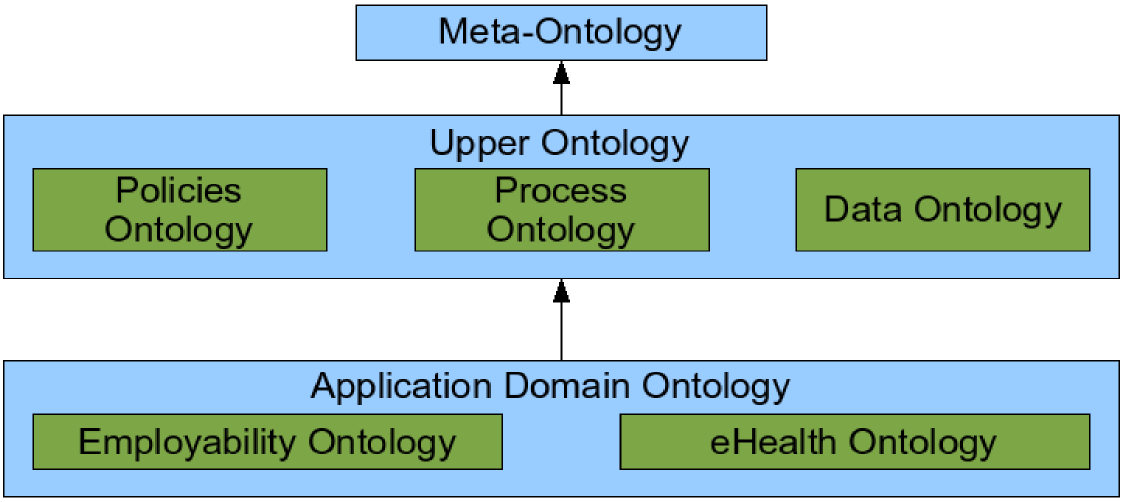


Figure 2: Ontology architecture

The *meta-ontology* (Figure 2) introduces generic concepts that are shared by all lower-level ontologies and reflects the underlying nature of business processes, whereas the *application domain ontologies* capture the knowledge about the domain regardless of specific applications. For example, the concept **Actor** is defined in the meta-ontology as any type of participants (i.e. human, Web service, data) taking part in a process. Existing meta-ontologies that are stable already exist that can be reused here (see e.g., (4)). These ontologies are formatted in open W3C standards and can be found in the area of semantic business process management (for example in X1). Given the diverse types of applications that are sought to interoperate via business processes, the challenge lies in agreeing on the contextualisations of these generic concepts in the upper ontology.

We have identified three generic ontologies; namely the *upper policies ontology*, the *upper process ontology* and the *upper data ontology*. The upper policies ontology describes the context, such as the identity of an actor, his/her credentials, the role of the actor in the service, to which a service is subject. In addition, it defines how information is exchanged securely between services. Suppose a school request access to personal data of a prospective student (e.g. his/her previous scholar achievement), then the service receiving the request (e.g. Kenteq) needs to verify that the school has the appropriate authorisation to access these data.

The upper process ontology describes processes and activities in an abstract manner in terms of:

1. Assumptions, i.e. the initial state of the activity or process.
2. Pre-conditions, i.e. the state of the process prior to its execution.
3. Post-conditions, i.e. the state of the process after completion.
4. Effects, i.e. the state of the world at the end of the process or activity.

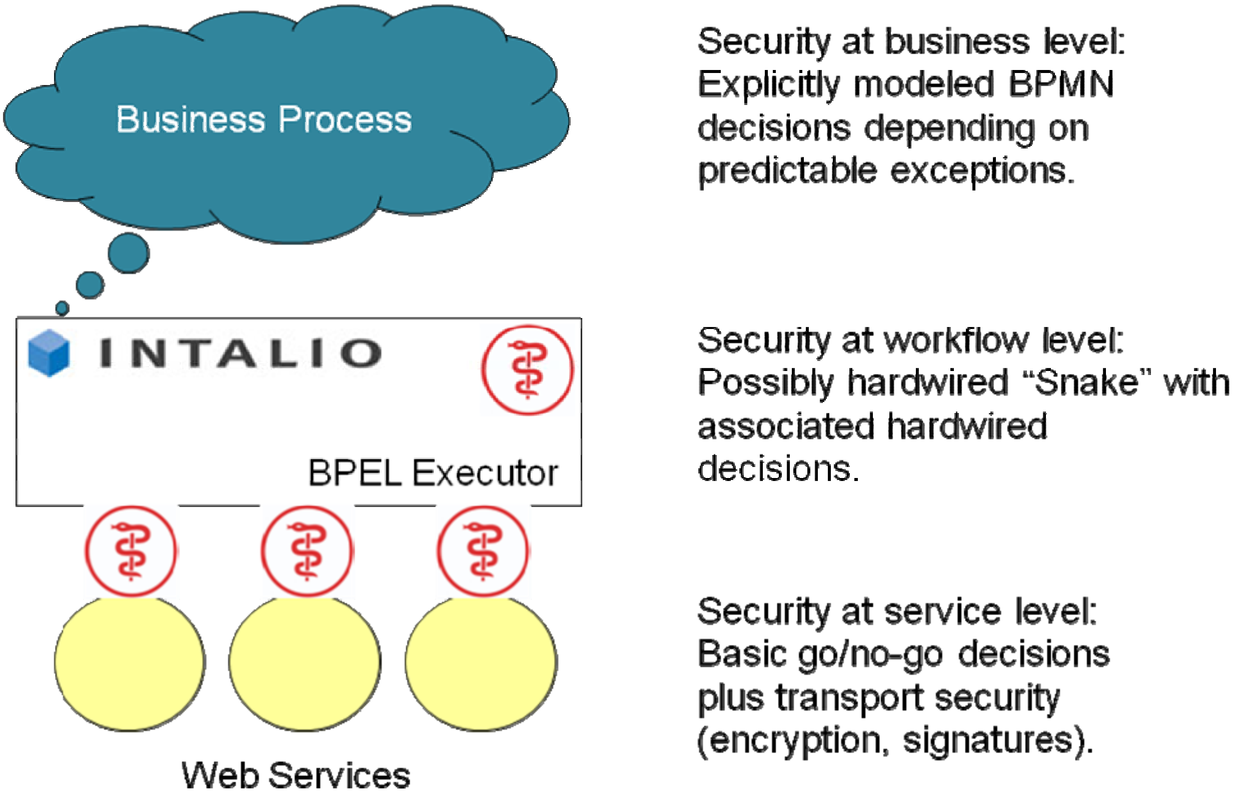
It should also include typical workflow patterns expressing the relation between the different elements. For example, the statement `follows(activity1, activity2)` declares that `activity2` should be processed after `activity1`. These generic processes will then be used to generate processes specific to an application domain.

The upper data ontology contains the type of (i) data structures and (ii) the processes that create or consume instance data.

The capability of Web services augmented with semantic markups can enable different tasks within WP3. On the one hand, concepts in the ontology can be used to compose request to Web services. Suppose the “*send the certificates of candidate C to X*” (C is an instance of **Candidate** and X is an instance of **Assessor** OR **Manager**) has been sent to a Web service. The constraint on X means that we need to check that it is the actual assessor of C, but also that it has the role manager. On the other hand, Web service composition states the order in which messages are exchanged between services to generate a process model. The process model is composed of a set of preconditions (i.e. input parameters) and a set of effects (i.e. output parameters) declared in an explicit semantic (i.e. with concepts from the ontology).

Integration of Adaptable Secure Business Processes into the TAS^3 Architecture

From an integration point of view, there are two basic positions in the TAS^3 Architecture where Business Processes and the associated modeling tools come into play: at the secure transaction level, and at the application level. In Figure 3 the interaction between the business process management platform and the TAS^3 security levels. The applications are contained in business processes, the secure transaction level of the TAS^3 system is represented by the snake symbol.



Security at business level: Explicitly modeled BPMN decisions depending on predictable exceptions.

Security at workflow level: Possibly hardwired “Snake” with associated hardwired decisions.

Security at service level: Basic go/no-go decisions plus transport security (encryption, signatures).

Figure 3: TAS^3 three-tiered security of business processes

The secure transaction architecture of TAS^3 consists of a collection of components (which likely will act as services) which are orchestrated by a well-defined, finite state diagram that can be documented as a process. The nature of this process is such that it can be exhaustively implemented using a tool such as Intalio|Designer, and subsequently can be executed. For early testing and process validation, this execution can be controlled by the Intalio|Server and be monitored by Intalio|Workflow. The call-outs to all underlying services can be truly executed as well. However, for production, this would be too slow and cumbersome (comparable by implementing Ethernet packet handling as part of application code). For production, a hard-wired implementation will be required, but the Intalio|Server variant will be very valuable for evaluation and fine-tuning.

At the application level, it is obvious that the business process needs to be made security (TAS^3) aware. This can be done in three ways: at the application level, at the BPEL executor level, and at a system level. Application level integration means that all details of the security handling and (more importantly) all security exceptions need to be modeled by the business analysis. This will very quickly become unmanageable, and cause unnecessary complexity and redundancy in nearly every business process. BPEL executor level integration means that the BPEL engine, such as Intalio|Server, gets hard-wired extensions to support security callouts and the associated exception handling. System level integration means that a layer below the business/application level intercepts the calls generated by the workflow and wraps them into secure transactions. This last level is extremely complex and reserved for true legacy integration attempts only.

The results of the secure transaction architecture hard-wired implementation can be extended in a straightforward way to fill in the BPEL executor level integration gap. This leads us to a roadmap for the integration:

1. Exhaustive modeling of the TAS^3 Secure Transaction Architecture in BPMN (from WP2).
2. Implementation of the full Secure Transaction Architecture (STA) in BPMN, leading to executable BPEL that calls out to all relevant services (or stubs when the services are not available yet) (with WP8).
3. Parallel recoding of the STA in the BPEL executor and inserting the BPEL STA as subprocess in a chosen business process (from WP9).
4. Exhaustive tests and comparisons of the two approaches in 3. (with WP12).
5. Decommissioning of the BPEL subprocess STA in favour of the hard-coded BPEL executor variant in all instances of Intalio|Server used in the project.
6. Investigations of the possibilities to add STA plus the missing BPMN elements to existing systems that can only be accessed at the system level, i.e. not at the business process or transaction level.

4 Conceptual Design

High-Level Ontology covering Business Processes

STARLab‘s DOGMA (Developing Ontology-Grounded Methods and Applications) ontology framework is based on database semantics and model theory (5). Its *double articulation* principle and grounding in *natural language representation of knowledge* makes DOGMA particularly fit for representing business-level as well as technical terminology and semantics typically found in business process models and their web service implementations respectively.

According to the double articulation philosophy a DOGMA ontology consists of an ontology base of *lexons*, which holds (multiple) intuitive conceptualizations of a domain, and a layer of reified ontological commitments. The latter essentially are views and constraints that within a given context allow an application to commit to the selected lexons. Contexts group commitments allow ontological patterns to be represented and compared at various levels of granularity. In this way, scalable ontological solutions for eliciting and applying complex and overlapping collaboration patterns can be built.

Another fundamental DOGMA characteristic is its grounding in the linguistic representation of knowledge. This is exemplified most clearly in the linguistic nature of the lexons, with terms and role strings chosen from a given (natural) language, and that constitute the basis for all interfaces to the ontology. Linguistic "grounding" is achieved through *elicitation contexts*, which in DOGMA are just mappings from identifiers to source documents such as generalized glosses, often in natural language.

DOGMA-MESS (meaning evolution support system) (first introduced in (6)) is a methodology based on DOGMA that involves the *owning* community in defining and evolving its common ontology *collaboratively*. Doing so, it advances state-of-the-art single-user ontology management approaches that are typically found in literature (7).

DOGMA-MESS' collaborative ontology evolution process is driven by the current semantic interoperability requirements that emerge from the community dynamics. For example, semantic interoperability requirements identify the conceptions that are needed in the common ontology in order to ensure *semantic interoperability.* Semantic interoperability is interoperability (as defined in ATHENA (8) and INTERTOP (9)), e.g., data exchange between web services underlying inter-organisational business processes, by means of an ontology that is shared by the involved information systems (10).

DOGMA-MESS ontology evolution process defines a cycle consisting of four main phases (11):

1. Community grounding: The core domain expert interprets the domain and identifies the conceptions that are needed to enable semantic interoperability. This leads to the identification of a meta-ontology framework. The resulting *community representation* is an initial common ontology that includes the conceptions identified in semantic interoperability requirements, and that is hooked into the chosen meta-ontology framework.
2. Perspective rendering: All stakeholders interpret the semantic interoperability requirements and the initial common ontology, and render stakeholder perspectives. The result is a set of subjective representations that contextualise the conceptions identified in the semantic interoperability requirements. Perspective rendering can be formally restricted by *perspective policies* (see (12)). Furthermore, these perspectives are stored and versioned in the DOGMA ontology framework.
3. Perspective unification: All stakeholders interpret the set of stakeholder perspectives and collaboratively agree on a unified perspective on the initial ontology. The result is a minimal

inter-subjective representation of the conceptions identified in the semantic interoperability requirements that are socially accepted. The initial ontology is updated with the unified perspective accordingly.

4. Community Commitment: All stakeholders interpret the new ontology version and commit their applications (e.g., web service interfaces or business process models) to it. The result is a formal representation of the semantic interoperability in the community.

This four-phase cycle is repeated until satisfying consensus on the relevant conceptions is achieved with minimal effort. The explicit rendering of stakeholders perspectives allows us to capture the ontology evolution process completely, and validate the ontology against these perspectives respectively. Ultimately, co-evolving communities with their ontology will increase overall stakeholder satisfaction. This makes DOGMA-MESS convenient for this project given the continuously evolving business processes within and between stakeholders.

Tool support for DOGMA-MESS is reported in (13)and (14).

Process Modelling

Defining a clear methodology on how business requirements and procedures are translated to ready to be executed processes is clearly on the goals that the TAS^3 Consortium is committed to deliver.

Business Processes have become widely popular recently thanks to the adoption of key standards such as BPMN and the maturity of technologies. However it is important to keep in mind that Business Process Management is first a methodology that is enabled by technologies and not the other way around. Thus Intalio has decided to appoint process experts in order to transfer knowledge and train the different TAS^3 members on the fundamental concepts for Process Modelling. Instead of illustrating concepts with case studies, it has been decided to focus on a real world process provided by Kenteq: the Kenteq APL process that describes the ‘Performance Review’ procedure that each employee has to go through annually.

The first objectives are as follow:

- How can you identify business requirements?
- How do you collaborate to refine scenarios and use cases to support the business requirements?
- How to best leverage BPMN (Business Process Modelling Notation) to design business processes?
- Process Modelling Best Practices: what are the industry practices?
- How to make the difference between a process and an application?
- How to ensure that the processes modelled are ready to be executed?

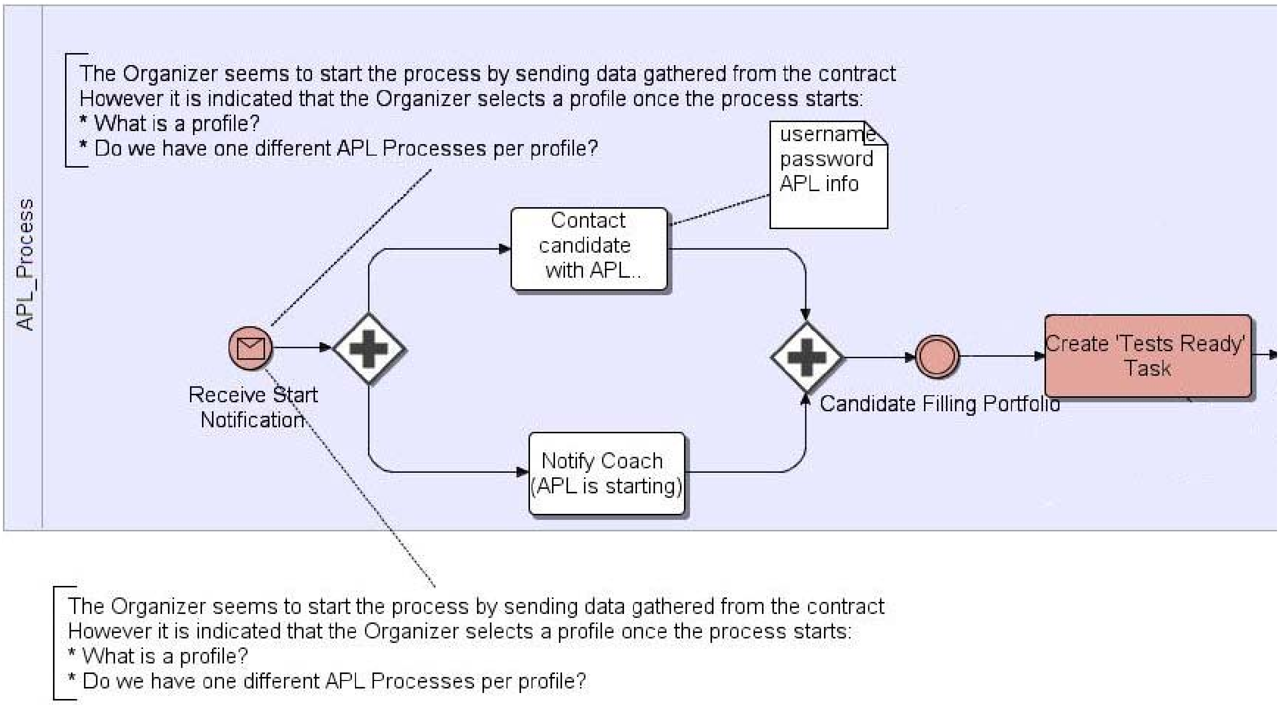

APL_Process
The Organizer seems to start the process by sending data gathered from the contract
However it is indicated that the Organizer selects a profile once the process starts:
* What is a profile?
* Do we have one different APL Processes per profile?
username
password
APL info
Contact candidate with APL..
Receive Start Notification
Notify Coach (APL is starting)
Candidate Filling Portfolio
Create 'Tests Ready' Task
The Organizer seems to start the process by sending data gathered from the contract
However it is indicated that the Organizer selects a profile once the process starts:
* What is a profile?
* Do we have one different APL Processes per profile?

Figure 4: Start of the BPMN Model of the Kenteq APL Process

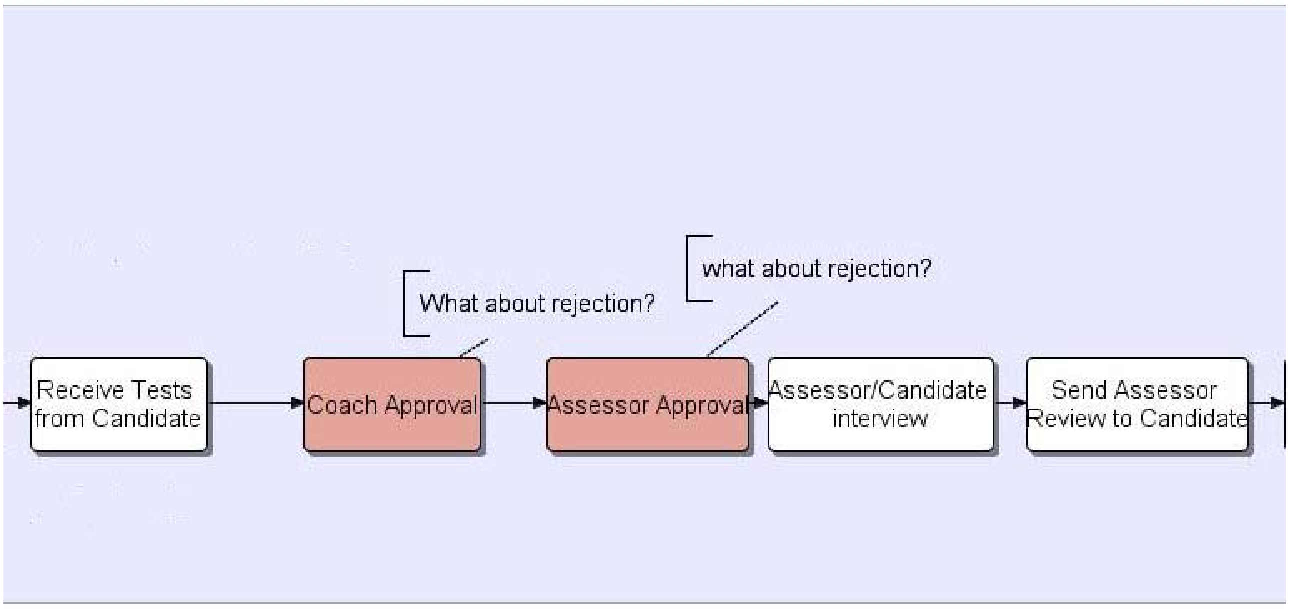

What about rejection?
what about rejection?
Receive Tests from Candidate
Coach Approval
Assessor Approval
Assessor/Candidate interview
Send Assessor Review to Candidate

Figure 5: BPMN Model of the Kenteq APL Process – part 2: coach and assessor tasks

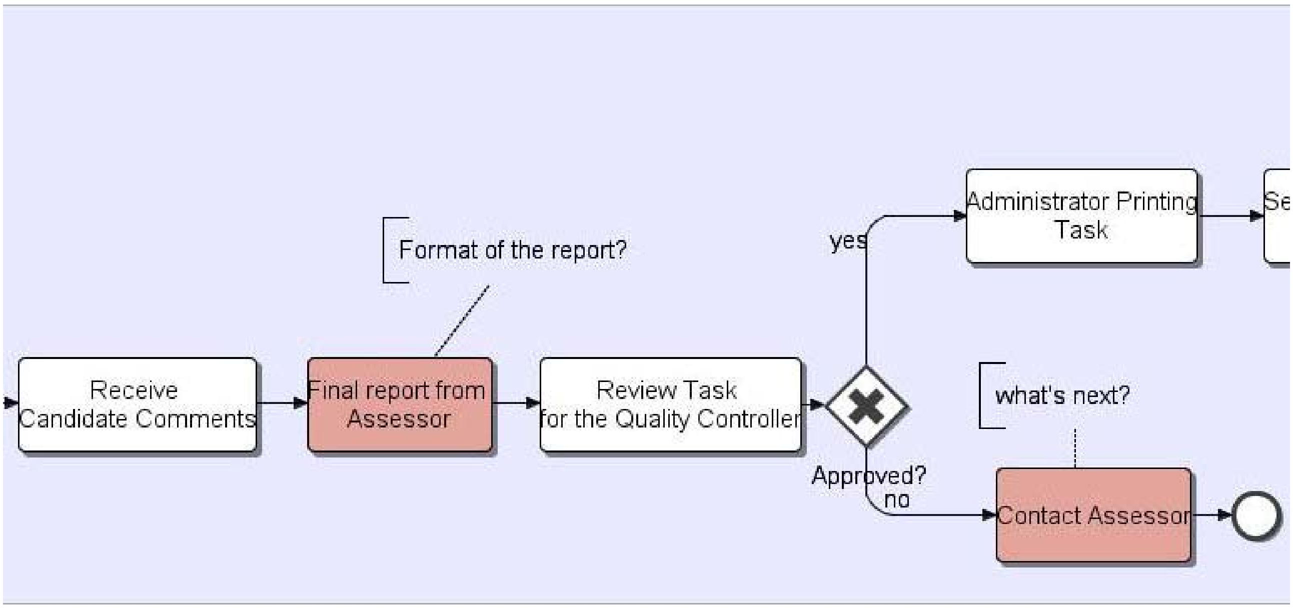

Format of the report?
yes
Administrator Printing Task
Se
Receive Candidate Comments
Final report from Assessor
Review Task for the Quality Controller
what's next?
Approved?
no
Contact Assessor

Figure 6: BPMN Model of the Kenteq APL Process – part 3: assessor, quality controller

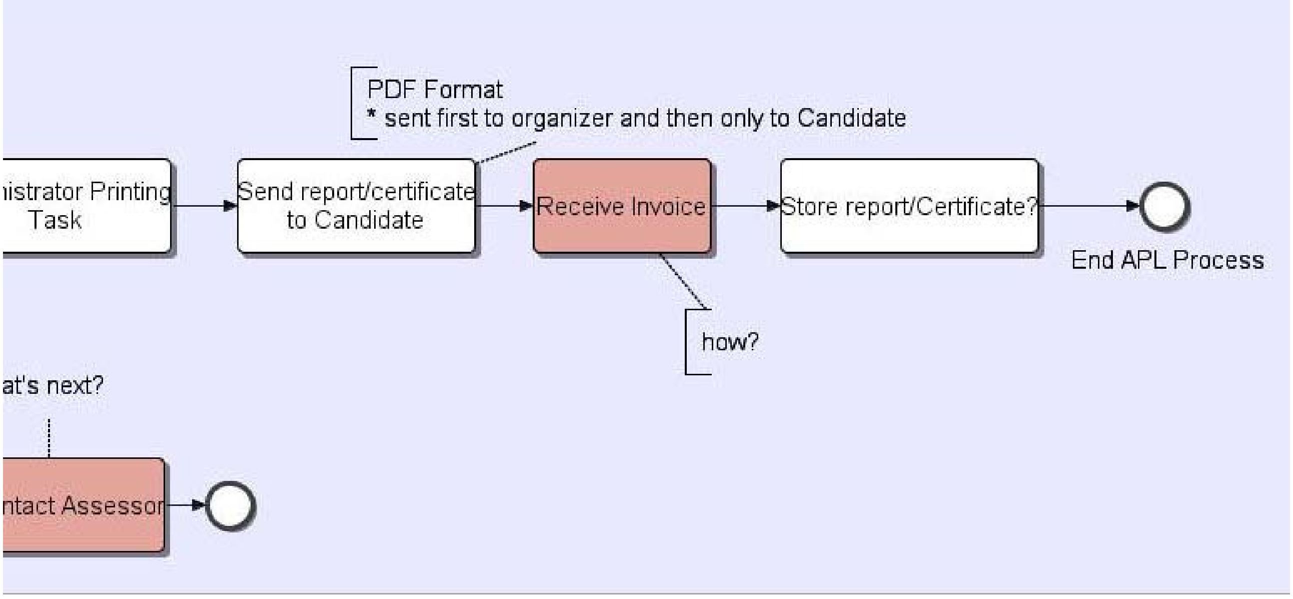

PDF Format
* sent first to organizer and then only to Candidate
iistrator Printing Task
Send report/certificate to Candidate
Receive Invoice
Store report/Certificate?
End APL Process
how?
at's next?
ntact Assessor

Figure 7: BPMN Model of the Kenteq APL Process – part 4: end of the process

During this exercise, the participants will soon realize that BPMN is a very advanced language and Intalio|BPMS can support most of the real life in-company processes as well as you quickly find limitations when you start to deal with inter-company processes where you have to share part of your business processes. Security becomes then a major concern and it is important to capture the security concerns at the Process Modelling stage.

Modelling the Kenteq APL process in the context of the TAS[3] project will allow to progress on the following tasks:

- Visualization of security specifications at the process design level

Several specifications exist today to address security concerns at the runtime level (WS-Security/SAML). However there is no method to easily expose those security concerns visually to the Process Analysts. BPMN will need to be annotated in order to provide the right information.

- Human Activities

It is crucial to define an assertion mechanism to identify the role that can be involved when interacting with other participants (IT Systems, processes or human). BPMN will be annotated once again to provide the right information

Intalio process experts and product engineering team will provide extensive help to refine the requirements and ensure that the correct annotations are created in Intalio|Designer.

Secure Processes

Process-aware task management

Currently, the Intalio|Tempo workflow engine manages each task on its own, i.e., no connections between different tasks of a process are taken into account. A view spanning more than one task of a process, however, is necessary e.g. to support authorisation constraints involving more than one task. Such constraints include (cf. above) separation of duties and binding of duties. Another related requirement is the explicit (manual or automatic) assignment of specific persons to workflow roles (not single tasks). This involves handling the concept of workflow roles and will be treated in more detail in its own section below.

There are different possibilities to implement this:

- The first possibility is to leave the execution level as it is and implement this feature solely on the modelling and transformation level. The modelling tool needs to be enhanced by a component to specify authorisation constraints. The component generating the executable BPEL process must then take these constraints into account and generate BPEL code that explicitly assigns people to tasks while respecting the constraints.
- The second possibility is to evaluate the authorisation constraints in the Tempo workflow engine. Again, an enhancement of the modelling tool is needed. The constraints are then deployed to Tempo together with the task descriptions, which then takes them into account when creating tasks and assigning them to people.

The first possibility would require generating lots of code for the evaluation of the constraints, which would make it difficult to trace the runtime behaviour of BPEL processes. Furthermore, the BPEL execution would need access to the people directory, which is currently not the case. The second possibility is a natural fit because all other user authentication and assignment (authorisation) functionality is currently performed by Tempo, the human workflow engine.

The "task metadata" transferred to the Tempo engine on the creation of a task already includes a `processID` field. This field can be used to provide information about the context of the task instance, i.e. which process instance it belongs to, to the TMS (Task Management Service). Furthermore, Tempo must be enhanced to be able to link process instances to process definitions (and the corresponding authorisation constraints which will be deployed into the Tempo engine at the same time as the process itself is deployed to the BPEL execution engine).

Concept of workflow roles

Workflow roles are a concept for grouping the responsibility for performing several tasks. Thus, this group of tasks can be assigned to a person in one step, the same holds for delegation of the assignment. It should be noted that workflow roles are a different concept than organisational roles: Workflow roles are activated for a specific instance of the workflow only, while organisational roles are more static. Thus, the concept is natural and desirable.

Currently, the roles given as attributes of a task specify a mapping to organisational roles. The possibility for such a mapping is still desirable. Furthermore, the introduction of workflow roles must not cause incompatibilities to existing process models.

We have the following transition in mind: For the task meta-data, a new attribute `workflowRole` is introduced. When this field is present, assignment of the task is based on the assignment of the role, otherwise the existing `userOwner` and `roleOwner` fields are used. Workflow roles are defined in an annotation to the process as a whole, which is deployed just like authorisation constraints (cf. prior section). This annotation might also specify a mapping of workflow roles to organisational roles. The definition of workflow roles implies binding of duties. It might also specify a delegation policy for the workflow role (cf. below).

On the user interface level, the task-list will be extended to provide the user with information about the workflow roles he has been assigned.

Delegation of workflow roles

Intalio|Tempo currently allows *forwarding* (reassign a task to another user or role) and *delegation* (a user can name a delegate to process tasks on his/her behalf). However, these possibilities leave a wide gap in granularity, between re-assigning only a single task, and re-assigning all tasks in all processes. With the concept of workflow roles, however, it is possible to delegate the involvement in one specific workflow instance, which is an intuitive level of granularity.

The user interface for delegation will be integrated into the workflow console, where the user can delegate his assigned workflow roles to others and list his current delegations.

Policies for assignment and delegation of workflow roles

Assignment and delegation of workflow roles to users are closely related operations: Role delegation is a special kind of role assignment where the role is already assigned to the delegator.

Assignment and delegation will be restricted by policies. Such restrictions can include the organisational roles of the assignee/delegatee, the role of the delegator, the delegation depth for chained delegation, or application-dependent attributes like skills or qualifications.

The deployment process is the same as for other authorisation constraints (cf. above), as delegation policies relate to definitions of business processes.

Reactions to security violations

This requirement will be fulfilled by the well-known event handler concept: Security events discovered by the application-dependent policy evaluation point (ADPEP) will be passed to the process execution engine, which raises a BPEL event. This event will cause the execution of an event handler if one exists. Event handlers can be modelled in the BPMN modelling tool and will be translated into BPEL event handlers.

For integrating ADPEP into the process execution engine, several solutions have been proposed. In a first step, ADPEP functionality will be integrated into the process definitions themselves. Later on, the ADPEP will be integrated into the Intalio server or implemented as a stand-alone component on the operating system’s level.

Process Adaptation

In order to support adaptation of business processes also for running process instances we focused in the report period on structural adaptations. The intention and goal of the first project year was to provide basic mechanisms for later developing more elaborated adaptation facilities.

For that, we have investigated a formal model of BPMN processes adequate to catch the process status and how to specify adaptations. For process adaptation we follow the steps:

- Adapt the process by using change operations on BPMN level and workflow pattern libraries
- Transform the change operations and changed workflow to the execution level by
    - Mapping to BPEL (on a schema level, e.g. using the Intalio mapper)
    - Migrating process instances: investigate the state of the instances, select concerned instances, and adapt the particular running process instances.

Structural process adaptation will be supported by a library of process patterns. There will be proposed alternatives for structural changes and if possible automatic adaptations are supported. The adaptation architecture is shown in Figure 8 with modelling and execution levels.

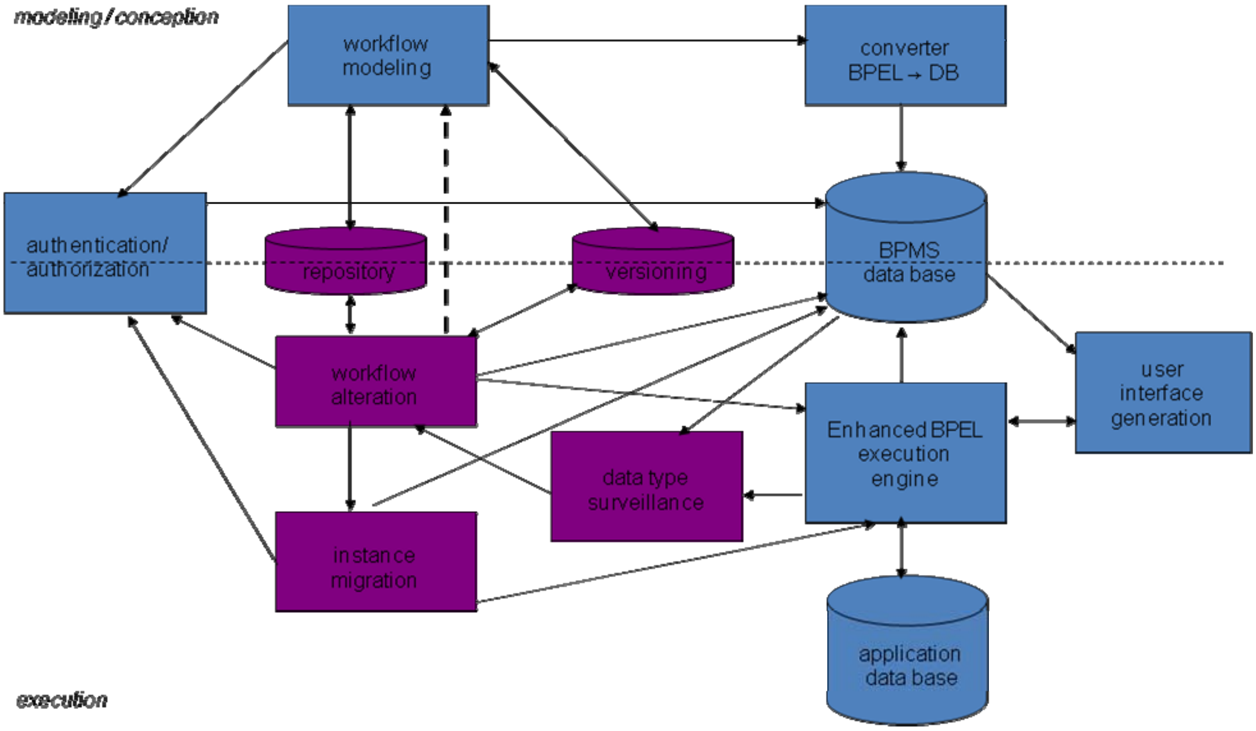


Figure 8: Adaptability architecture of a WFMS

Incentives for process adaptations are many-fold. So-called ad-hoc processes, i.e. processes that are not fully designed, e.g. because there are a lot of different users with very specific requirements to the process require to enhance the process during execution. Another reason results from exceptions caused by violation of security rules, e.g. choose other data that are accessible to the actual user may be aggregated data or a certificate will be needed. Also other exceptions like “specific data are not available” could be the reason for changes of active processes.

Because processes often depend on comprehensive underlying data, we propose a method to use data dependencies on detecting if a process needs to be changed and adaptations are proposed (23). The enhanced system architecture is shown in Figure 9.

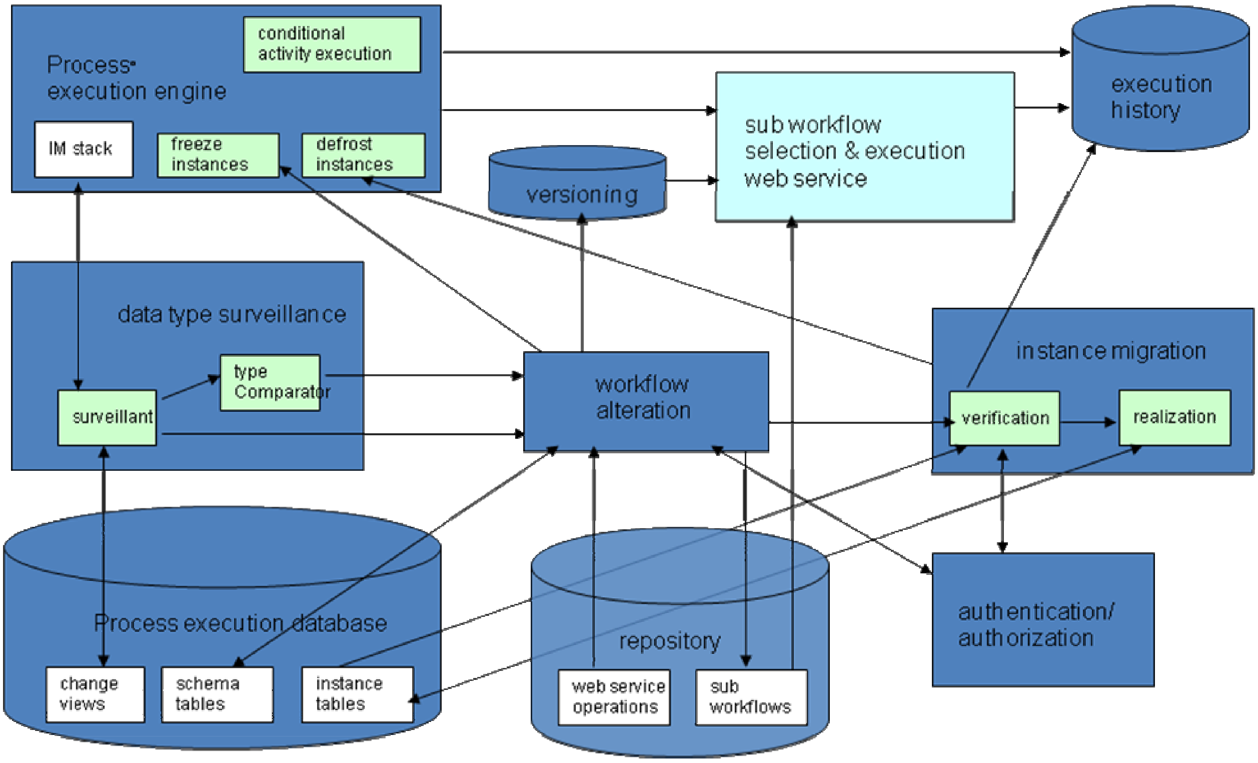


Figure 9: System architecture of a business process execution engine enabling data-induced adaptation of processes

Additional work sets up a formal model of adaptation operations at the BPMN level and realizes the transformation to the execution level of business processes with instance migration of running processes. The model is already established, the implementation and validation is on-going work (24). Adaptation functions are identified on the BPMN level and the BPEL level, in order to be able to transform adapted processes and to manage process versions. As basic model we support structural process changes with operators like AddTasks, DeleteTasks, AddSequenceFlows, SetSequenceFlowConditionExpressions. A crucial issue is the correct handling of data flow within the process instances.

Achieving an adaptable process management framework will need a lot of future work during the project.There should be support for fully automatic adaptation for special cases, e.g., adding/changing data or subprocesses to query data sources, or adaptation because of security or trust issues. Another category are semi-automatic adaptations guided by users, e.g. user interface of an assesor or candidate in the employability scenario. We want to provide a library of process patterns to support the automatic and semi-automatic adaptation. An important issue to come to a powerful adaptation support is to provide specification of process semantics in order to match adequate new tasks or subprocesses or to combine tasks to subprocesses. Also, we want to use credentials and security rules to guide adaptation.

5 Evaluation Scenario

The Kenteq APL has been modelled in BPMN. Figure 10 until Figure 16 show the current status. The diagrams are organized in a time line from left to right.

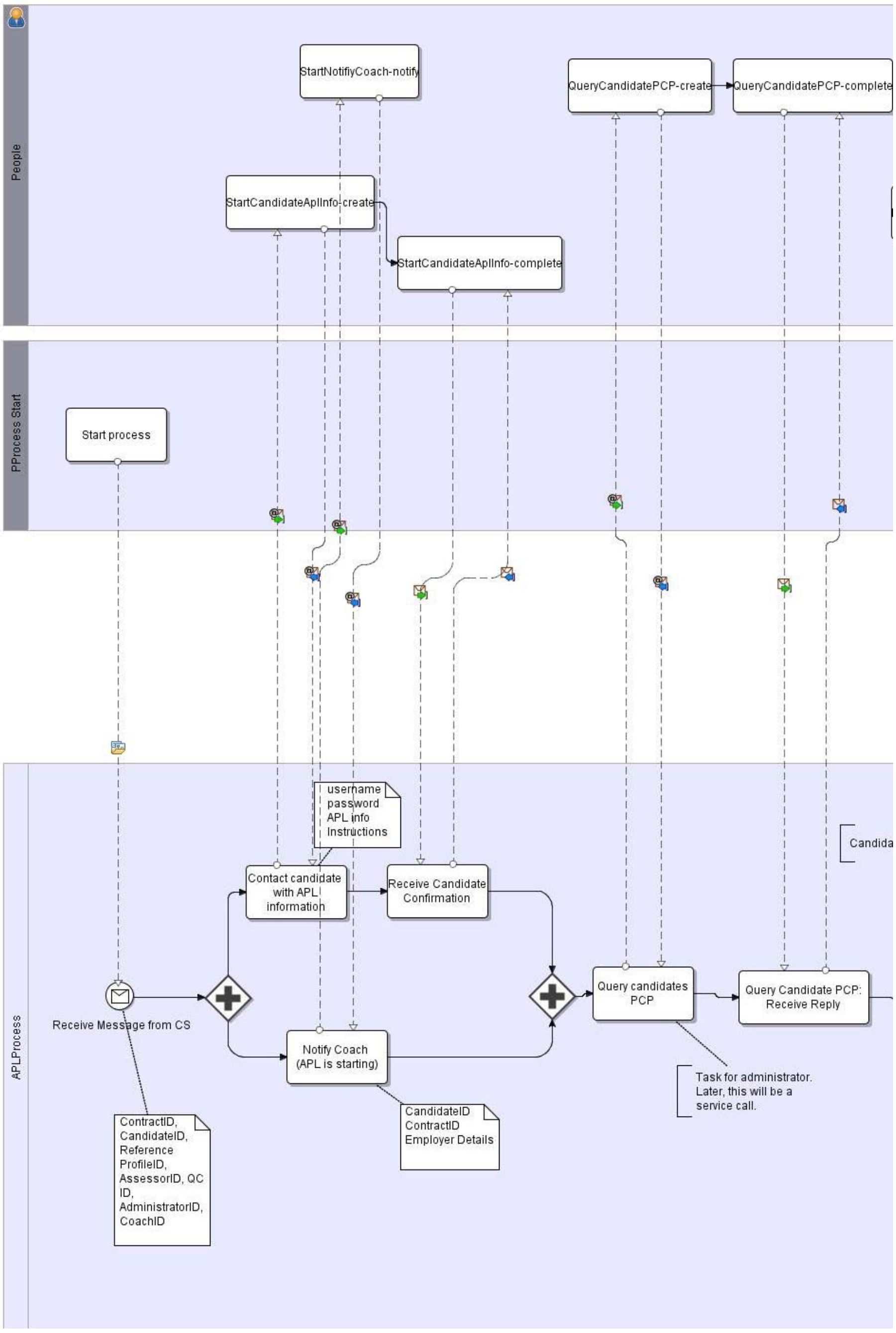

People
StartNotifiyCoach-notify
QueryCandidatePCP-create
QueryCandidatePCP-complete
StartCandidateAplInfo-create
StartCandidateAplInfo-complete
PProcess Start
Start process
APLProcess
username
password
APL info
Instructions
Candida
Contact candidate with APL information
Receive Candidate Confirmation
Query candidates PCP
Query Candidate PCP: Receive Reply
Receive Message from CS
Notify Coach (APL is starting)
Task for administrator. Later, this will be a service call.
CandidateID
ContractID
Employer Details
ContractID, CandidateID, Reference ProfileID, AssessorID, QC ID, AdministratorID, CoachID

Figure 10: Overview of the BPMN model of the process- start

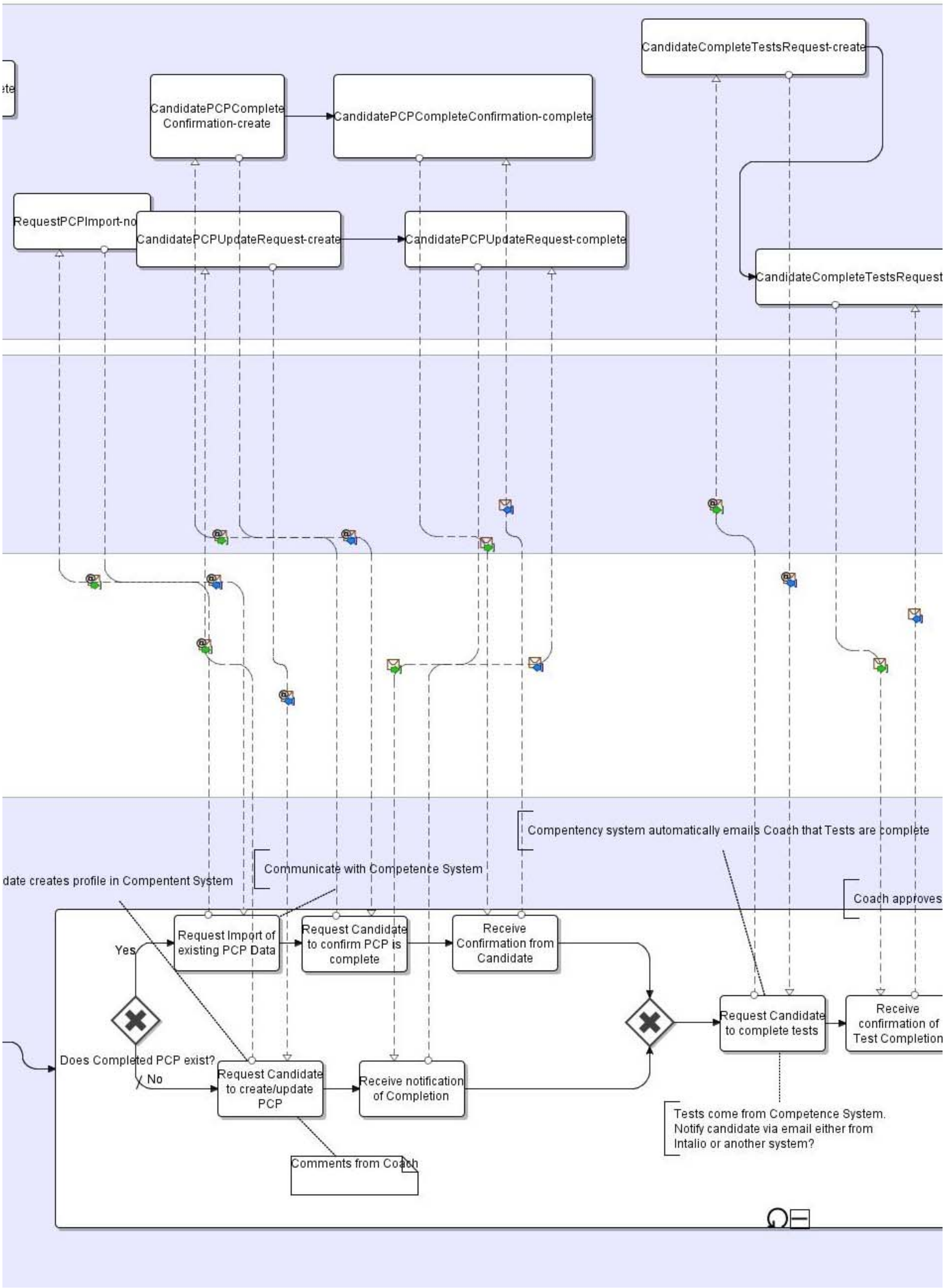

CandidateCompleteTestsRequest-create
CandidatePCPComplete Confirmation-create
CandidatePCPCompleteConfirmation-complete
RequestPCPImport-no
CandidatePCPUpdateRequest-create
CandidatePCPUpdateRequest-complete
CandidateCompleteTestsRequest
Compentency system automatically emails Coach that Tests are complete
Communicate with Competence System
date creates profile in Compentent System
Coach approves
Yes
Request Import of existing PCP Data
Request Candidate to confirm PCP is complete
Receive Confirmation from Candidate
Request Candidate to complete tests
Receive confirmation of Test Completion
Does Completed PCP exist?
No
Request Candidate to create/update PCP
Receive notification of Completion
Tests come from Competence System. Notify candidate via email either from Intalio or another system?
Comments from Coach

Figure 11: Overview of the BPMN model of the process - part 2

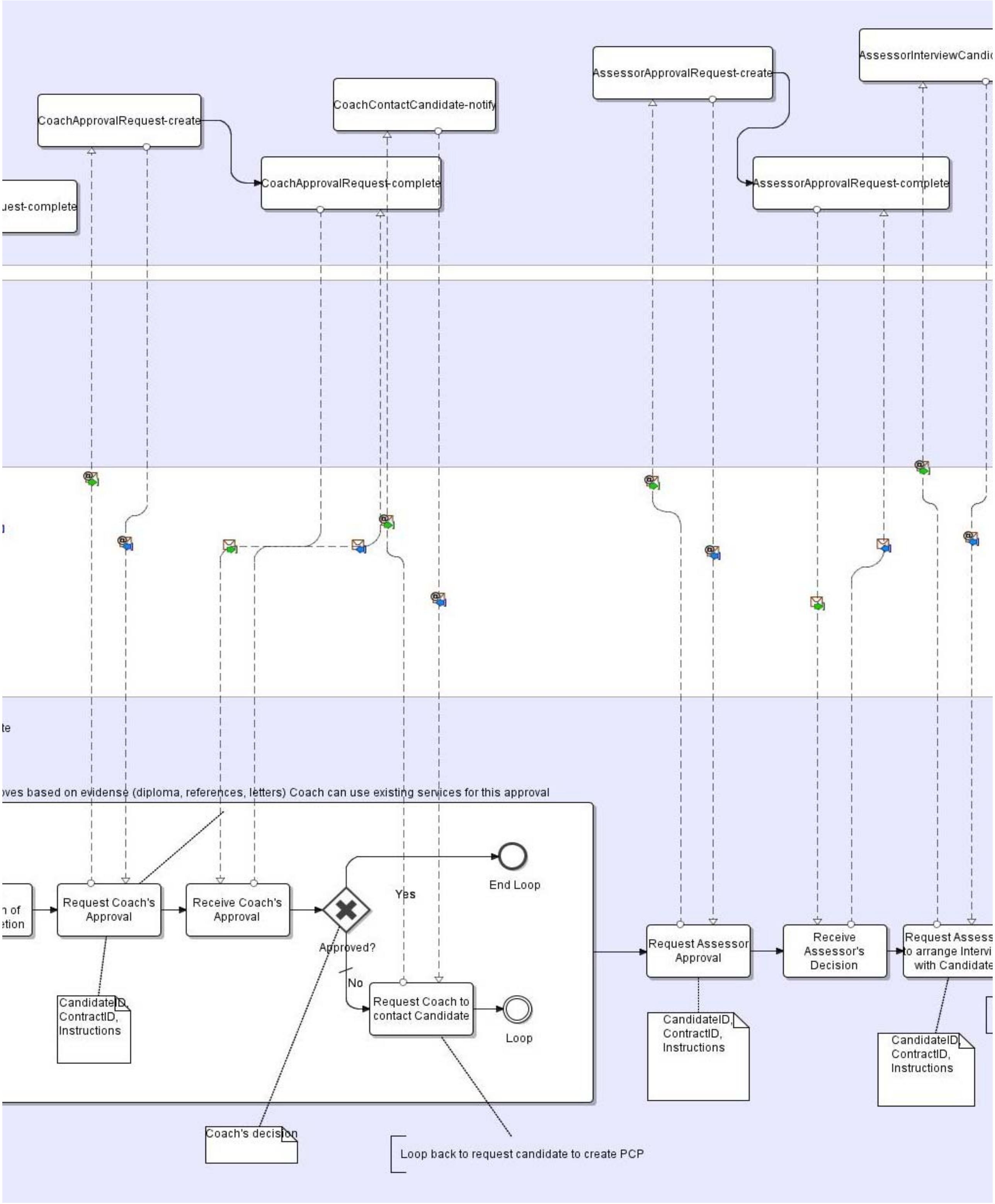

CoachApprovalRequest-create
CoachContactCandidate-notify
AssessorApprovalRequest-create
AssessorInterviewCandi
uest-complete
CoachApprovalRequest-complete
AssessorApprovalRequest-complete
te
oves based on evidense (diploma, references, letters) Coach can use existing services for this approval
n of
etion
Request Coach's Approval
Receive Coach's Approval
Yes
End Loop
Approved?
No
Request Coach to contact Candidate
Loop
Request Assessor Approval
Receive Assessor's Decision
Request Assess
to arrange Intervi
with Candidate
CandidateID, ContractID, Instructions
CandidateID, ContractID, Instructions
CandidateID, ContractID, Instructions
Coach's decision
Loop back to request candidate to create PCP

Figure 12: Overview of the BPMN model of the process - part 3

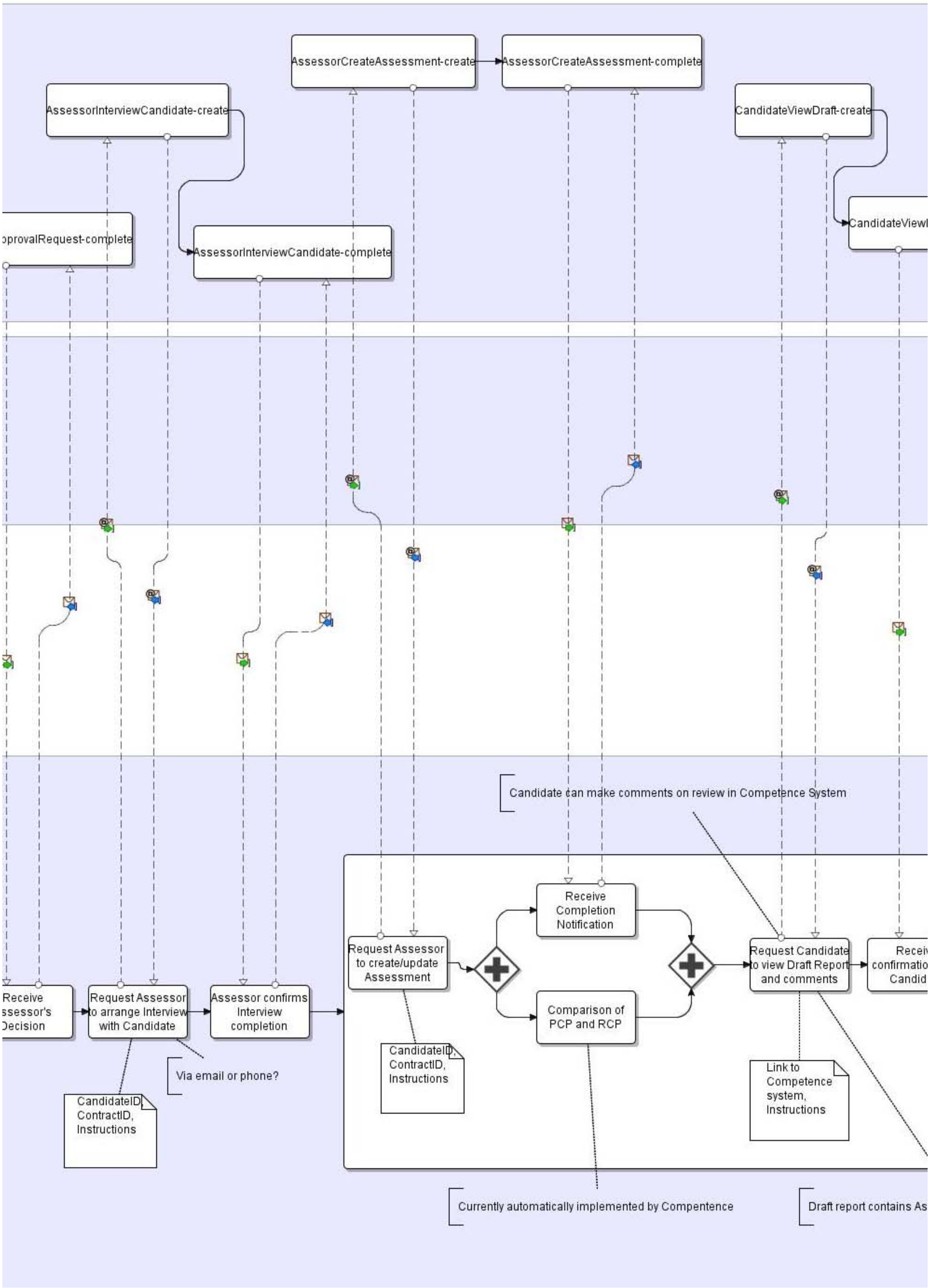

AssessorCreateAssessment-create
AssessorCreateAssessment-complete
AssessorInterviewCandidate-create
CandidateViewDraft-create
CandidateViewI
pprovalRequest-complete
AssessorInterviewCandidate-complete
Candidate can make comments on review in Competence System
Receive Completion Notification
Request Assessor to create/update Assessment
Request Candidate to view Draft Report and comments
Receiv confirmatio Candid
Receive ssessor's Decision
Request Assessor to arrange Interview with Candidate
Assessor confirms Interview completion
Comparison of PCP and RCP
CandidateID, ContractID, Instructions
Via email or phone?
CandidateID, ContractID, Instructions
Link to Competence system, Instructions
Currently automatically implemented by Compentence
Draft report contains As

Figure 13: Overview of the BPMN model of the process - part 4

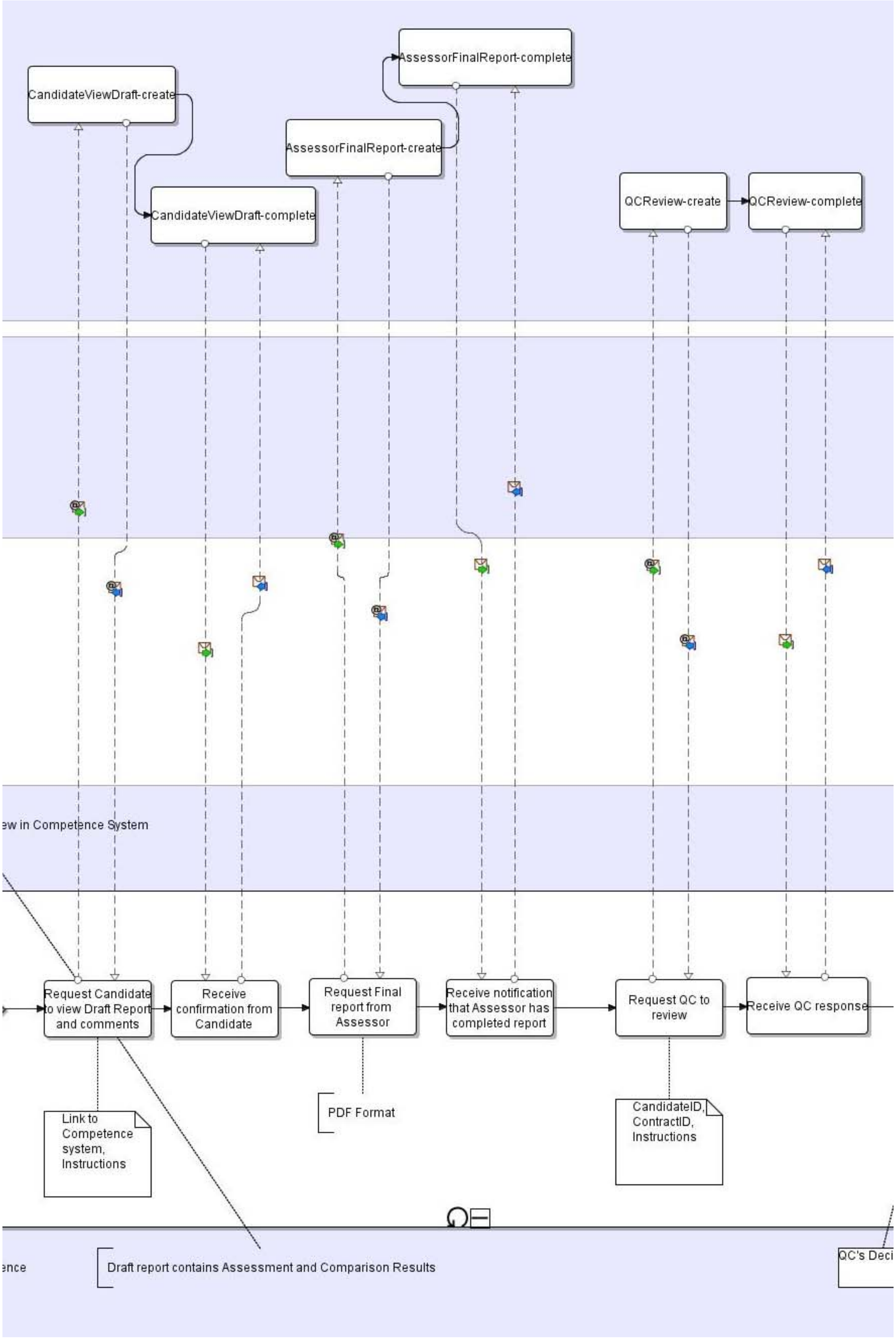

CandidateViewDraft-create
CandidateViewDraft-complete
AssessorFinalReport-create
AssessorFinalReport-complete
QCReview-create
QCReview-complete
ew in Competence System
Request Candidate to view Draft Report and comments
Receive confirmation from Candidate
Request Final report from Assessor
Receive notification that Assessor has completed report
Request QC to review
Receive QC response
Link to Competence system, Instructions
PDF Format
CandidateID, ContractID, Instructions
ence
Draft report contains Assessment and Comparison Results
QC's Deci

Figure 14: Overview of the BPMN model of the process - part 5

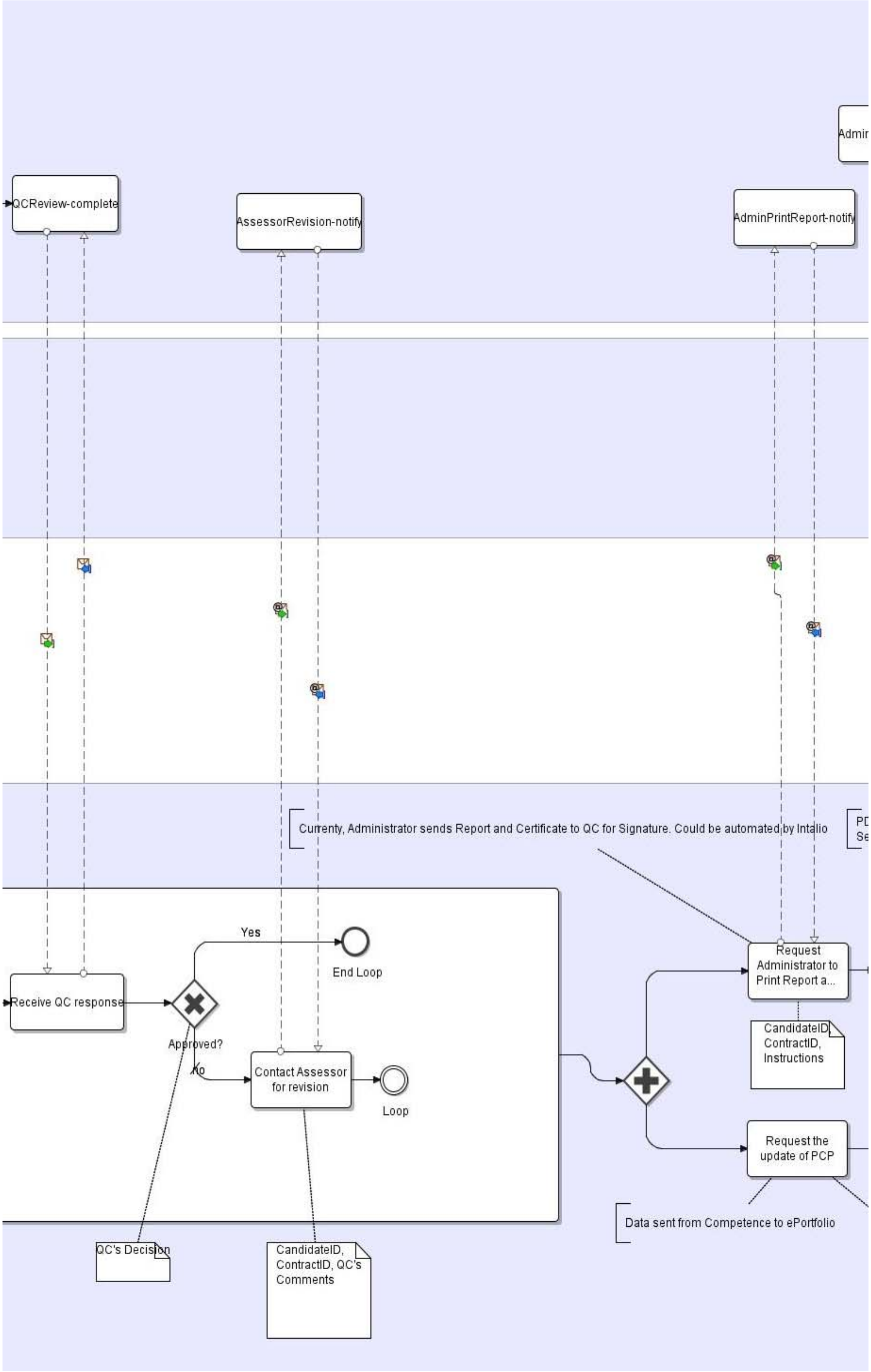

Admir
QCReview-complete
AssessorRevision-notify
AdminPrintReport-notify
Currenty, Administrator sends Report and Certificate to QC for Signature. Could be automated by Intalio
PD
Se
Yes
End Loop
Receive QC response
Approved?
No
Contact Assessor for revision
Loop
Request Administrator to Print Report a...
CandidateID, ContractID, Instructions
Request the update of PCP
Data sent from Competence to ePortfolio
QC's Decision
CandidateID, ContractID, QC's Comments

Figure 15: Overview of the BPMN model of the process - part 6

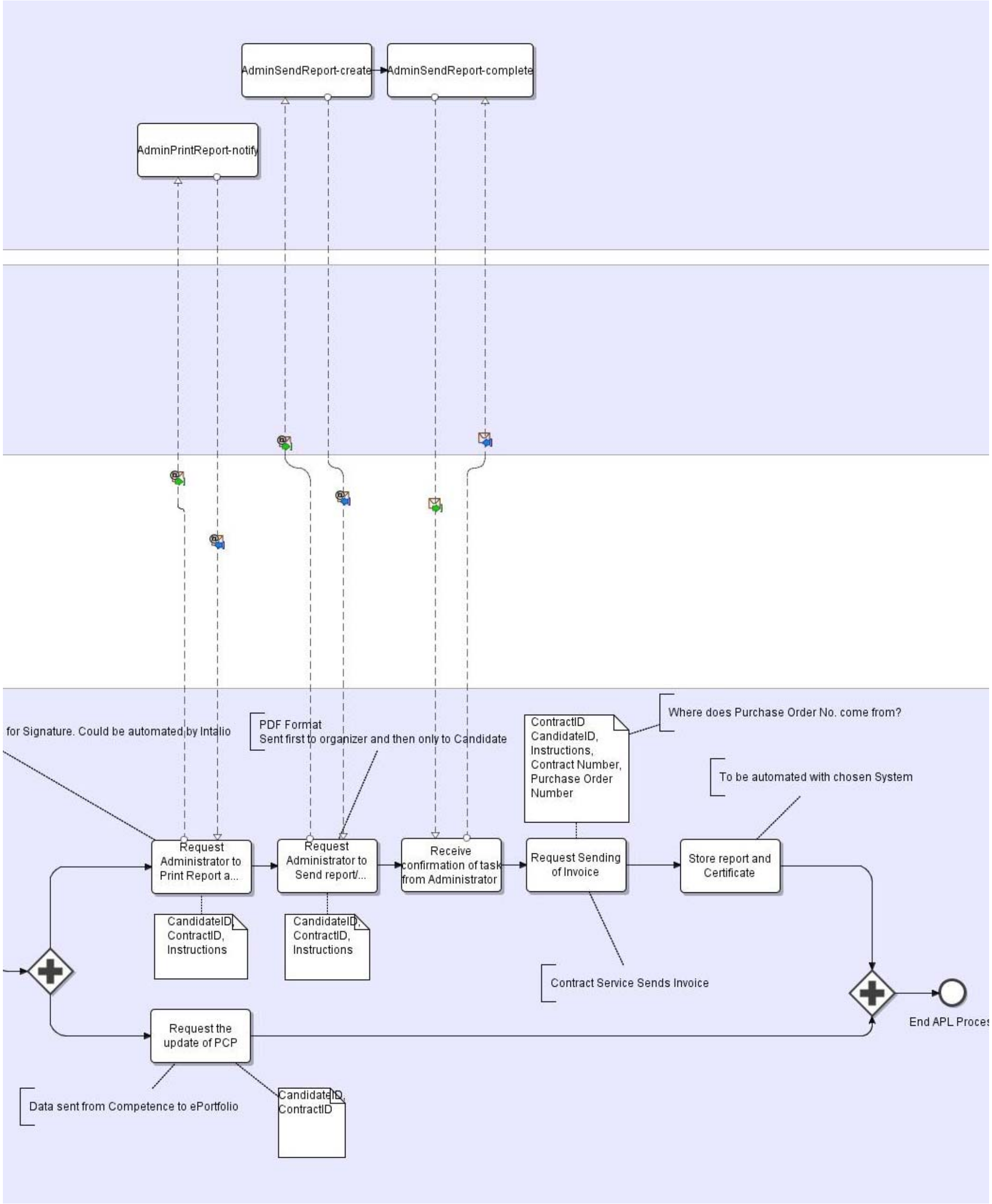

AdminSendReport-create
AdminSendReport-complete
AdminPrintReport-notify
for Signature. Could be automated by Intalio
PDF Format
Sent first to organizer and then only to Candidate
ContractID
CandidateID,
Instructions,
Contract Number,
Purchase Order
Number
Where does Purchase Order No. come from?
To be automated with chosen System
Request Administrator to Print Report a...
Request Administrator to Send report/...
Receive confirmation of task from Administrator
Request Sending of Invoice
Store report and Certificate
CandidateID,
ContractID,
Instructions
CandidateID,
ContractID,
Instructions
Contract Service Sends Invoice
End APL Proces
Request the update of PCP
Data sent from Competence to ePortfolio
CandidateID,
ContractID

Figure 16: Overview of the BPMN model of the process - part 7

The BPMN diagram is separated into three pools: The pool on the bottom represents the executable process with its control and data flow. The pool in the middle represents the externally visible web service interfaces of the process. Currently, this pool only contains an interface to start the process. Finally, the pool on the top represents the human notifications and tasks that are used in the process, i.e. the human workflow component of the process. The executable process communicates with the tasks and notifications using *Create* and *Complete* messages (the latter are used for tasks only). The *Create* and *Complete* activities represent interfaces of the Task Management Service (TMS).

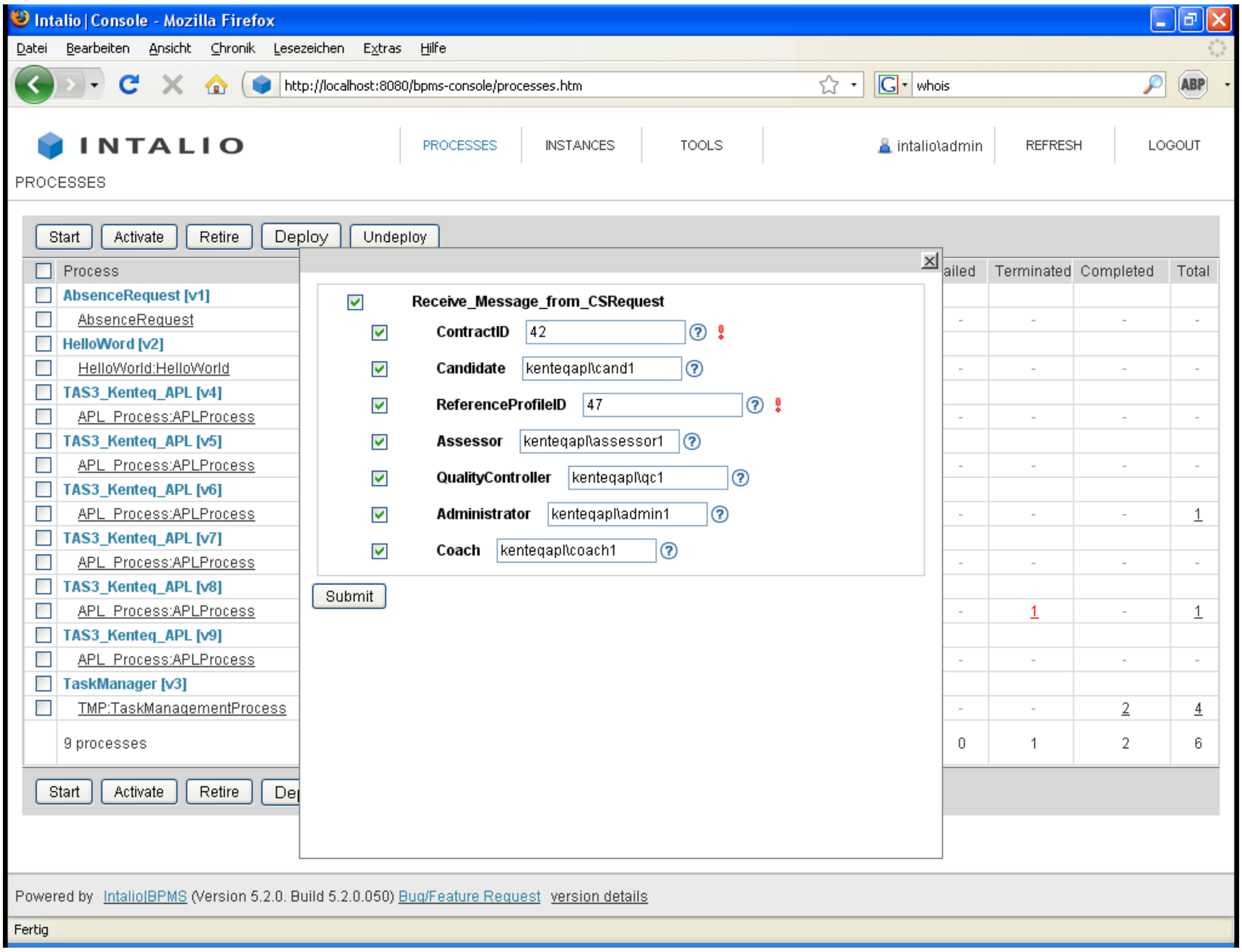
The process involves several human actors:

- The APL *candidate*, i.e. the employee who wants to achieve APL.

- The APL *coach*, who helps to fill out the portfolio.
- The *assessor*, who executes the actual APL process.
- The *quality controller* reviews the assessment for QA purposes.
- The *administrator* performs administrative tasks at the APL provider. Most of these tasks are mechanical in nature and are thus candidates for automation.

The current process model uses a single pool for all actors. The assignees of each task are set dynamically.

The BPMN model introduced here covers the current APL process at Kenteq, after all prerequisites (contracts, assignment of responsibilities) have been fulfilled. Thus, the actor list differs from the one given in section 2.1 of the “Design Requirements” (Deliverable D 1.4): On the one hand, actors from outside Kenteq are not included. On the other hand, it includes actors that only perform Kenteq-internal activities (administrator, quality controller) and were thus not included in the D 1.4 version.


Intalio | Console - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://localhost:8080/bpms-console/processes.htm
whois
INTALIO
PROCESSES
INSTANCES
TOOLS
intalio\admin
REFRESH
LOGOUT
PROCESSES
Start
Activate
Retire
Deploy
Undeploy
Submit
Start
Activate
Retire
Dep

☑ Receive_Message_from_CSRequest

☑ ContractID 42

☑ Candidate kenteqapl\cand1

☑ ReferenceProfileID 47

☑ Assessor kenteqapl\assessor1

☑ QualityController kenteqapl\qc1

☑ Administrator kenteqapl\admin1

☑ Coach kenteqapl\coach1

□ Process	ailed	Terminated	Completed	Total
□ AbsenceRequest [v1]				
□ AbsenceRequest	-	-	-	-
□ HelloWord [v2]				
□ HelloWorld:HelloWorld	-	-	-	-
□ TAS3_Kenteq_APL [v4]				
□ APL_Process:APLProcess	-	-	-	-
□ TAS3_Kenteq_APL [v5]				
□ APL_Process:APLProcess	-	-	-	-
□ TAS3_Kenteq_APL [v6]				
□ APL_Process:APLProcess	-	-	-	1
□ TAS3_Kenteq_APL [v7]				
□ APL_Process:APLProcess	-	-	-	-
□ TAS3_Kenteq_APL [v8]				
□ APL_Process:APLProcess	-	1	-	1
□ TAS3_Kenteq_APL [v9]				
□ APL_Process:APLProcess	-	-	-	-
□ TaskManager [v3]				
□ TMP:TaskManagementProcess	-	-	2	4
9 processes	0	1	2	6

Powered by Intalio|BPMS (Version 5.2.0. Build 5.2.0.050) Bug/Feature Request version details

Fertig

Figure 17: Web interface for starting a process

The process starts with a web service call from the contract service. The identity of all human actors in the process is determined by the contract service and passed into the process. This serves as a proof of concept for the ability to select actors dynamically at runtime. Intalio|BPMS also provides a web interface that allows to send the start message to the service. As shown in Figure 17, all arguments of the process start message can be entered there, including the user names of the actors.

Detailing the APL process

In the following, parts of the whole BPMN diagram of Figure 10 until Figure 16 will be shown in an enlarged view in order to illustrate specific features of Intalio|BPMS and interesting aspects of the Kenteq APL process.

Flow control

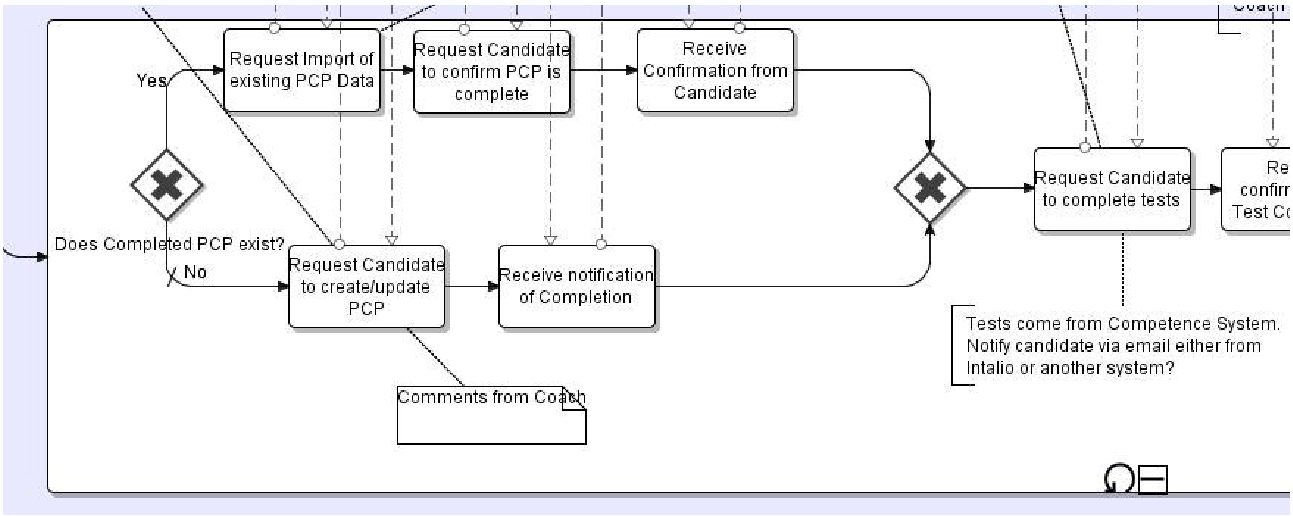

Yes
Request Import of existing PCP Data
Request Candidate to confirm PCP is complete
Receive Confirmation from Candidate
Does Completed PCP exist?
No
Request Candidate to create/update PCP
Receive notification of Completion
Comments from Coach
Request Candidate to complete tests
Re confirr Test Co
Tests come from Competence System. Notify candidate via email either from Intalio or another system?

Figure 18: Excerpt from the process model demonstrating flow control – part 1

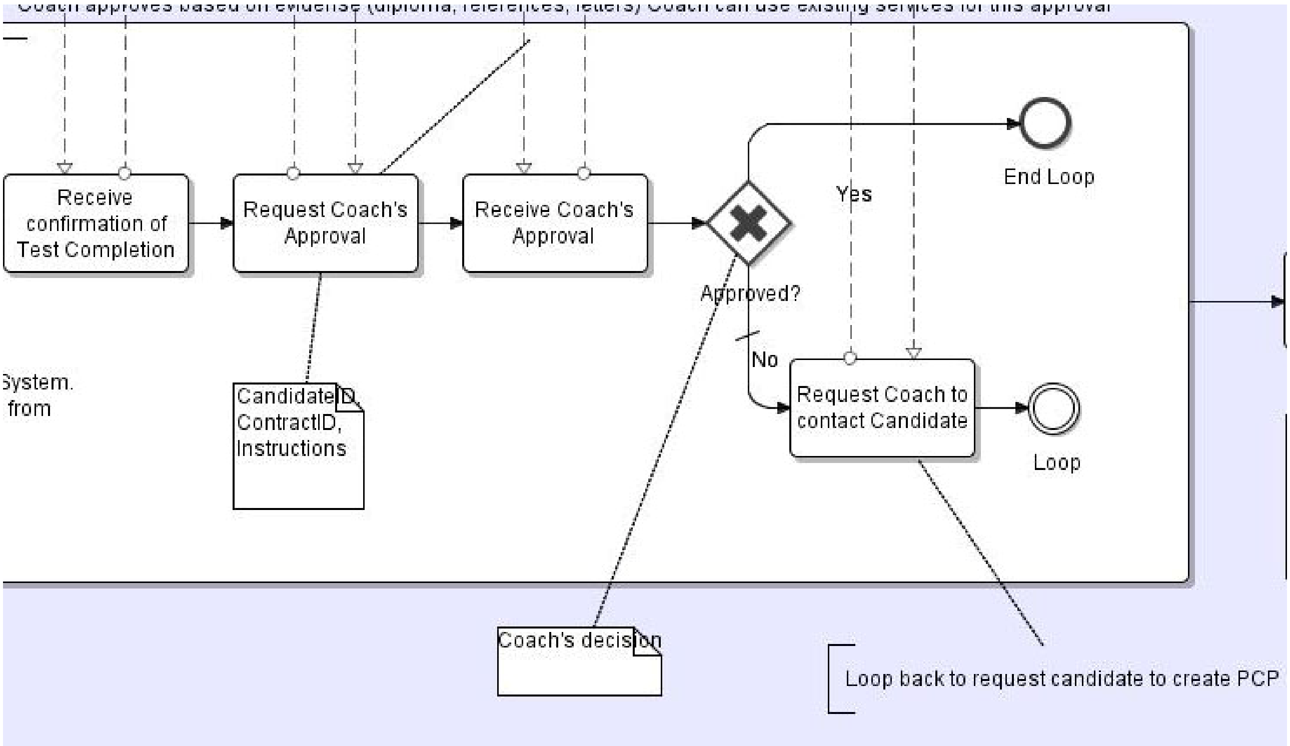

Coach approves based on evidense (diploma, references, letters) Coach can use existing services for this approval
Receive confirmation of Test Completion
Request Coach's Approval
Receive Coach's Approval
Approved?
Yes
End Loop
No
Request Coach to contact Candidate
Loop
System.
from
CandidateID, ContractID, Instructions
Coach's decision
Loop back to request candidate to create PCP

Figure 19: Excerpt from the process model demonstrating flow control – part 2

The APL process modelled so far in Figure 18 and Figure 19 make use of several BPMN sequence flow constructs:

- Exclusive (XOR) data-based sequence flow forking and joining: Based on whether a completed Personal Competency Profile (PCP) for the candidate exists or not, the existing PCP is imported and confirmed by the candidate, or the candidate has to create or update a PCP.
- Looping: The PCP has to be updated by the candidate until it is approved by the coach. Based on the coach's decision, control flow either goes through the loop body again or leaves the loop.

Security issues

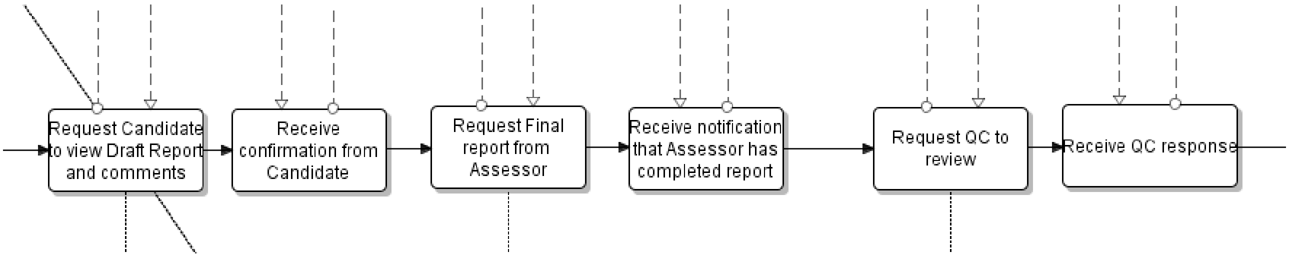
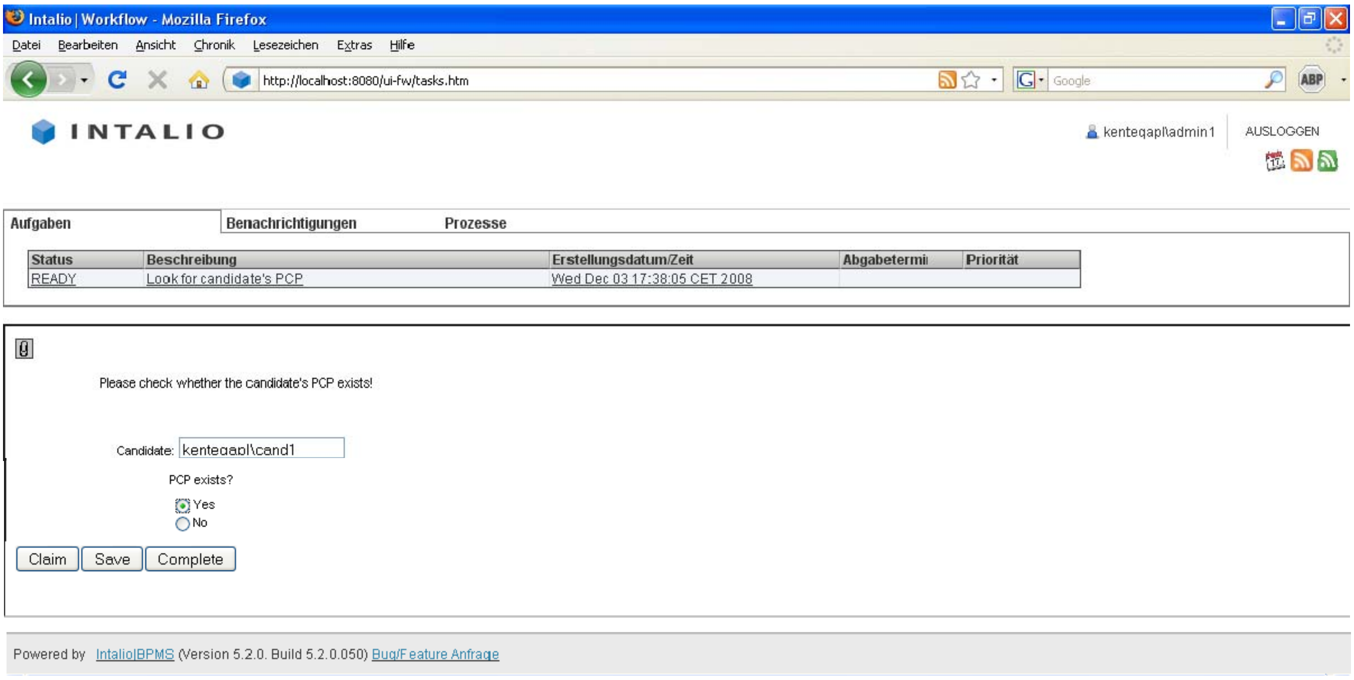

Request Candidate to view Draft Report and comments
Receive confirmation from Candidate
Request Final report from Assessor
Receive notification that Assessor has completed report
Request QC to review
Receive QC response

Figure 20 Excerpt from the process model demonstrating a SoD use case

At several points in the APL process, activities of one actor need to be confirmed by another one in order to be effective. For example, the final report written by the assessor needs to be reviewed by the quality controller (cf. Figure 20). This is a classic example for a separation of duty

Interaction with human process participants

The most basic method for people to interact with business process is the *workflow console* that is part of Intalio|BPMS. Figure 21 shows an example of a task presented in the workflow console. The displayed form provides the user with the information he needs to perform the task. The data entered in the form (in this case, a binary “Yes”/”No” decision) is sent back to the process and can be used, e.g. for flow control.


Intalio | Workflow - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://localhost:8080/ui-fw/tasks.htm
Google
INTALIO
kenteqapl\admin1
AUSLOGGEN
Aufgaben
Benachrichtigungen
Prozesse

Status	Beschreibung	Erstellungsdatum/Zeit	Abgabetermi	Priorität
READY	Look for candidate's PCP	Wed Dec 03 17:38:05 CET 2008		

Please check whether the candidate's PCP exists!

Candidate: kenteqapl\cand1

PCP exists?

● Yes
○ No

Powered by Intalio|BPMS (Version 5.2.0. Build 5.2.0.050) Bug/Feature Anfrage

Figure 21: Task in the workflow console

6 Conclusions

This document represents the first phase of the design of a business process management platform for the TAS^3 project. We will enhance this report according the project plan in the next project phases with more detailed and refined results.

Regarding the requirements described in deliverable D1.2 and D1.4 we designed the diverse components of a basic platform for secure process management, i.e. a business process modelling platform and execution engine for processes in service-oriented applications, security issues of business processes, and adaptability of business processes. In order to enhance the functionality and to achieve an adequate required functionality and quality of these components we will need to underpin business processes with ontologies. These components have to interoperate with each other and have to be integrated in the TAS^3 security architecture. For that we have worked on the system architecture.

The focus lies on providing a business process modelling platform to enable the partners to model business processes in the application scenarios, and to provide the possibility to run the modelled business processes. Therefore there are organized training sessions for business process modelling with the Intalio tools and we modelled a real-world business process of the employability scenario, the Kenteq APL process, in cooperation of modelling experts and domain experts. On this basis we will be able to model more complex and demanding new flexible processes in the employability area and also for e-health applications. Furthermore, modelling the TAS^3 security process is planned in 2009, so that we can use the resulting process to check diverse process alternatives and can investigate the integration of business processes with the TAS^3 security levels.

A further focus handled with security of processes, which concepts will be required, what is provided by standards, which enhancements we need. The result will be used in future work for enhancing the graphical process modelling tool with features to specify process security in a more comprehensive way.

In respect to adaptability we achieved concepts for structural adapting processes, which serves as fundamental platform for future more challenging adaptation concepts. Further on there are results on handling adaptation of processes which are induced by changes of data used by the process. We already started with implementation of the concepts.

The ontology component started with setting up the ontologies starting from the upper level. Next tasks will be to enhance process modelling with semantics of the application and of security aspects by annotation, as well as to work on using ontology specifications during process adaptation and execution.

A main issue of WP3 in 2009 is to implement a first software version which will help to validate the planned concepts against the requirements of the TAS^3 security system and also against the applications. Then we have to go on integrating the processes platform in the TAS³ architecture in close cooperation with WP8. Adaptability and process security will need further refinement in order to yield a first prototypical more powerful business process modelling and handling platform.

7 References

1. Web Services Business Process Execution Language v2.0. *OASIS Standard.* [Online] http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

2. **OMG.** Business Process Modelling Notation (BPMN) Information. [Online] October 2008. http://www.bpmn.org/.

3. **Intalio.** Intalio|BPMS - Business Process Management System. [Online] September 2008. http://www.intalio.com.

4. **OASIS.** BPEL4People v1 Contribution (Draft) - OASIS WS-BPEL Extension for People (BPEL4People) TC. [Online] http://www.oasis-open.org/committees/document.php?document id=27505&wg_ abbrev=bpel4people.

5. **OASIS-Standard.** Web Services Business Process Execution Language v2.0. [Online] http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

6. **Intalio.** Intalio / Tempo: The Open Source Workflow Framework . [Online] September 2008. http://tempo.intalio.org.

7. WS-HumanTask v1.0 Contribution. [Online] http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf.

8. *The Consistency of Task-Based Authorization Constraints in Workflow Systems.* **Tan, Crompton, Gunter.** 2004. 17th IEEE Computer Security Foundations Workshop.

9. *Bridging the gap between business models and workflow specifications.* **J. Dehnert, W. van der Aalst.** 2004, Int. J. Coop. Inf. Syst., Bd. 13, S. 289-332.

10. *Automated derivation of executable business processes from choreographies in virtual organisations.* **Ingo Weber, Jochen Haller, Jutta Mülle.** Inderscience, 2008, Int. Journal of Business Process Integration and Management, Bd. 3, S. 85-95.

11. *Building Conference Proceedings Requires Adaptive Workflow and Content Management.* **J. Mülle, K. Böhm, N. Röper, T. Sünder.** Seoul, Korea, 2006. Proc. 32nd Intl. Conf. on Very Large Data Bases.

12. **E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana.** Web Services Description Language (WSDL) 1.1. *W3C Note, World Wide Web Consortium.* [Online] 15 March 2001. http://www.w3.org/TR/wsdl.

13. *The Semantic Web.* **T. Berners-Lee, J. Hendler, and O. Lasilla.** May 2001, Scientific American, pp. 34-43.

14. *Towards principles for the design of ontologies used for knowledge sharing.* **Gruber, T.R.** Deventer, The Netherlands, 1993. Formal Ontology in Conceptual Analysis and Knowledge Representation. pp. 907-928.

15. *Semantic EPC: Enhancing Process Modeling Using Ontology Languages.* **O. Thomas and M. Fellmann.** 2007. Workshop on Semantic Business Process and Product Lifecycle Management.

16. *An Ontology Engineering Methodology for DOGMA.* **P. Spyns, Y. Tang, and R. Meersman.** 3 2008, Journal of Applied Ontology, S. 13-39.

17. *DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering.* **A. de Moor, P. De Leenheer, and R. Meersman.** 2006. 14th International Conference on Conceptual Structures. S. 189-203.

18. **M. Hepp, P. De Leenheer, T. Mens, and Y. Sure (eds).** *Ontology Management for the Semantic Web, Semantic Web Services, and Business Applications.* Berlin : Springer, 2008.

19. **Consortium, ATHENA.** Advanced technologies for interoperability of heterogeneous enterprises networks and their applications. 2006.

20. **Consortium, INTEROP.** Interoperability research for networked enterprises applications and software. *Network of excellence, annex 1-description of work.* 2003.

21. **Leenheer, P. De.** Ontological Foundations for Community Evolution. *Ph.D. thesis, Vrije Universiteit Brussel, manuscript.* 2009.

22. *Keynote: Towards Ontological Foundations of Agent Community Evolution.* **Leenheer, P. De.** 2008. Proceedings of 32nd Annual IEEE International Computer Software and Applications Conference. S. 523-528.

23. *Context dependency management in ontology engineering: a formal approach.* **P. De Leenheer and A. de Moor, and R. Meersman.** Journal on Data Semantics, S. 26-56.

24. *T-Lex: a Role-based Ontology Engineering Tool.* **D. Trog, J. Vereecken, S. Christiaens, P. De Leenheer, and R. Meersman.** Montpellier, France, 2006. Proceedings of the On The Move to Meaningful Internet Systems Workshops. S. 1191-1200.

25. *DOGMA-MESS: A Tool for Fact-Oriented Collaborative Ontology Evolution.* **P. De Leenheer and C. Debruyne.** Monterrey, Mexico, 2008. Proceedings of On the Move to Meaningful Internet Systems 2008: ORM.

26. **Weingardt, Dominik.** Extending a WFMS with Data-Induced Adaptability (in german). *diploma thesis.* University of Karlsruhe, Germany, March 2008.

27. **Haberecht, Thorsten.** Structural adaptation of BPMN/BPEL-Workflows and integration with the Intalio BPMS (in german). *draft of a diploma thesis.* University of Karlsruhe, 2008.

List of Figures