A HUB SYSTEM FOR CLOUD-COMPUTING BASED BUSINESS-COLLABORATION

Automating Ontology-Enabled Electronic Business-Service Discovery TECHNICAL REPORT: C-2011-39

Alex Norta
Department of Computer Science, University of Helsinki,
P.O. Box 68 (Gustaf Hällströmin katu 2b), FI-00014 University of Helsinki,
Finland
firstname.lastname@cs.helsinki.fi

Keywords: SOC, BPM, B2B, broker, hub, collaboration, trust, reputation, ontology

Abstract: The management and coordination of business-process collaboration experiences changes because of globalization, specialization, and innovation. Service-oriented computing (SOC) is a means towards business-process automation and recently, many industry standards emerged to become part of the service-oriented architecture (SOA) stack. In a globalized world, organizations face new challenges for setting up and carrying out collaborations in semi-automating ecosystems for business services. For being efficient and effective, many companies express their services electronically in what we term business-process as a service (BPaaS). Companies then source BPaaS on the fly from third parties if they are not able to create all service-value in-house because of reasons such as lack of reasoures, lack of know-how, cost- and time-reduction needs. Thus, a need emerges for BPaaS-HUBs that not only store service offers and requests together with information about their issuing organizations and assigned owners, but that also allow an evaluation of trust and reputation in an anonymized electronic service marketplace. In this paper, we analyze the requirements, design architecture and system behavior of such a BPaaS-HUB to enable a fast setup and enactment of business-process collaboration. Moving into a cloud-computing setting, the results of this paper allow system designers to quickly evaluate which services they need for instantiationg the BPaaS-HUB architecture. Furthermore, the results also show what the protocol of a backbone service bus is that allows a communication between services that implement the BPaaS-HUB. Finally, the paper analyzes where an instantiation must assign additional computing resources vor the avoidance of performance bottlenecks.

1 INTRODUCTION

In a globalized business setting, enterprises may run complex supply chains across several tiers that comprise many geographical regions. Original equipment manufacturers (OEM) maintain intricate in-house processes of which parts are outsourced to suppliers. With the emergence of SOC (Allen et al., 2006; E.A.Marks and Bell, 2006), such business-to-business (B2B) collaboration with business processes as a service (BPaaS) may be semi-, or fully automated and run on platforms and infrastructure from computing clouds (N.Antonopoulos and Gillam, 2010).

The predicted increase in BPaaS leads to a considerable communication overhead for enterprises if they intend to match service requests with service provisions. To manage the increasing communication overhead due to many BPaaS, intelligent HUBs, e.g., broker systems, should exist in between as middleware. To mention several examples, in (Muñoz Frutos, 2009), a *BPaaS-HUB* for business grids presents a backward compatible and lightweight approach that uses semantic annotations in service descriptions. A quality of service (QoS) ontology in (Tran et al., 2009) in combination with a ranking algorithm is used in a HUB to facilitate automatic and dynamic service discovery and selection. A *BPaaS-HUB* concept in (Loreto et al., 2009), bridges the integration gap between telephone companies and the IT world. Depending on the location of a mobile device, an automatic service assignment occurs for mashup creations. We define a mashup as a Web page or application that combines data or functionality from two or more external sources to create a new service. Service HUBs function in (Klien et al.,) as an open and distributed environment of geographic information Web services that are searchable with the help of ontology-based metadata.

The references above show that *BPaaS-HUBs* appear for various application domains. However, a solid understanding and system model is missing for a *BPaaS-HUB* so that business- and logistics managers, industrial marketers and so on, may engage in setting up cross-organizational B2B collaboration in a semi-automated way. Such a *BPaaS-HUB* should permit a layman without SOC knowledge to engage in service matching while the required computing complexity remains hidden from a user by the HUB. This paper fills the gap and presents a *BPaaS-HUB* architecture that we present conceptually and also as a model that we verify with formal methods. That way we answer the research question for the development of BPaaS-HUBs, namely, how must a *BPaaS-HUB* be designed for facilitating the speedy discovery of trustworthy and reputable service offers that are matchable with service requests? We deduce several sub-questions from the main research questions:

1. What are the requirements for a *BPaaS-HUB* that take into account human users being laymen?
2. With what architectural styles and patterns are the requirements translated into a conceptual *BPaaS-HUB* architecture?
3. How does a *BPaaS-HUB* guide service discovery so that a successful matching with a service request is achieved and enactment commences while ensuring there are no performance bottlenecks?
4. What is the protocol of a service bus for allowing the interaction between services that instantiate a *BPaaS-HUB* of this paper?

Consequently, the structure of the paper is as follows. Section 2 presents high-level characteristics of B2B collaboration from which we deduce a set of requirements that take into account the user being a layman with respect to service-oriented and cloud computing. In Section 3, a *BPaaS-HUB* architecture adheres to the set of requirements by incorporating corresponding architectural styles and patterns. Section 4 presents a *BPaaS-HUB* architecture in a formalized model that permits an analysis with formal methods. Section 5 discusses the results of applying verification methods on the formal *BPaaS-HUB* model that influence how computing capacity is allocated for respective functionalities of a *BPaaS-HUB* instantiation. Section 6 demonstrates how a *BPaaS-HUB* architecture is instrumental for evaluating its instantiation with existing system applications and to which extent they need to be modified and extended to become suitable realization-services. Section 7 concludes the paper and presents future work.

2 B2B CHARACTERISTICS

To explain the characteristics of B2B collaboration, first Section 2.1 shows conceptually the vertical and hierarchical nature of service composition. Section 2.2 presents a framework for feasibly matching services. Finally, Section 2.3 deducts from the previous sections requirements for a *BPaaS-HUB* architecture in the sequel.

2.1 THE COLLABORATION PYRAMID

Observing B2B collaborations in the EU research project CrossWork (Grefen et al., 2009), particular features are characteristic. An OEM organizes the creation of value in an in-house process that is decomposable into different perspectives, e.g., control flow of tasks, information flow, personnel management, allocation of production resources, and so on. A complex product of an OEM typically comprises many components of which several need to be acquired from suppliers. The reasons for acquiring parts externally are manifold, e.g., the OEM cannot produce with the same quality, or an equally low price per piece, the production capacity is not available, required special know-how is lacking, and so on.

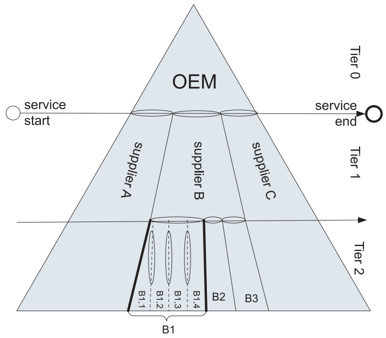

Figure 1: A collaboration pyramid in B2B.

In the scenario of Figure 1, the horizontal ellipses denote the client/server integration of outsourced parts of the in-house process to lower-level clients across several tiers of a supply chain (Norta and Grefen, 2007). The outsourced business process receives refinements by the supplier that remain opaque to the service consumer. Also the supplier only has awareness of the OEM's outsourced process part but the remaining in-house process equally remains opaque. For client/server integration several projects investigated enterprise interoperability (Alonso et al., 1999; Lazcano et al., 2001; Hoffner et al., 2005; Mehandjiev and Grefen, 2010).

Vertical ellipses in Figure 1, depict a peer-to-peer (P2P) collaboration within a cluster of small and medium sized enterprises (SME). If several SMEs form a composed service in a P2P way, they become a supplier for a higher-level service consumer. For managing such P2P service collaboration (Kutvonen et al., 2007), their lifecycle needs to be managed (a) for business-community formation to compose services, (b) for the evolution of such business communities through epochs with changing members and modified services, and eventually (c) for the dissolution of P2P business communities.

2.2 SERVICE MATCHING

Assuming a SOC-automation of the collaboration pyramid in Figure 1 with BPaaS, matching service requests and service offers becomes a challenge when we only employ computationally expensive, high-quality formal methods. For example, computationally expensive but of high quality are Petri-net-based approaches (Reisig and Rozenberg, 1998) that support service-based business process collaborations (Aalst, 2002; Bonchi et al., 2007; Martens, 2003a; Martens, 2003b; Reisig et al., 2005; van der Aalst et al., 2008) because of the state-space-explosion problem.

For a *BPaaS-HUB*, we consider a stepwise matching approach as depicted in Figure 2. On the one hand, the amount of services decreases with every lower matching step while, on the other hand, the matching methods are increasingly computationally expensive but of higher quality towards the lower matching levels. The top level is a matching of service offers and requests based on extracted and ontologically clarified keywords contained in service descriptions. A matching of left over services subset requires on the next level machine-readable service-level agreements (SLA) with, e.g., WS-Agreement (Andrieux et al., 2007) or WSLA[1]. As an example for this matching type, in (Müller et al., 2009), matching templates and instantiations involve computing the adherence of the latter to templates. The next service-matching level employs BPEL specifications and uses heuristics. For example in (Eshuis and Norta, 2009), tree representations of BPEL processes are the basis for applying matching heuristics. Finally, the left over subset of services is small enough to use high-quality methods that are

[1] http://www.research.ibm.com/wsla/

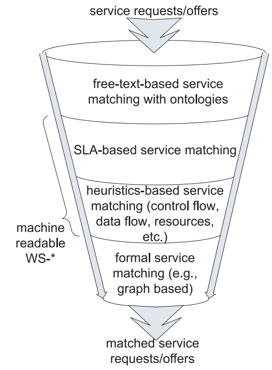

Figure 2: Increased complexity layers for service matching.

computationally expensive. For example in (Norta and Eshuis, 2009), a Petri-net based matching of processes also verifies the soundness of a service composition.

2.3 *BPAAS-HUB* REQUIREMENTS

For a *BPaaS-HUB* architecture in the sequel, we deduct requirements from the discussed B2B characteristics.

1. A *BPaaS-HUB* must allow laymen who have no or little SOC knowledge to engage in service discovery and matching.
2. Since the HUB is part of an anonymized service ecosystem, users must be able to check the trustworthiness and reputation of service offers and requests.
3. The HUB must support resolving ambiguities in the human-and machine readable service representations.
4. The HUB must support feasible service matching as, e.g., described in Section 2.2.
5. The user interaction with the HUB must be logged for extracting business intelligence.

3 A SERVICE-HUB ARCHITECTURE

We specify a conceptual system architecture for a *BPaaS-HUB*. Conceptual architectures (also known as logical architectures) facilitate the understanding of the interactions between components and the functionalities provided by the system. For a *BPaaS-HUB* architecture, we follow design principles, styles and patterns (Bengtsson, 2002; Gamma et al., 1995). Architectural styles comprise a description of component types and their topology, a description of the pattern of data and control interaction among the components, and an informal description of the benefits and drawbacks of using a particular style.

The conceptual architecture depicted in Figure 3 utilizes the principles of separation of concern, it follows a layer style, employs a pipes-and-filters pattern and pattern-based components for abstracting data repositories.

Separation of Concerns: For breaking the system complexity down to manageable parts, we introduce separations of concerns with the characterizing questions who, with, what and how. In Figure 3, columns show these separations:

WHO: refers to the business entities a user searches for. They may be services in specific domains, organizations, or persons related to service categories. *WITH*: refers to establishing on the fly the ontological infrastructure needed to resolve ambiguity issues in service descriptions. *WHAT*: refers to the need for pulling in additional service-related information from the Web cloud for a trust-enhancing mashups. *HOW*: refers to the application infrastructure necessary for the services to be matched and enacted. Additionally we propose social mining techniques for analyzing the logged user interaction with the HUB and extracting business intelligence that way.

Layer Style: A layer style separates vertically a *BPaaS-HUB* architecture, characterized by communication exchanges only permitted to the adjacent higher or lower layer. The advantage of this architecture is a limitation of communication exchanges between layers that facilitate a decoupling and replacement with alternative components.

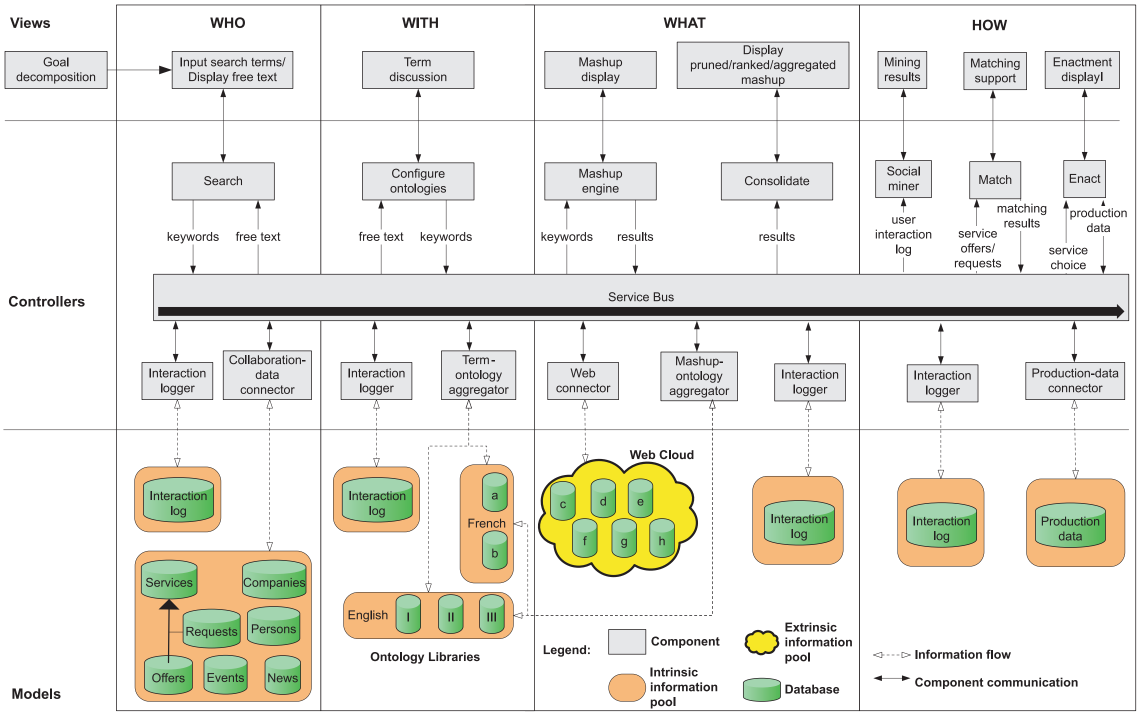

Figure 3: Architecture of a BPaaS-HUB.

In Figure 3, the top layer called *Views*, depicts all user-interface components. The middle layer termed *Controllers*, shows components with application logics while the lowest layer termed *Models*, contains all system intrinsic or third-party extrinsic information sources from the Web cloud for trust-building mashups.

The ontology libraries in Figure 3 group members of language categories. Other categorization options may delimit according to geographic regions, industrial domains, product families, market segments, and so on. Note that individual ontology libraries can be members of several category sets. For every concern-separating column, a dedicated database logs the user interaction with a *BPaaS-HUB*.

Pipes and Filters Pattern: The components of the controller layer instantiate a pipes-and-filters pattern enforced by a service bus. In a fully automated scenario, an ontology-supported *Goal decomposition* delivers input for what business entities are sought after. The automated goal decomposition may support a human user of a *BPaaS-HUB* in a semi-automated scenario or may be entirely circumvented by a user. A service search results both in human-readable text and optional machine-readable WS-* specifications that belong to the SOA stack. All types of service representations potentially contain ambiguities. Hence, an analysis of search results may take place that culminates in a dynamically linked library of ontology libraries for resolving ambiguities in the service representations.

In Figure 3, following a pipes-and-filters pattern, a mashup engine performs automated searches for trust and reputation establishment in user-selected information pools of the Web cloud. The results of that search may be numerous, erroneous and processing them as a user is cognitively stressful. Hence, a consolidation must take place in which result classification takes place into refuse versus the remainder that is ranked according to ontological relevance and/or aggregated where possible. The logged user interaction with a *BPaaS-HUB* may be mined for generating business intelligence. Additionally, a component in a *BPaaS-HUB* stands for matching of services in stages as described in Section 2.2. Finally, the enactment of machine-readable WS-* service representations commences.

Abstract Data Repository: On the *controller* layer of Figure 3, the *collaboration data connector*, *term ontology aggregator*, *interaction logger* and *mashup ontology aggregator* are components of the architectural style *abstract data repository* (Klein and Kazman, 1999). This architectural style, on the one hand, keeps the producers and consumers of shared ontologies from having knowledge of each other's existence and the details of their implementations. On the other hand, this architecture style also keeps details of shared data-repository implementation a secret from the producers and consumers. This secret is embodied in abstract interfaces to the data repositories that further reduce the coupling between data producers and consumers.

4 ARCHITECTURE FORMALIZATION

For studying the runtime-behavior of a *BPaaS-HUB* architecture, we translate the conceptual architecture of Figure 3 to a formal notation, namely, a Colored Petri-net (CPN) (Jensen, 1992; Jensen, 1996). CPN is a graphical oriented language for design, specification, simulation and verification of systems such as communication protocols, distributed systems, automated production systems. Informally, the CPN notation comprises states, denoted as circles, transitions, denoted as rectangles, arcs that connect states and transitions but never states with other states or transitions with other transitions, and tokens with color, i.e., attributes with values. Arcs carry an inscription in a CPN-ML expression that evaluates to a multiset or a single element. We use CPNtools[2] for designing, evaluating and verifying the models[3] in this paper.

The full formalization of the conceptual *BPaaS-HUB* architecture in Figure 3 is shown in Figure 4. Note that double-lined transitions are so-called substitution transitions with lower-level refinements. The model in Figure 4 answers one research question stipulated in the introduction of this paper, namely what is the protocol of a *BPaaS-HUB* service bus. Thus, the input- and output states of the substitution transitions in Figure 4 are the protocol of that service bus. Note that all substitution transitions of Figure 4 are located above the service bus inside the controller-layer. The atomic transitions correspond to components in Figure 4 located below the service bus.

In Table 1, all token colors of the top-level *BPaaS-HUB* formalization we show as variables with corresponding descriptions. For all token colors listed in tables, their types are given in their corresponding CPN-models in association with the respective states these tokens reside in.

In the remainder of this section we first explain fundamental properties of CPN that are relevant for the analysis. Next, we show formalizations in CPN notation of components in Figure 3 that are separated by the concerns who, with, what, how. For each CPN-model, the states, transitions and token colors are described in coherence with the explanations given for the conceptual *BPaaS-HUB* architecture. In the subsections below, after an explanation of the formalized components follows a summary of the state-space analysis from the appendices of this paper.

4.1 GOAL DECOMPOSITION

The starting point for interaction with the system is a process of goal decomposition for searching service provisions. If a human user interacts with the system, the goal discovery and decomposition happens in an intuitive way. However, if a software agent interacts with a *BPaaS-HUB*, the goal decomposition must be more formalized, which the model in Figure 5 caters for. The variables in the model for goal decomposition in Figure 5 are fully contained in Table 1. The model assumes that users have a predetermined number of goals they are aware of for searching their service provision. A corresponding amount of concrete, chosen goals are assembled into a goal hierarchy for which we assume there exist horizontal, vertical and lateral relationships that influence each other positively or negatively.

4.2 WHO

Within the *Who*-column, relevant terms are extracted from two sources, one being databases within a *BPaaS-HUB*, and more terms from customer documentation. For doing so, *BPaaS-HUB* comprises two respective components. All token colors for the *HUBTermExtraction*-component in Figure 6 are listed in Table 1. However, for the *CloudTermExtraction*-component, Table 2 lists not yet mentioned token colors.

The formalized *HUBTermExtraction*-component of Figure 6, starts with picking an earlier created goal hierarchy for a respective *BPaaS-HUB* user. A need for keywords is calculated to search for services that satisfy the goals and the first are assigned on the top-level of a *BPaaS-HUB* assuming that the human or software user provides a concrete number of search keywords. On the top-level, *BPaaS-HUB*-internal information about services offers, service-issuing organizations and persons involved returns to the *HUBTermExtraction*-component where it must first be determined whether a generated service-option is acceptable for further processing. Simultaneously, automated term-detection and extraction happens based on the human-readable, textual service description.

[2]http://cpntools.org

[3]http://www.cs.helsinki.fi/u/anorta/BPaaS-HUB.cpn

Variable	Explanation
user	Human or software agent who interacts with a *BPaaS-HUB* for finding a trustworthy service provision.
goalamount	Amount of objectives pursued for finding service provision.
goal	An individual objective of a user that influences service-provision search.
sopDB	Entries in the DB with data about services, their issuing organizations, and involved persons.
collabDBresult	When terms are invoked as keywords, the amount of textual descriptions returned about services, their issuing organizations, and persons.
conf	True once a multilingual ontology repository is configured for the domain of service-provision exploration.
kneed	Amount of keywords needed to search for service provision.
ontorep	A particular ontology repository.
da	Amount of documents.
document	*BPaaS-HUB*-external documents injected for enhanced ontology-repository configuration.
term	Concrete term extracted from an external document that has not yet been used for ontology-repository configuration.
docuterm	Concrete term extracted from an external document that has already been used for ontology-repository configuration.
il	Information library used for mashup generation.
x,y	Counter variables for preliminary concepts and document terms respectively.
seoffer	Service offer.
serequest	Service request.
option	A boolean and if true, it means the social mining for trust and reputation results in an approval for a service request.
terms	Amount of terms that need to be considered for establishing multilingual categories and ontology repositories.
category	A category for types of terminology.
lanID	Identification number for a specific human language.
processed	If true, a term has been processed for multilingual category analysis to establish ontology repositories.

Table 1: Token colors of a formalized *BPaaS-HUB* top-level.

Note, for simplicity, we refer to service description but that also comprises descriptions of the service-issuing organizations and persons involved. Additionally, the HUBTermExtraction-component sends a first enabling flag for the following *CloudTermExtraction*-component that we describe below. That is important as it is also possible that no generated service-description is chosen for further *BPaaS-HUB* processing. If the user chooses a concrete generated service, that information is also passed on for the next components together with a counter for the amount of chosen services and corresponding information for a *BPaaS-HUB* log.

Next, in the *HUBTermExtraction*-component, the chosen textually described services need to be further processed for automated term extraction with tool support. When that processing completes, a set of terms from a *BPaaS-HUB* domain comes into existence for the next steps in service exploration. The term extraction continues eventually with mashed up text generated from the web-cloud. Finally, there is one special situation that the HUBTermExtraction-component must cater for. When one service is left over for term extraction but the latter step fails, the actual automated term extraction must be sidestepped and the subsequent components enabled.

The external *BPaaS-HUB* terminology extraction is taken from mashed up information about services that stems on the one hand, from the web-cloud and one the other hand, from so-called customer documents delivered by customers who want to discover a specific service provision. The *CloudTermExtraction*-component depiction of Figure 7 caters for these two terminology sources. The objective of this component is to contribute to an ad-hoc ontology-repository creation for the universe of encountered terminologies. Not yet described token colors we explain in Table 2. While the terms from a *BPaaS-HUB* domain are taken into account for this objective, the *CloudTermExtraction*-component also extracts automatically terms from web-cloud sources that are mashed

Variable	Explanation
optcon	Concept options for terms.
context	Taken into account classification context for a set of terminologies.
pattern	For a set of terms in a context, a harvested pattern for term classification.
clas	Counter for the amount of classes for a set of terminologies.
ontrepID	An identification number for a newly created ontology repository to provide terminology concepts and properties.
approved	A counter for amount of approvers of ontology repositories.

Table 2: Token colors of the formalized *CloudTermExtraction*-component.

Variable	Explanation
member	An expert who is in an expert group for voting on a term definition.
agree	May be true or false depending on the decision of a member of the expert group.
s, c	Counter variables.
v	A counter for all casted votes.

Table 3: Token colors of the formalized *TermVote*-component.

up on the fly into human-readable text. Additionally, specific high-level customer documents may be inserted in the service-provision-search process with very specific terminology. A *divert* step in the *CloudTermExtraction*-component ensures that at least one term enters the generation of an ontology repository. Finally, automatically detected terminology must be discussed and voted in by a group of experts, which is located in a component explained below.

Terminology detected from customer-injected documentation must be discussed by an expert group for agreeing on a definition. This process is carried out in the *TermVote*-component for which not yet listed token colors we explain in Table 3. The prerequisite for commencing a voting process on terminology is the establishment of an expert group from a pool of candidates. A term from customer documentation that has not yet been discussed, is picked and available for definition discussion by the chosen members of the expert group. Eventually, every expert states whether she agrees with a definition or not. If a particular agreement threshold is not met, the entire voting term-definition discussion must be repeated and agreement votes casted again until the threshold is met. In the latter case, the status of a respective term is changed to sufficiently approved by the experts and it can be used for the creation of an ontology repository.

4.3 WITH

In this column, the *OntoRepBuild*-component of Figure 9 comprises four refined components that we explain in the sequel of this section. The general process for building ontology repositories commences with the discovery of concepts for the harvested set of terms that take into account that these terms may belong to different human languages. Next, the discovered concepts are classified before a release to a repository. After analyzing the terminology context, the concepts are assembled into ontology repositories and finally transferred into an ontology configuration. Table 4 lists and explains the token colors of the *OntoRepBuild*-component and all contained components.

The *ConceptDiscovery*-component of Figure 10 shows that two sources exist for concepts that enter the ontology-repository creation. Those are the one hand, concepts harvested from terms that are not covered so far by ontology repositories and on the other hand, concepts from large documents and online feeds. To use these concepts for ontology creation, the context of a set of terms must be detected too. The analysis report from Appendix 7 shows that the *ConceptDiscovery*-component has no cycles and that the transitionsare the most stressed for discovering concepts either from uncovered terms or from large documents .

The *BuClReConcept*-component of Figure 11 first builds concept classes that are solidified in a subsequent step with taking patterns and contexts into account. First, the concept is preliminary and later released as a final step. In the *HarRevChAs*-component of Figure 12, the harvested concepts must be reviewed by human experts and agreed upon taking patterns and the context into account. When an approval exists, the ontology repository

Variable	Explanation
pattern	Found in a set of terms that make the latter fit into a concept class.
context	Required for a term's concept-class assignment.
da	Counter for amount of documents.
approved	A human reviewer agrees with a created concept class for a set of terms based on the detected context.
release	A counter for all approved ontology repositories that need to be assembled.
cats	Amount of categories.

Table 4: Token colors of the formalized *OntoRepBuild*, *BuClReConcept* and *MultiLangMan*-components.

Variable	Explanation
mashed	A boolean to determine whether documents have been successfully mashed up for display to the user.
mdocu	An mashed document for user display.
feed	A detected document from the web cloud that relates to a service in a *BPaaS-HUB* under investigation.
done	A boolean that indicates an additional information library has been chosen.

Table 5: Token colors of the formalized *Mashup*-components.

is assembled and available.

Since terms may stem from different human languages, the *MultiLangMan*-component of Figure 13 checks terms and extends generated categories for human languages. These categories are related to each other in hierarchies.

4.4 WHAT

In this column, a *BPaaS-HUB* uses the *Mashup*-component for allowing users to explore background information from the web-cloud to estimate the trustworthiness and reputation. The not yet explained token colors of this component we list in Table 8. This component mashes up the processed text about service offers into one for display to the user who wants to further explore the trustworthiness and reputation of that respective service. Thus, a user may choose to consult additional offered information libraries with high-quality, specific data about a service under investigation. Once such dedicated libraries are chosen, *BPaaS-HUB* checks for service-related information and presents it too in the mashup to the user. For all functionalities of the *Mashup*-component, the ontology configuration is utilized.

4.5 HOW

The last component in the lifecycle of *BPaaS-HUB* interaction is the *CarryOut*-component of Figure 15. Here, the trust and reputation of a chosen service is further explorable with mining the historical experience-data harvested within a *BPaaS-HUB* domain from past service use. If the service is concretely chosen, a matching with the user's service request is performed in accordance with the approach of Figure 2. If the matching fails, a request is issued to modify the service provision until it adheres to the specification of the service request. Finally, the

Variable	Explanation
choose	A boolean to indicate if a chosen service-provision is picked for enactment.
continue	A boolean to determine if a chosen service-provision enactment should be terminated or not.
matches	A boolean that indicates a chosen service offer provision adheres to a service request specification following Figure 2.

Table 6: Token colors of the formalized *Mashup*-components.

service enactment commences until it is not continued any more and the service-provision choice is moved to the final state of completion where the lifecycle ends.

5 PERFORMANCE ANALYSIS OF THE FORMALIZED ARCHITECTURE

With the model-checking functionality of CPNtools, analysis reports are generated automatically (see appendices). For brevity, only compressed explanations of analysis properties are given here but in (Jensen et al., 2007) the reader finds extended discussions.

5.1 PROPERTIES OF CPN STATE-SPACE ANALYSIS

The formalized *BPaaS-HUB*-models of this section are translated into so-called respective *state spaces* for performing analysis. The basic idea underlying state spaces is to compute all reachable states and state changes of the CPN-model and represent these as a directed graph where nodes represent states and arcs represent occurring events. Next, the state-space graph is translated into a *strongly connected component graph* (SCC-graph). The nodes in the SCC-graph are subgraphs called strongly connected components (SCCs) and informally explained, free of loops that may be contained in the state-space graph. The structure of the SCC-graph comprises useful information about the overall behavior of the model being analyzed.

Following the state-space analysis reports in the appendices, the checked properties we informally explain as follows. If the number of nodes in the state space and SCC-graph is equal, it means the state space is free of circles that could result in the model not terminating. The *boundedness properties* tell how many (and which) tokens a place may hold when all reachable markings are considered. The *best upper integer bound* of a place specifies the maximal number of tokens that can reside on a place in any reachable marking. The *best lower integer bounds* for a place specifies the minimal number of tokens that can reside on the place in any reachable marking. The *best upper multi-set bound* of a place specifies for each color in the color set of the place the maximal numbers of tokens that is present on this place with the given color in any reachable marking. The *best lower multi-set bound* of a place specifies for each color in the color set of the place the minimal number of tokens that is present on this place with the given color in any reachable marking.

The home properties tell us that there exists a single *home marking* M_{home}, which can be reached from any reachable marking. This means that it is impossible to have an occurrence sequence which cannot be extended to reach M_{home}. In other words, it is not possible to end up in a situation that makes it impossible to reach M_{home}.

The *liveness properties* cover several aspects. A transition is *live* if from any reachable marking we can always find an occurrence sequence containing the transition. If every transition is life then a CPN is life in its entirety. A *dead marking* is part of the liveness properties, which is a marking where no binding elements are enabled. A dead marking can be a home marking because any marking can be reached from itself by means of the trivial occurrence sequence of length zero. A transition is *dead* if there are no reachable markings in which it is enabled. If a model has dead transitions, it corresponds to parts of the model that can never be activated. Hence, we can remove dead transitions from the model without changing the behavior of it.

The motivation for the *fairness* property is to detect the transitions in a CPN that can not fire infinitely often while being enabled infinitely often. There are four fairness notions, namely, *impartial* if a transition occurs infinitely often in every infinite run of a CPN. A transition is *fair* if it occurs infinitely often in every infinite run of the net where the transition is enabled infinitely often. A *just* transition occurs infinitely often in every infinite run of the net where it is continuously enabled from a marking onward. Finally, a transition is *notfair* if it is not just. Impartial considers all infinite runs while fair and just only consider some infinite runs.

5.2 RESULTS OF STATE-SPACE ANALYSIS

From a developer perspective, the motivation for analysing the models in Section 4 is to see if they terminate correctly, which is testable with a token game of the *BPaaS-HUB* models in CPNtools. Secondly, detected loops in a model means the system implementers must think carefully about enforceable termination criteria. Detected performance peaks mean, during runtime, provisions must be in place for elastic resource assignment, which is important in cloud-computing environments.

The practical relevance of liveness checks mean for dead transitions that never needed functionality is present in a component, which is undesirable as it does not contribute to automation efforts. Live transitions are functionalities of a component that are used at least sometimes. This means that system implementers must ensure for high runtime robustness of such functionality. If there is no consistent home marking, developers should expect increased testing efforts of developed components.

A summary of the analysis results we provide in Table 7 where the first column lists the components of the BPaaS-HUB. When a component represents a hierarchical refinement, to the left a *s* marker is positioned. Loops exist when the the state space has more nodes and arcs than the SCC-graph. If the boundedness properties reveal spikes in token numbers and the liveness properties of transitions show differences, performance peaks for respective transitions are given, which is indicated with a corresponding transition label. In the sequel, we provide textual explanations of those performance peaks. The liveness column shows that no component comprises dead transitions that never fire but on the other hand, no transition is live, i.e., fired in any marking. Finally, the last column states whether a component has a consistent final marking, i.e., home marking that is also a not varying dead marking.

	Component	Loops	Performance Peaks	Liveness	Varying Termination
	GoalDecomposition	no	goal decomposition	ND/NL	no
	HUBTermExtraction	no	extract terms	ND/NL	yes
	CloudTermExtraction	yes	configure ontology repository, ontology configuration from documents	ND/NL	yes
s	TermVote	yes	pull crowd	ND/NL	yes
s	ConceptDiscovery	no	all concept discovery transitions	ND/NL	yes
s	BuClReConcept HarRevChAs	yes	check approved, assemble ontology repository, check approved, assemble ontology repository	ND/NL	yes
s	MultiLangMan	yes	check approved	ND/NL	yes
	Mashup	no	no	ND/NL	yes
	CarryOut	yes	no	ND/NL	yes

Table 7: Summary of analysis results.

The result of the automated model analysis for the *GoalDecomposition*-component from Appendix A shows that the number of nodes and arcs of the state space and the SCC-graph are the same, which means there can not be infinite occurrence sequences that will never terminate. From the depiction in Figure 5, the transition labeled *goal decomposition* is the most stressed given the best integer bounds. Furthermore, the home marking corresponds to the dead marking, which underlines that the goal decomposition terminates finitely. All transitions fire in some marking but none is never fired as the liveness properties reveal.

The analysis of the *HUBTermExtraction*-component in Appendix B shows there are no cycles included as the number of nodes in the state space and and SCC-graph are the same. The boundedness properties show performance demands are highest for the transition labeled *extract terms*. While all transitions fire in at least one marking, the final marking of the *HUBTermExtraction*-component varies.

The analysis report for the *CloudTermExtraction*-component in Appendix C reveals that the SCC-graph has less nodes than the state space. Thus, it is possible to end up in a cycle that does not terminate. This test result is realistic when one takes into account the scenario of information overflow from the cloud for a particular service under investigation. To prevent information overflow, it is important to ensure a search-exit criteria is in place.

With respect to performance, the analysis results in Appendix C state it is computationally most expensive to configure ontologies as it is possible to have a large number of not yet covered terminology resulting from the web-cloud. Furthermore, the *CloudTermExtraction*-component does not have a unique termination as no home marking exists, which is underlined by the many possible dead markings. Since there are neither dead nor live transitions, we conclude all are used at least once in a possible marking. However, the fairness properties have differentiating results. Since *divert* and *extraction* are fair while the remaining transitions in the *CloudTermExtraction*-component are not fair, specific focus should be placed on the two mentioned functionalities during development and sufficient computing-resource assignment to them during runtime.

The analysis results from the *TermVote*-component also reveal possible loops without exit as the SCC-graph has less nodes than the state space. The home and liveness properties in Appendix C1 support that conclusion. The fairness properties show that all transitions are impartial with the exception of the *pull crowd* transitions. Thus, spiking performance demand in the *TermVote*-component is possible.

For the *ConceptDiscovery*-component, the analysis result in Appendix D1 show that no cycles exist as the number of nodes and arcs in the state space and the SCC-graph are the same. The biggest performance stress is on the transitions for the concept discovery from uncovered terms and from large documents. All transitions of the *ConceptDiscovery*-component are used at least in one marking while the outcome result in terms of final markings is not determined but may vary.

For the components named *BuClReConcept* and *HarRevChAs* are analyzed together and the results are in Appendix D2. They show that although the state space and the SCC-graph have the same number of nodes, the latter has less arcs, which tells that circles are existent that may not terminate. A circle does occurs when a harvested concept is never approved by a human review. With respect to performance demands, the analysis report in Appendix D2 shows that the transition labeled *check approved* in *HarRevChAs* is the most stressed followed by *assemble ontology repository*. In the *BuClReConcept*-component, the transitions for building concept classes and their classification must perform best. The remaining transitions are not fair and as such less stressed when the *BuClReConcept* and *HarRevChAs*-component are used. Finally, the analysis results in Appendix D2 show that all transitions are fired in at least one type of marking and there is no unique termination marking.

The analysis results for the *MultiLangMan*-component in Appendix D3 show there are less arcs in the state space than in the SCC-graph. Thus, non-terminating loops of the *MultiLangMan*-component are possible. The amount of tokens is biggest in the state *multilingual category* while the most performing transition is labeled *check approved*. While all transitions are used in at least one marking, the outcome is not deterministic of running the the *MultiLangMan*-component.

For the *Mashup*-component of Figure 14, the analysis results in Appendix E reveal no cycles exist and following the boundedness properties, performance demands are evenly levelled. All transitions fire in at least one marking and the final marking may vary.

For the analysis of the *CarryOut*-component of Figure 15, the results in Appendix F show a loop exists as the number of nodes in the state space is bigger than in the SCC-graph. Furthermore, the performance demands are balanced, all transitions are fired in at least one marking and the terminal marking may vary.

6 INSTANTIATION STUDY OF *BPAAS-HUB*

In the framework of the ContentFactory[4] (CF) research project, we implement a *BPaaS-HUB* architecture from Section 3 for conducting case studies with industry. The evaluation compares the requirements postulated in Section 2.3 with a *BPaaS-HUB* architecture and gives applications from the ongoing implementation.

For satisfying Requirement 1, the HUB-architecture comprises a *View* layer with several graphical user-interface components. In the CF-project, we implement a user friendly business-service registry termed Collab[5] that links stored service data of service offers and requests with service-responsible persons and service-issuing organizations. Collab stores service-experience ratings from users for reputation assessments. For keyword extraction, Collab sends the free-text description to the Likey (Paukkeri et al., 2008) application.

For Requirement 2, a mashup component is part of a *BPaaS-HUB* architecture. We consider the PULS (Yangarber and Steinberger, 2009) application for populating the mashup component. Currently, PULS surveils, prunes, ontologically ranks, and aggregates large amounts of online news for surveilling the spread of emerging diseases. However, ongoing PULS extensions cater for an in-depth exploration of domain-specific patterns for business domains such as acquisition, takeover and buyout, investment, nomination, new product release, innovation, marketing, ownership/stake; divestment/reduction of stake.

For Requirement 3, a *BPaaS-HUB* architecture includes components for creating ontology libraries. We use the TermFactory[6] application for allowing terminologists to define extracted keywords that enter ontology libraries for respective HUB-application contexts.

The matching component in the HUB-architecture satisfies Requirement 4 and would incorporate matching levels as described in Section 2.2. Currently, we implement an application for realizing the matching heuristics in (Eshuis and Norta, 2009). In a first version, the BPEL-representations of one service offer and one service request enter the matching application and converted process trees are compared for their similarity.

For Requirement 5, the HUB-architecture includes logging components for several stages of user interaction and a social mining component for the extraction of business intelligence. As an example for populating the

[4] ContentFactory, funded by Tekes (Finnish Funding Agency for Technology and Innovation), http://www.verkko-ope.net/cf/

[5] http://db.cs.helsinki.fi/∼tkt_coll/collab/

[6] http://www.helsinki.fi/∼lcarlson/CF/doc/TFManual.html

Variable	Explanation
choose	A boolean to indicate if a chosen service-provision is picked for enactment.
continue	A boolean to determine if a chosen service-provision enactment should be terminated or not.
matches	A boolean that indicates a chosen service offer provision adheres to a service request specification following Figure 2.

Table 8: Token colors of the formalized *Mashup*-components.

mining component, the ProM framework[7] could allow the extraction of processes from logs in a *BPaaS-HUB* to explore what interaction steps lead to popular service matches. ActiveBPEL[8] is an open-source option for populating the *Enact* component.

6.1 APPLICATION-ASSIGNMENT TO *BPAAS-HUB* MODEL

For the formalized model of a *BPaaS-HUB* in Section 4, the industry-applications listed above are assigned in Table 9 to respectively formalized components. The first listed industry-applications in Table 9 is the *TermMANAGER*[9], which is a browser-based tool designed for term management, user management, the term work process and enhanced translation. *TermMANAGER* is designed to support the utilization of a company's term data for the purpose of enhancing translation and improving quality. The second listed industry-application in Table 9 is *TermFinder* for language- and communication-skill training with connected electronic treatment. That way it is possible to quickly build tailored terminology databases with the help of an extensive glossary of concepts in different languages.

		Application				
	Component	**TermMANAGER**	**TermFINDER**	**TermFactory**	**PULS**	**Likey**
	GoalDecomposition	X				
	HUBTermExtraction	X	X			X
	CloudTermExtraction	X	X			X
s	TermVote	X				
	OntoRepBuild			X		
s	ConceptDiscovery			X		
s	BuClReConcept			X		
s	HarRevChAs			X		
	MultiLangMan				X	
	Mashup				X	X
	CarryOut					

Table 9: Assignment of existing industry applications to the formalized *BPaaS-HUB* model.

With respect to the scientific applications in Table 9, *TermFactory* provides access to multilingual resources and the collaboration management of such resources with a toolset that enables a distributed development of domain-specific terminologies suitable for human- and machine use. In Table 9, *TermFactory* is assigned to cover the functionalities model in the components for building on-the-fly ontology repositories that require terminology-concept and context discovery.

PULS focuses on the multi-lingual management of document collections and tested business-news streams. The multi-lingual abilities comprise automatic learning of classes of domain-specific terms and concepts; the automatic learning of context-based patterns that capture relationships among domain concepts; and additionally, the tracking and organization of events found in dynamic document collections such as continuous news streams. In Table 9, *PULS* is assigned to cover the multi-language management of harvested terminologies. Furthermore, *PULS* is suitable for mashup generation of servce-background information from the web-cloud.

Likey is a scientific application that offers techniques for terminology extraction of human-readable text-documents. In Table 9, we assigne *Likey* as a complementary terminology-harvesting tool from within the *BPaaS-HUB* domain and from the external web-cloud.

[7] http://prom.win.tue.nl/tools/prom/
[8] http://sourceforge.net/projects/activebpel/
[9] http://portal.aacglobal.com/servlet/portal/en_en/termienhallinta/tuotteet/termienhallintapalvelut/termTOOLS/tools.tpl

Finally, note that Table 9 does not include any applications for covering the functionalities in the *CarryOut*-component. Thus, since the listed applications stem from CF-project partners, it means non cater for covering the latter component-functionality. However, many open-source and industry applications can be applied, e.g., ProM[10] for mining logged data, or WebSphere[11] for business-process enactment.

7 CONCLUSIONS

In this paper, we present an application architecture, the *BPaaS-HUB* for business-to-business collaboration that supports the discovery of trustworthy and reputable business-processes as electronic services. For the *BPaaS-HUB*, carefully extracted requirements are taken into account for the creation of a conceptual architecture that incorporates stiles and patterns. The conceptual architecture we translate into formalized colored Petri-net models that permit checking with verification methods. The results show the formal models terminate correctly and tell where loops occur, performance bottlenecks must be expected, whether all modeled functionality is used, and if runtime behavior that results in varying termination states suggests more strenuous testing efforts.

In answering the research questions, based on an empirically observed business-to-business collaboration model, the *BPaaS-HUB* requirements state that layman users must be enabled to intelligently find trustworthy service offers of acceptable reputation to match their deposited service requests.

For designing a high-quality architecture, a separation-of-concerns stile structures the *BPaaS-HUB* functionality along conceptual business-to-business collaboration interrogations. A layer-stile vertically separates the functionality to ensure it is possible to rapidly replace components on the fly. Since the *BPaaS-HUB* is designed for fitting into a cloud-computing environment, a service bus represents the component for realizing a pipes-and-filters pattern. Finally, an abstract-data-repository pattern facilitates the logging of service-related and *BPaaS-HUB*-usage data.

With respect to guidance through the *BPaaS-HUB*, a user first queries for a set of service provisions based on a number of related goals. The terminology in the textually described, found services are input for an ad-hoc ontology repository assembly that incorporate detected terminology concepts and contexts. For more detailed trust- and reputation exploration, information is pooled in from the web-cloud on the fly and offered to the user in a mashup. Further detailed trust-and reputation exploration stems from logged earlier searches for a respective set of services. Once a service provision is chosen after a matching with technical service-request details succeeds, the enactment commences.

For determining the protocol of the *BPaaS-HUB* service bus in the conceptual architecture, the formal colored Petri-net models deliver the answer. Thus, all input- and output states of the *BPaaS-HUB*-components with hierarchical refinements represent parts of the service-bus protocol.

For future work, we pursue the integration of identified applications for implementing the designed architecture into a proof-of-concept architecture and plan to conduct case studies with industry using the *BPaaS-HUB* for the discovery and matching of service offers and requests. We plan to explore HUB extensions for integrating a service-tendering procedure that allows users to place negotiable bids. Furthermore, the HUB will be studied as an integral component of a BPaaS breeding ecosystem within a cloud-computing environment for business collaboration. Furthermore, a user of a *BPaaS-HUB* will be confronted with interaction rollbacks to earlier stages of service discovery and evaluation. Since that has implications on the management of information and ontology configuration, it is interesting to explore at what stages and how interaction rollbacks can be performed without leaving behind orphaned data and processes that will result in a runtime breakdown.

[10] http://processmining.org/

[11] http://www-01.ibm.com/software/websphere/

APPENDIX A: GOALDECOMPOSITION

Statistics

```
State Space
    Nodes:  94
    Arcs:   309
    Secs:   0
    Status: Full

 Scc Graph
    Nodes:  94
    Arcs:   309
    Secs:   0
```

Boundedness Properties

Best Integer Bounds

```

                              Upper      Lower
GoalDecomposition'amount 1
                              1          0
GoalDecomposition'enable_next 1
                              1          0
GoalDecomposition'goal 1
                              7          4
GoalDecomposition'goal_hierarchy 1
                              1          0
GoalDecomposition'goal_pool 1
                              3          0
GoalDecomposition'pass 1
                              1          0
GoalDecomposition'user 1
                              1          0
GoalDecomposition'user_pool 1
                              1          0
```

Best Upper Multi-set Bounds

```
     GoalDecomposition'amount 1
1`(1,0,false)++
1`(1,1,false)++
1`(1,1,true)++
1`(1,2,false)++
1`(1,2,true)++
1`(1,3,true)
     GoalDecomposition'enable_next 1
                              1`1
     GoalDecomposition'goal 1
1`1++
1`2++
1`3++
1`4++
1`5++
1`6++
1`7
     GoalDecomposition'goal_hierarchy 1
1`(1,"hierarchy")
     GoalDecomposition'goal_pool 1
1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)++
```

```
1`(1,5)++
1`(1,6)++
1`(1,7)
     GoalDecomposition'pass 1
1`(1,0)
     GoalDecomposition'user 1
1`(1,1)++
1`(1,2)++
1`(1,3)
     GoalDecomposition'user_pool 1
 1`(1,3)
```

Best Lower Multi-set Bounds

```
     GoalDecomposition'amount 1
                         empty
     GoalDecomposition'enable_next 1
                         empty
     GoalDecomposition'goal 1
                         empty
     GoalDecomposition'goal_hierarchy 1
                         empty
     GoalDecomposition'goal_pool 1
                         empty
     GoalDecomposition'pass 1
                         empty
     GoalDecomposition'user 1
                         empty
     GoalDecomposition'user_pool 1
                         empty
```

Home Properties

Home Markings [94]

Liveness Properties

Dead Markings [94]
Dead Transition Instances None
Live Transition Instances None

Fairness Properties

No infinite occurrence sequences.

APPENDIX B: HUBTERMEXTRACTION

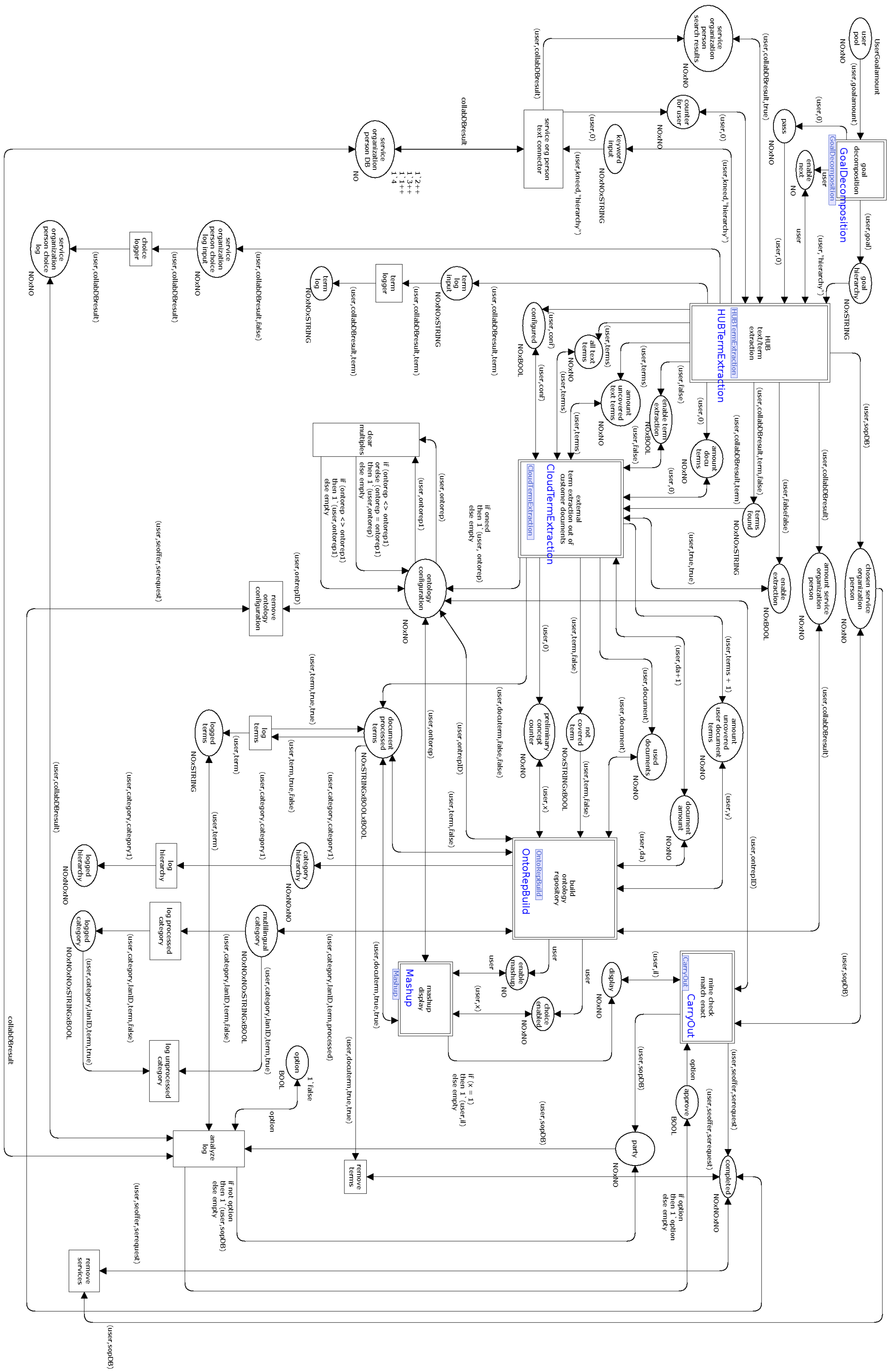

Figure 4: CPN representation of a *BPaaS-HUB*.

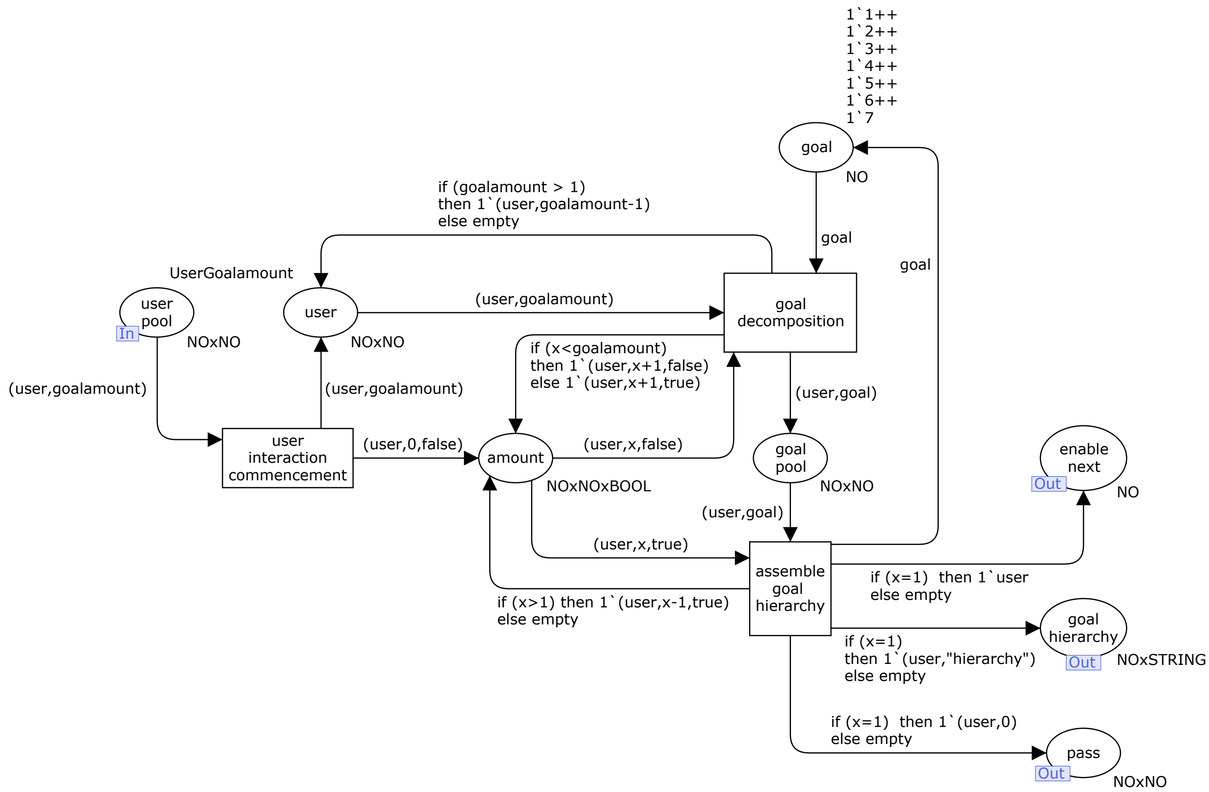

Figure 5: *GoalDecomposition*-component.

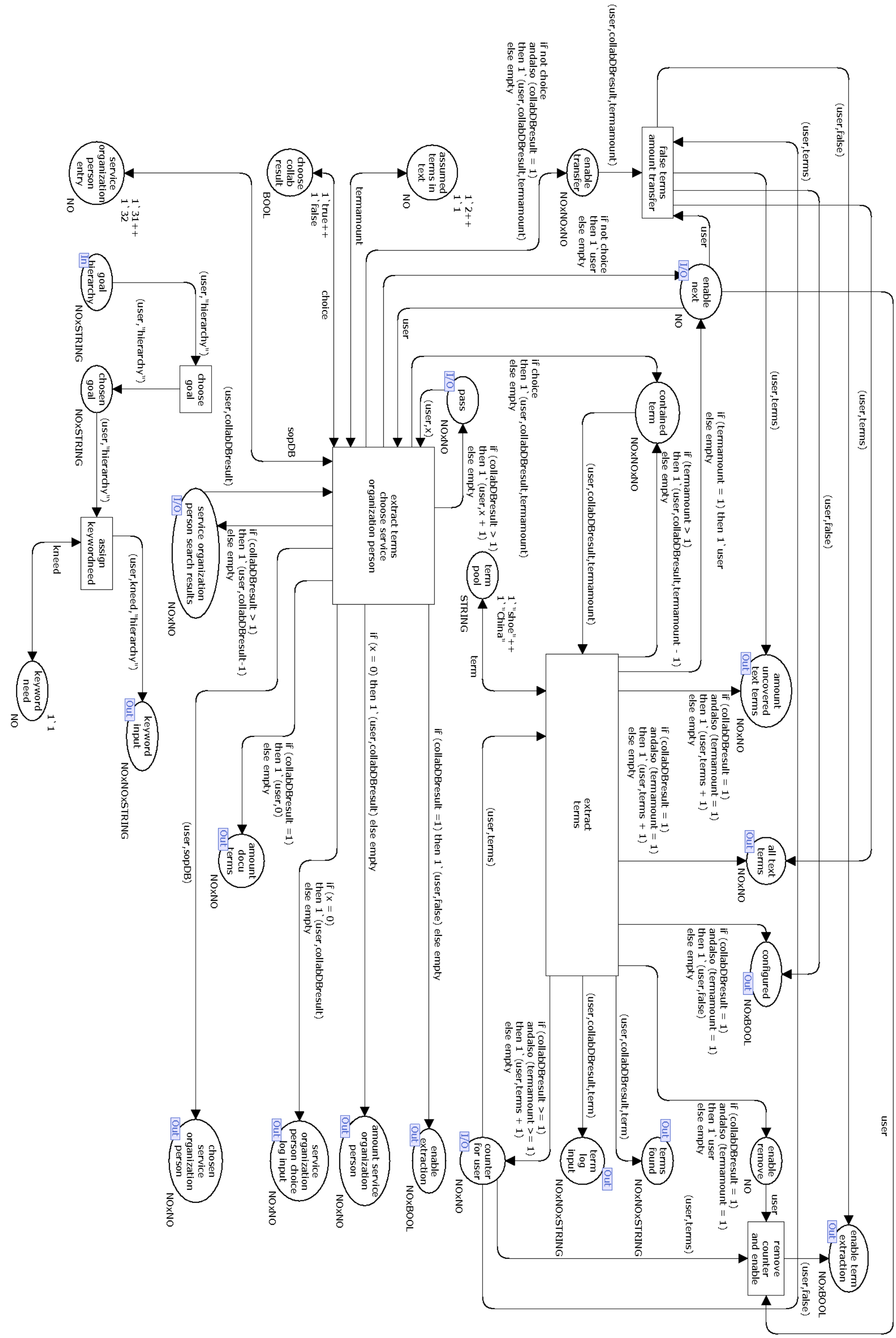

Figure 6: *HUBTermExtraction*-component.

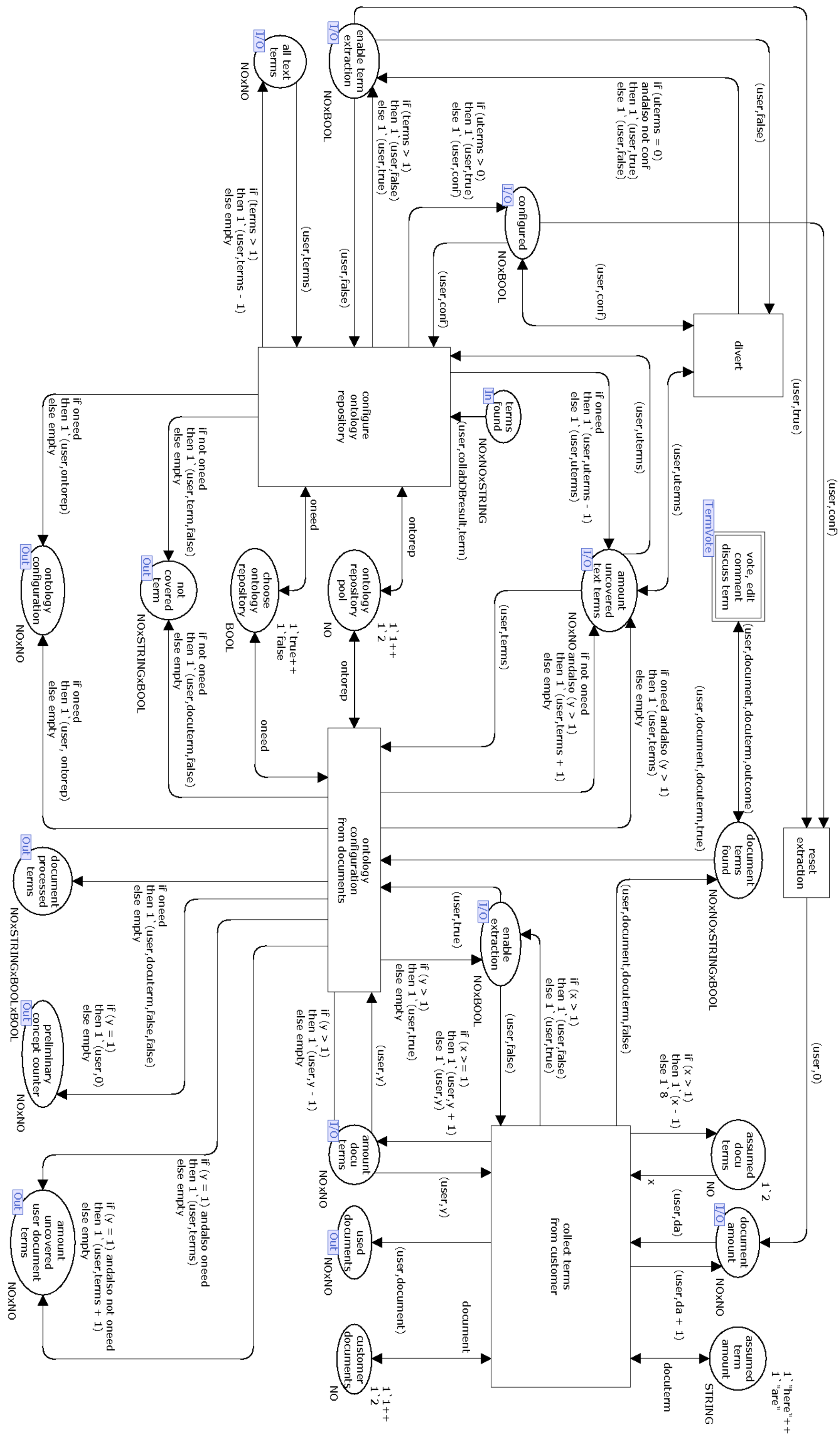

Figure 7: CloudTermExtraction component.

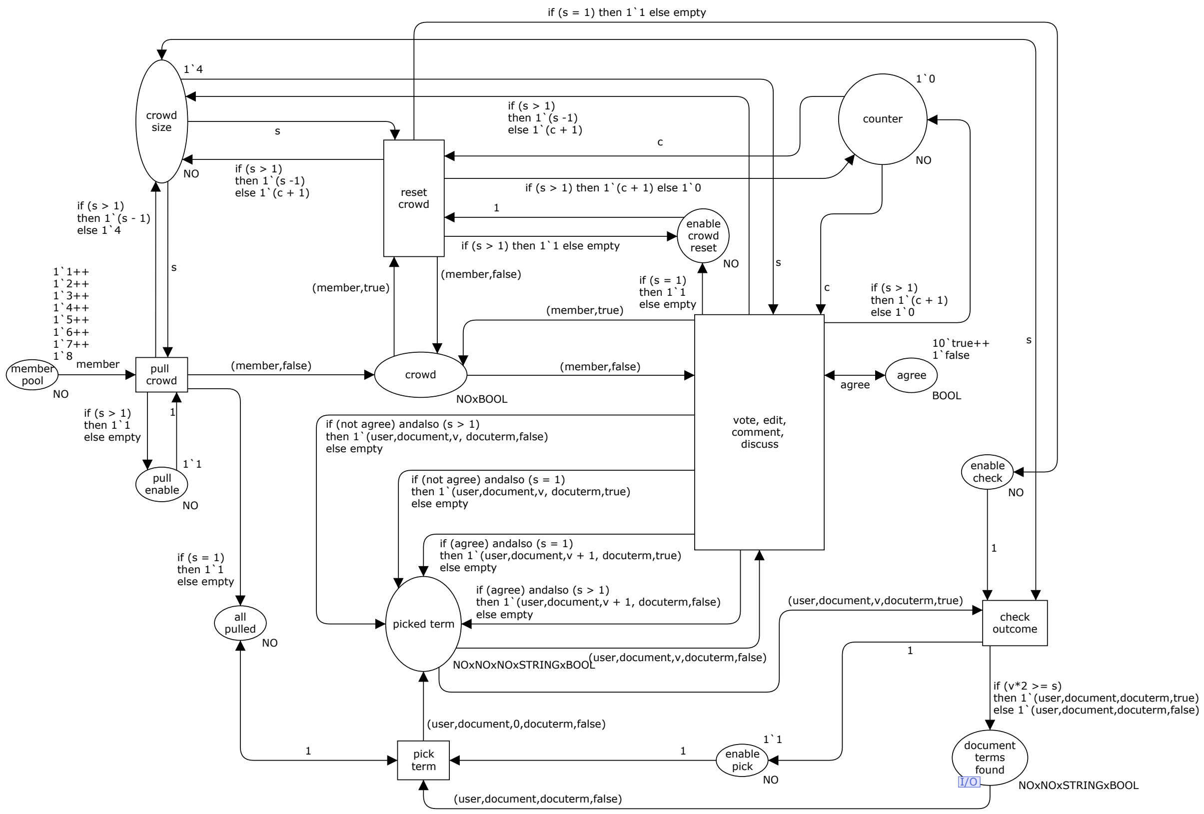

Figure 8: *TermVote*-component.

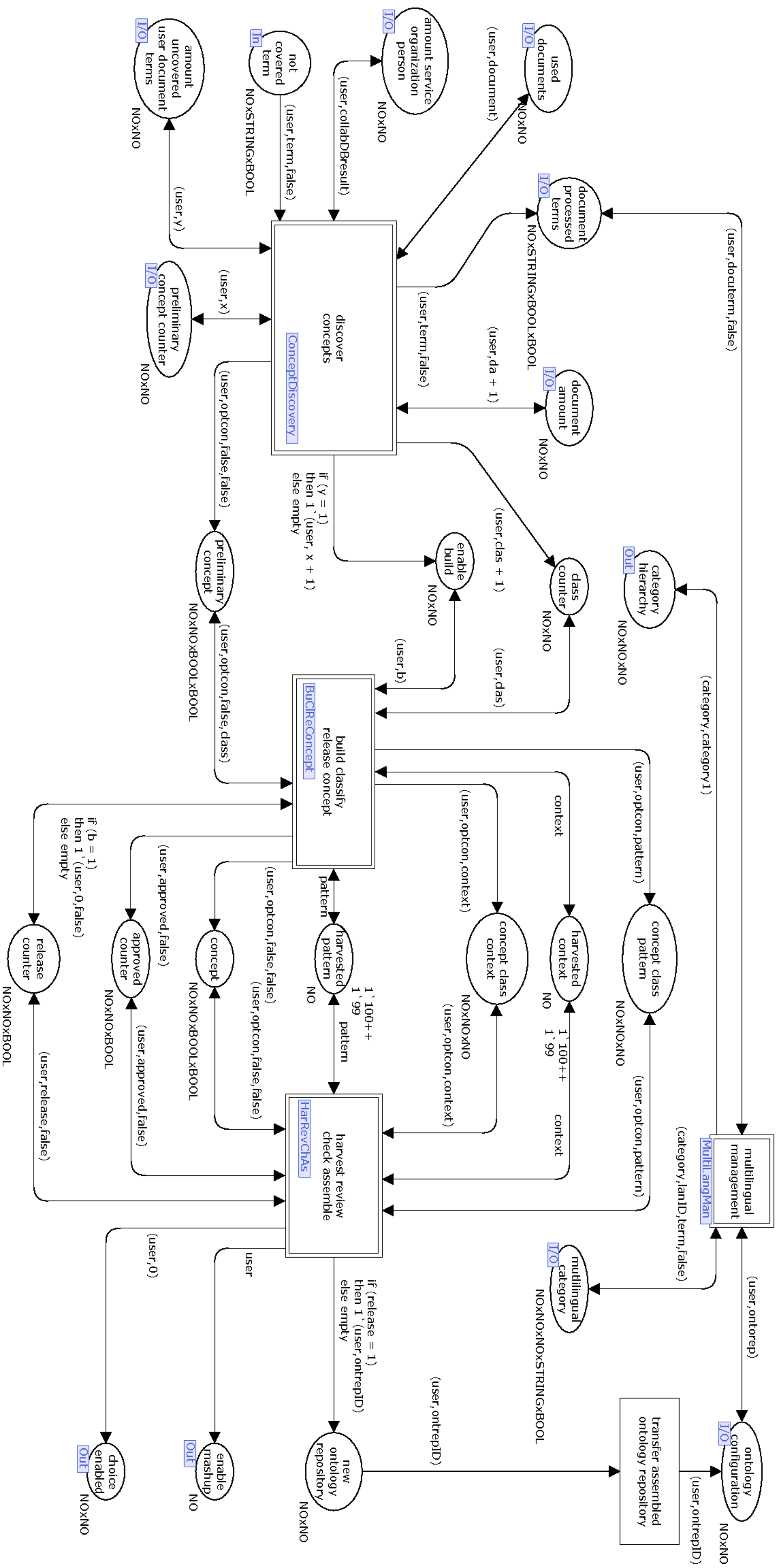

Figure 9: *OntoRepBuild*-component.

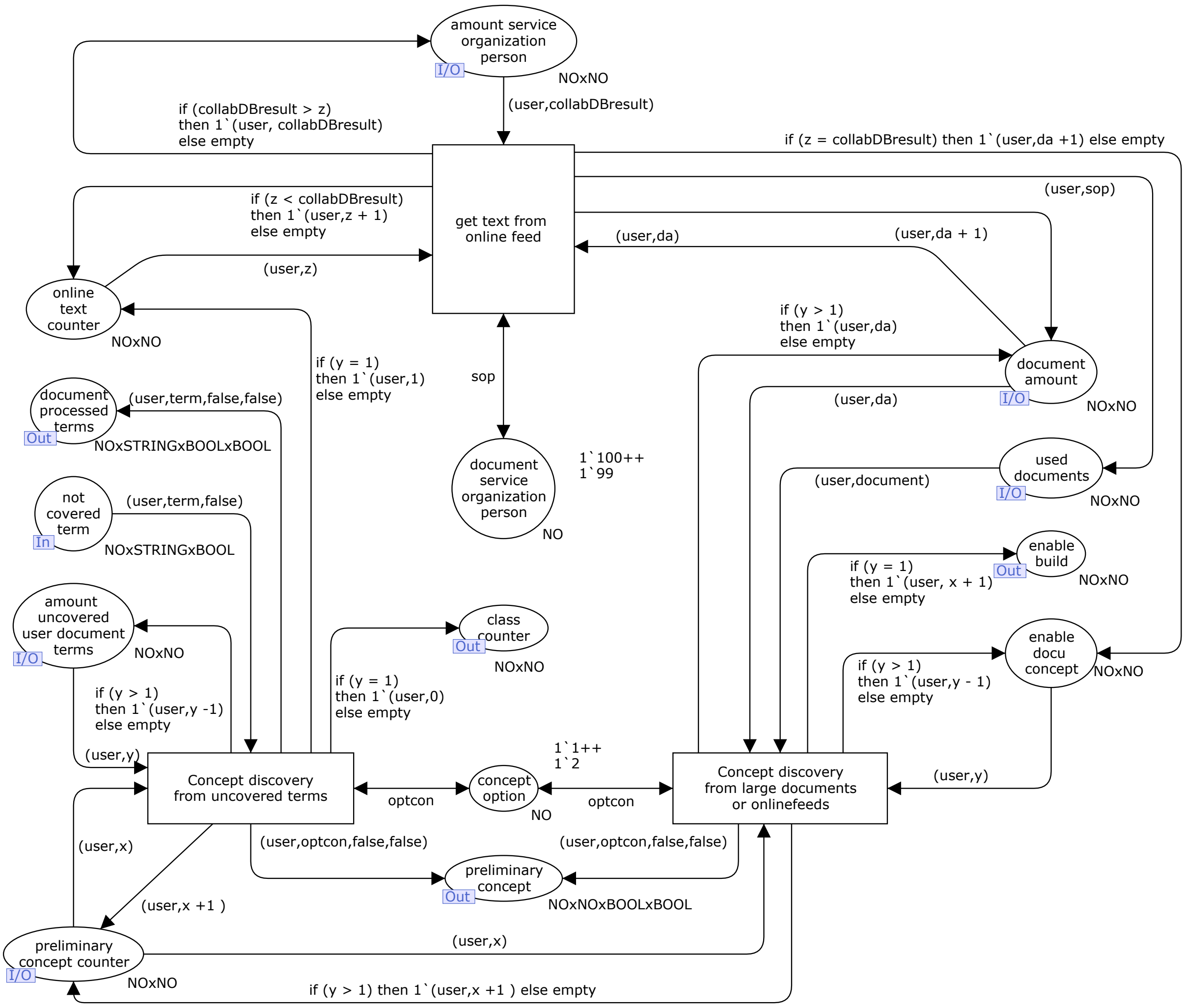

Figure 10: *ConceptDiscovery*-component.

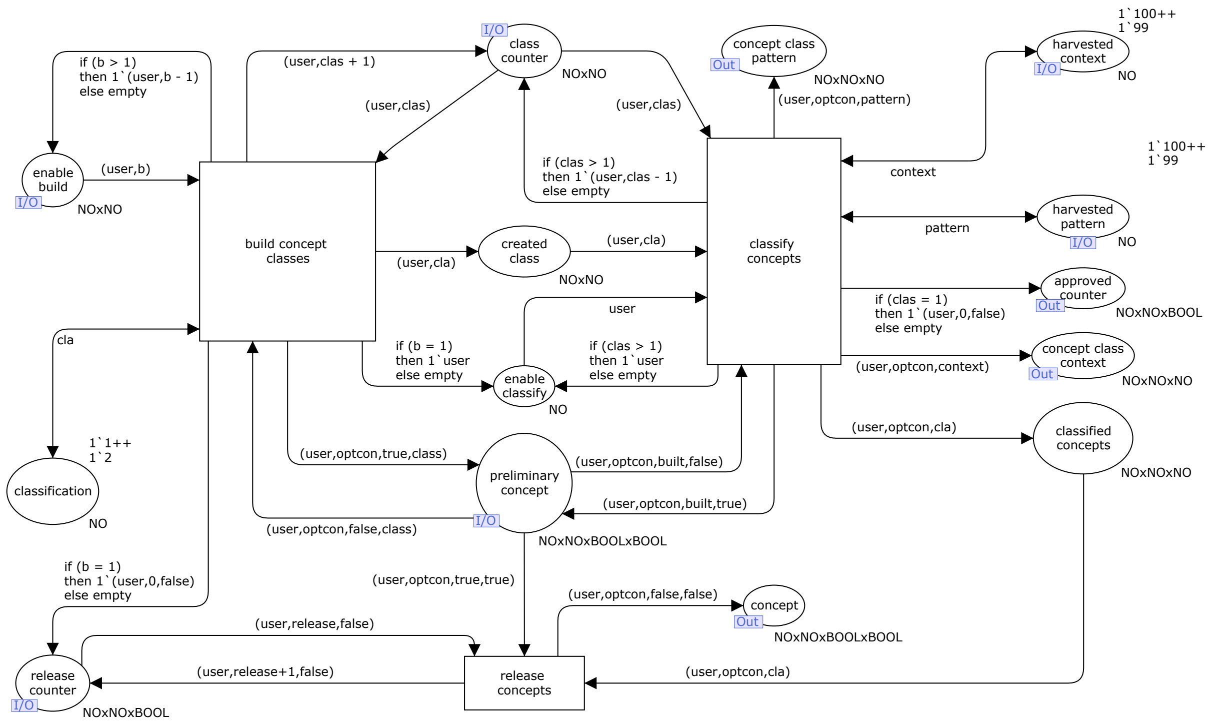

Figure 11: *BuClReConcept*-component.

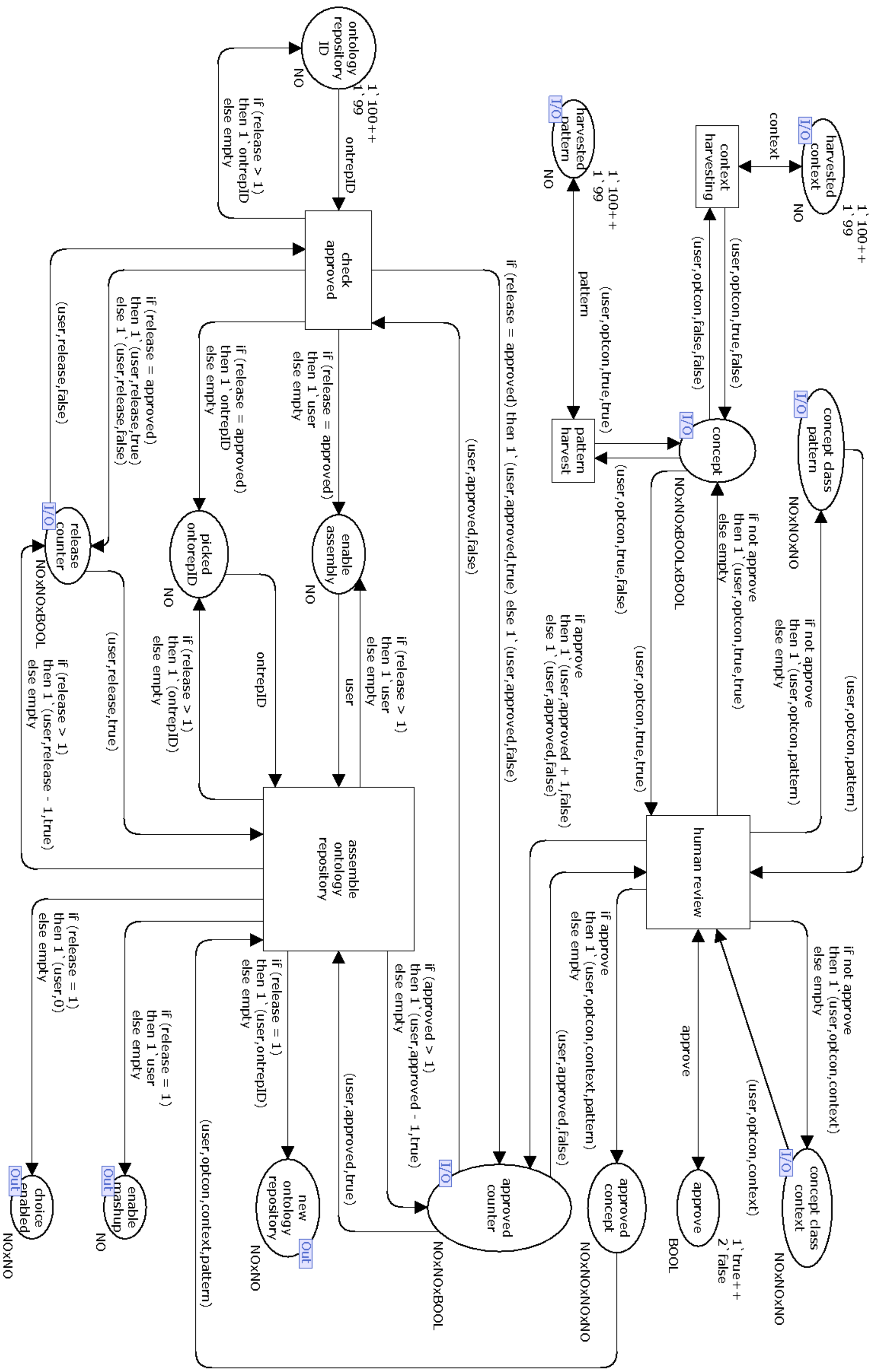

Figure 12: *HarRevChAs*-component.

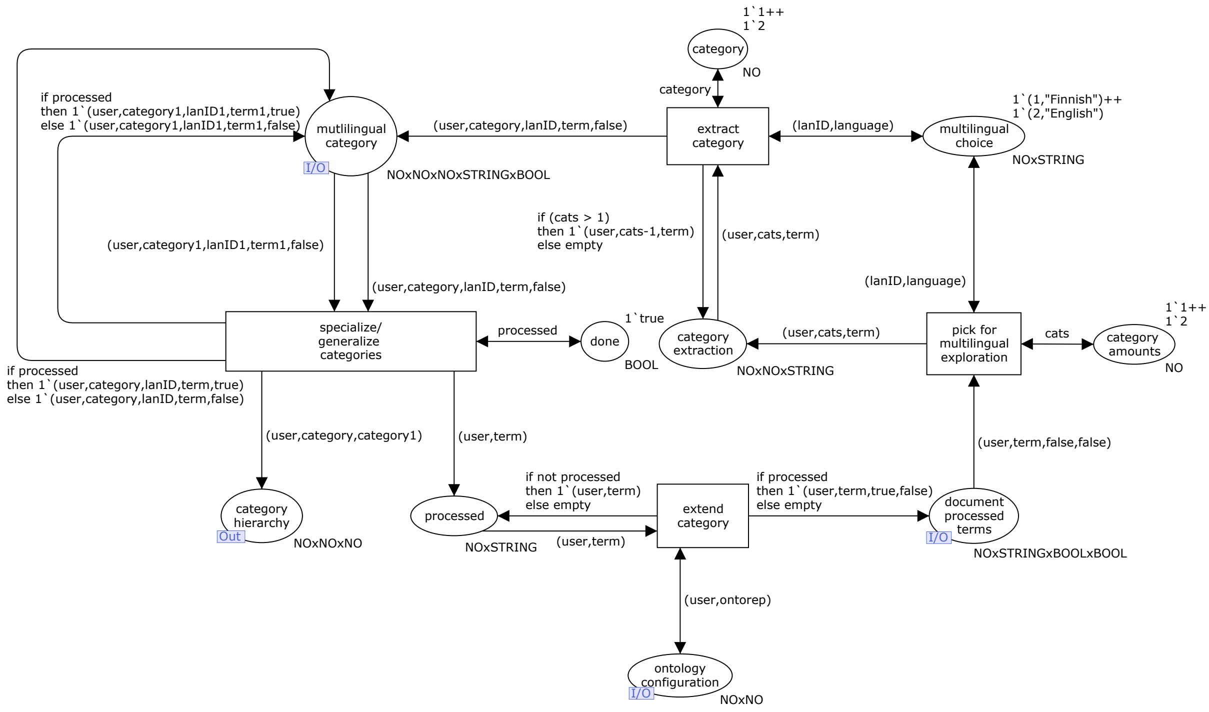

Figure 13: *MultiLangMan*-component.

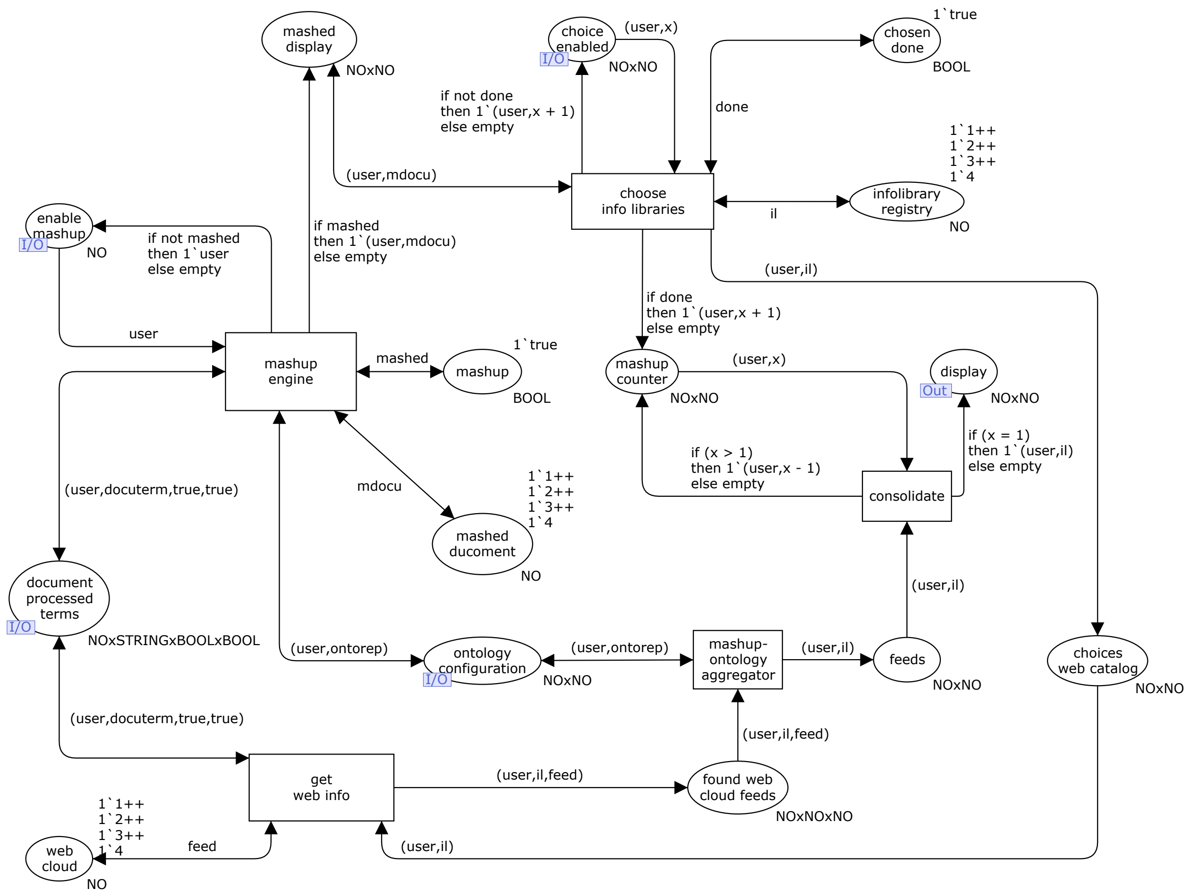

Figure 14: *Mashup*-component.

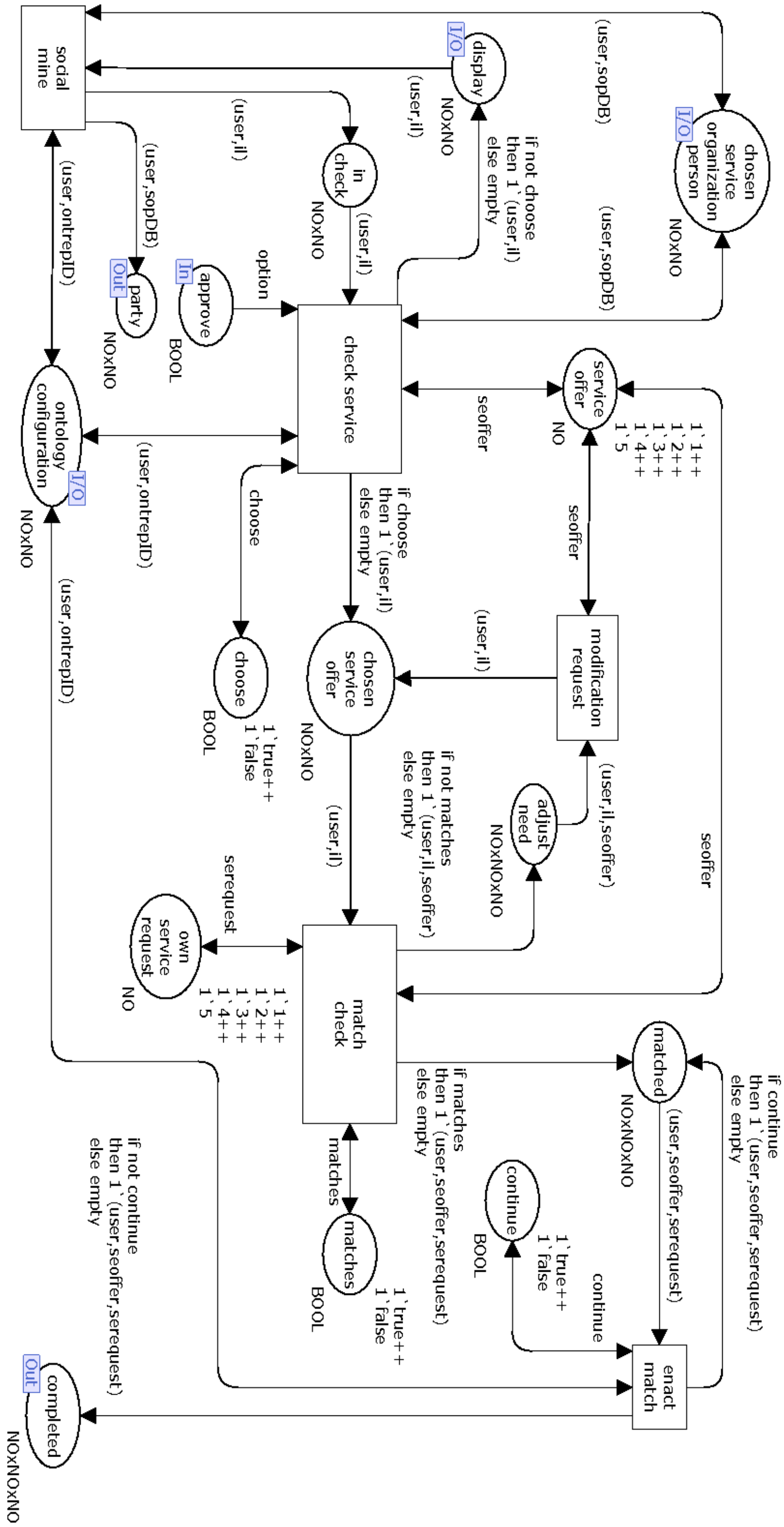

Figure 15: *CarryOut*-component.

Statistics

```
State Space
    Nodes:  3567
    Arcs:   9297
    Secs:   5
    Status: Full

Scc Graph
    Nodes:  3567
    Arcs:   9297
    Secs:   0
```

Boundedness Properties

Best Integer Bounds

```
                                    Upper      Lower
ContentFactoryHUB'all_text_terms 1
                                    1          0
ContentFactoryHUB'amount_docu_terms 1
                                    1          0
ContentFactoryHUB'amount_service_organization_person 1
                                    1          0
ContentFactoryHUB'amount_uncovered_text_terms 1
                                    1          0
ContentFactoryHUB'chosen_service_organization_person 1
                                    2          0
ContentFactoryHUB'configured 1
                                    1          0
ContentFactoryHUB'counter_for_user 1
                                    1          0
ContentFactoryHUB'enable_extraction 1
                                    1          0
ContentFactoryHUB'enable_next 1
                                    1          0
ContentFactoryHUB'enable_term_extraction 1
                                    1          0
ContentFactoryHUB'goal_hierarchy 1
                                    1          0
ContentFactoryHUB'keyword_input 1
                                    1          0
ContentFactoryHUB'pass 1
                                    1          0
ContentFactoryHUB'pick_server 1
                                    11         11
ContentFactoryHUB'service_organization_person_DB 1
                                    2          2
ContentFactoryHUB'service_organization_person_choice_log 1
                                    1          0
ContentFactoryHUB'service_organization_person_choice_log_input 1
                                    1          0
ContentFactoryHUB'service_organization_person_search_results 1
                                    1          0
ContentFactoryHUB'term_log 1
                                    4          0
ContentFactoryHUB'term_log_input 1
                                    4          0
ContentFactoryHUB'terms_found 1
                                    4          0
HUBTermExtraction'assumed_terms_in_text 1
                                    2          2
HUBTermExtraction'choose_collab_result 1
                                    2          2
```

```
HUBTermExtraction'chosen_goal 1
                              1          0
HUBTermExtraction'contained_term 1
                              1          0
HUBTermExtraction'enable_remove 1
                              1          0
HUBTermExtraction'enable_transfer 1
                              1          0
HUBTermExtraction'keyword_need 1
                              1          1
HUBTermExtraction'service_organization_person_entry 1
                              2          2
HUBTermExtraction'term_pool 1
                              2          2
```

Best Upper Multi-set Bounds

```
    ContentFactoryHUB'all_text_terms 1
1`(1,0)++
1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)
    ContentFactoryHUB'amount_docu_terms 1
1`(1,0)
    ContentFactoryHUB'amount_service_organization_person 1
1`(1,1)++
1`(1,2)
    ContentFactoryHUB'amount_uncovered_text_terms 1
1`(1,0)++
1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)
    ContentFactoryHUB'chosen_service_organization_person 1
2`(1,31)++
2`(1,32)
    ContentFactoryHUB'configured 1
1`(1,false)
    ContentFactoryHUB'counter_for_user 1
1`(1,0)++
1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)
    ContentFactoryHUB'enable_extraction 1
1`(1,false)
    ContentFactoryHUB'enable_next 1
1`1
    ContentFactoryHUB'enable_term_extraction 1
1`(1,false)
    ContentFactoryHUB'goal_hierarchy 1
1`(1,"hierarchy")
    ContentFactoryHUB'keyword_input 1
1`(1,1,"hierarchy")
    ContentFactoryHUB'pass 1
1`(1,0)++
1`(1,1)
    ContentFactoryHUB'pick_server 1
1`false++
10`true
    ContentFactoryHUB'service_organization_person_DB 1
1`1++
```

```
1`2
    ContentFactoryHUB'service_organization_person_choice_log 1
1`(1,1)++
1`(1,2)
    ContentFactoryHUB'service_organization_person_choice_log_input 1
1`(1,1)++
1`(1,2)
    ContentFactoryHUB'service_organization_person_search_results 1
1`(1,1)++
1`(1,2)
    ContentFactoryHUB'term_log 1
2`(1,1,"China")++
2`(1,1,"shoe")++
2`(1,2,"China")++
2`(1,2,"shoe")
    ContentFactoryHUB'term_log_input 1
2`(1,1,"China")++
2`(1,1,"shoe")++
2`(1,2,"China")++
2`(1,2,"shoe")
    ContentFactoryHUB'terms_found 1
2`(1,1,"China")++
2`(1,1,"shoe")++
2`(1,2,"China")++
2`(1,2,"shoe")
     HUBTermExtraction'assumed_terms_in_text 1
1`1++
1`2
     HUBTermExtraction'choose_collab_result 1
1`false++
1`true
     HUBTermExtraction'chosen_goal 1
1`(1,"hierarchy")
     HUBTermExtraction'contained_term 1
1`(1,1,1)++
1`(1,1,2)++
1`(1,2,1)++
1`(1,2,2)
     HUBTermExtraction'enable_remove 1
1`1
     HUBTermExtraction'enable_transfer 1
1`(1,1,1)++
1`(1,1,2)
     HUBTermExtraction'keyword_need 1
1`1
     HUBTermExtraction'service_organization_person_entry 1
1`31++
1`32
     HUBTermExtraction'term_pool 1
1`"China"++
1`"shoe"
```

Best Lower Multi-set Bounds

```
    ContentFactoryHUB'all_text_terms 1
                    empty
    ContentFactoryHUB'amount_docu_terms 1
                    empty
    ContentFactoryHUB'amount_service_organization_person 1
                    empty
    ContentFactoryHUB'amount_uncovered_text_terms 1
                    empty
    ContentFactoryHUB'chosen_service_organization_person 1
```

```
                         empty
     ContentFactoryHUB'configured 1
                         empty
     ContentFactoryHUB'counter_for_user 1
                         empty
     ContentFactoryHUB'enable_extraction 1
                         empty
     ContentFactoryHUB'enable_next 1
                         empty
     ContentFactoryHUB'enable_term_extraction 1
                         empty
     ContentFactoryHUB'goal_hierarchy 1
                         empty
     ContentFactoryHUB'keyword_input 1
                         empty
     ContentFactoryHUB'pass 1
                         empty
     ContentFactoryHUB'pick_server 1
1`false++
10`true
     ContentFactoryHUB'service_organization_person_DB 1
1`1++
1`2
     ContentFactoryHUB'service_organization_person_choice_log 1
                         empty
     ContentFactoryHUB'service_organization_person_choice_log_input 1
                         empty
     ContentFactoryHUB'service_organization_person_search_results 1
                         empty
     ContentFactoryHUB'term_log 1
                         empty
     ContentFactoryHUB'term_log_input 1
                         empty
     ContentFactoryHUB'terms_found 1
                         empty
     HUBTermExtraction'assumed_terms_in_text 1
1`1++
1`2
     HUBTermExtraction'choose_collab_result 1
1`false++
1`true
     HUBTermExtraction'chosen_goal 1
                         empty
     HUBTermExtraction'contained_term 1
                         empty
     HUBTermExtraction'enable_remove 1
                         empty
     HUBTermExtraction'enable_transfer 1
                         empty
     HUBTermExtraction'keyword_need 1
1`1
     HUBTermExtraction'service_organization_person_entry 1
1`31++
1`32
     HUBTermExtraction'term_pool 1
1`"China"++
1`"shoe"
```

Home Properties

Home Markings None

Liveness Properties

Dead Markings 120 [783,778,755,750,731,...]

Dead Transition Instances None
Live Transition Instances None

Fairness Properties

No infinite occurrence sequences.

APPENDIX C: CLOUDTERMEXTRACTION

Statistics

```
State Space
   Nodes:  12116
   Arcs:   31704
   Secs:   49
   Status: Full

Scc Graph
   Nodes:  7860
   Arcs:   25596
   Secs:   1
```

Boundedness Properties

Best Integer Bounds

```
                                   Upper       Lower
   CloudTermExtraction'assumed_docu_terms 1
                                   1           1
   CloudTermExtraction'assumed_term_amount 1
                                   2           2
   CloudTermExtraction'choose_ontology_repository 1
                                   2           2
   CloudTermExtraction'customer_documents 1
                                   2           2
   CloudTermExtraction'document_processed_terms 1
                                   2           0
   CloudTermExtraction'document_terms_found 1
                                   2           0
   CloudTermExtraction'ontology_repository_pool 1
                                   2           2
   ContentFactoryHUB'all_text_terms 1
                                   1           0
   ContentFactoryHUB'amount_docu_terms 1
                                   1           0
   ContentFactoryHUB'amount_uncovered_text_terms 1
                                   1           0
   ContentFactoryHUB'amount_uncovered_user_document_terms 1
                                   1           0
   ContentFactoryHUB'configured 1
                                   1           0
   ContentFactoryHUB'document_amount 1
                                   1           0
   ContentFactoryHUB'enable_extraction 1
                                   1           0
   ContentFactoryHUB'enable_term_extraction 1
                                   1           0
   ContentFactoryHUB'not_covered_term 1
                                   4           0
   ContentFactoryHUB'ontology_configuration 1
                                   4           0
   ContentFactoryHUB'preliminary_concept_counter 1
                                   1           0
   ContentFactoryHUB'terms_found 1
                                   2           0
   ContentFactoryHUB'used_documents 1
                                   2           0
   TermVote'agree 1                11          11
   TermVote'all_pulled 1           1           0
   TermVote'counter 1              1           1
```

```
    TermVote'crowd 1          2            0
    TermVote'crowd_size 1     1            1
    TermVote'enable_check 1 1              0
    TermVote'enable_crowd_reset 1
                              1            0
    TermVote'enable_pick 1    1            0
    TermVote'member_pool 1    2            0
    TermVote'picked_term 1    1            0
    TermVote'pull_enable 1    1            0
```

Best Upper Multi-set Bounds

```
    CloudTermExtraction'assumed_docu_terms 1
                            1`1++
1`2++
1`8
    CloudTermExtraction'assumed_term_amount 1
                            1`"are"++
1`"here"
    CloudTermExtraction'choose_ontology_repository 1
                            1`false++
1`true
    CloudTermExtraction'customer_documents 1
                            1`1++
1`2
    CloudTermExtraction'document_processed_terms 1
                            2`(1,"are",false,false)++
2`(1,"here",false,false)
    CloudTermExtraction'document_terms_found 1
                            2`(1,1,"are",false)++
2`(1,1,"are",true)++
2`(1,1,"here",false)++
2`(1,1,"here",true)++
2`(1,2,"are",false)++
2`(1,2,"are",true)++
2`(1,2,"here",false)++
2`(1,2,"here",true)
    CloudTermExtraction'ontology_repository_pool 1
                            1`1++
1`2
    ContentFactoryHUB'all_text_terms 1
                            1`(1,1)++
1`(1,2)
    ContentFactoryHUB'amount_docu_terms 1
                            1`(1,0)++
1`(1,1)++
1`(1,2)
    ContentFactoryHUB'amount_uncovered_text_terms 1
                            1`(1,0)++
1`(1,1)++
1`(1,2)++
1`(1,3)
    ContentFactoryHUB'amount_uncovered_user_document_terms 1
                            1`(1,0)++
1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)
    ContentFactoryHUB'configured 1
                            1`(1,false)++
1`(1,true)
    ContentFactoryHUB'document_amount 1
                            1`(1,0)++
```

```
1`(1,1)++
1`(1,2)
      ContentFactoryHUB'enable_extraction 1
                              1`(1,false)++
1`(1,true)
      ContentFactoryHUB'enable_term_extraction 1
                              1`(1,false)++
1`(1,true)
      ContentFactoryHUB'not_covered_term 1
                              1`(1,"China",false)++
2`(1,"are",false)++
2`(1,"here",false)++
1`(1,"shoe",false)
      ContentFactoryHUB'ontology_configuration 1
                              4`(1,1)++
4`(1,2)
      ContentFactoryHUB'preliminary_concept_counter 1
                              1`(1,0)
      ContentFactoryHUB'terms_found 1
                              1`(1,2,"China")++
1`(1,2,"shoe")
      ContentFactoryHUB'used_documents 1
                              2`(1,1)++
2`(1,2)
      TermVote'agree 1        1`false++
10`true
      TermVote'all_pulled 1
                              1`1
      TermVote'counter 1      1`0++
1`1
      TermVote'crowd 1        1`(1,false)++
1`(1,true)++
1`(2,false)++
1`(2,true)
      TermVote'crowd_size 1
                              1`1++
1`2
      TermVote'enable_check 1
                              1`1
      TermVote'enable_crowd_reset 1
                              1`1
      TermVote'enable_pick 1
                              1`1
      TermVote'member_pool 1
                              1`1++
1`2
      TermVote'picked_term 1
                              1`(1,1,0,"are",false)++
1`(1,1,0,"are",true)++
1`(1,1,0,"here",false)++
1`(1,1,0,"here",true)++
1`(1,1,1,"are",false)++
1`(1,1,1,"are",true)++
1`(1,1,1,"here",false)++
1`(1,1,1,"here",true)++
1`(1,1,2,"are",true)++
1`(1,1,2,"here",true)++
1`(1,2,0,"are",false)++
1`(1,2,0,"are",true)++
1`(1,2,0,"here",false)++
1`(1,2,0,"here",true)++
1`(1,2,1,"are",false)++
```

```
1‘(1,2,1,"are",true)++
1‘(1,2,1,"here",false)++
1‘(1,2,1,"here",true)++
1‘(1,2,2,"are",true)++
1‘(1,2,2,"here",true)
          TermVote'pull_enable 1
                                                 1‘1
```

Best Lower Multi-set Bounds

```
          CloudTermExtraction'assumed_docu_terms 1
                                                 empty
          CloudTermExtraction'assumed_term_amount 1
                                                 1‘"are"++
1‘"here"
          CloudTermExtraction'choose_ontology_repository 1
                                                 1‘false++
1‘true
          CloudTermExtraction'customer_documents 1
                                                 1‘1++
1‘2
          CloudTermExtraction'document_processed_terms 1
                                                 empty
          CloudTermExtraction'document_terms_found 1
                                                 empty
          CloudTermExtraction'ontology_repository_pool 1
                                                 1‘1++
1‘2
          ContentFactoryHUB'all_text_terms 1
                                                 empty
          ContentFactoryHUB'amount_docu_terms 1
                                                 empty
          ContentFactoryHUB'amount_uncovered_text_terms 1
                                                 empty
          ContentFactoryHUB'amount_uncovered_user_document_terms 1
                                                 empty
          ContentFactoryHUB'configured 1
                                                 empty
          ContentFactoryHUB'document_amount 1
                                                 empty
          ContentFactoryHUB'enable_extraction 1
                                                 empty
          ContentFactoryHUB'enable_term_extraction 1
                                                 empty
          ContentFactoryHUB'not_covered_term 1
                                                 empty
          ContentFactoryHUB'ontology_configuration 1
                                                 empty
          ContentFactoryHUB'preliminary_concept_counter 1
                                                 empty
          ContentFactoryHUB'terms_found 1
                                                 empty
          ContentFactoryHUB'used_documents 1
                                                 empty
          TermVote'agree 1                       1‘false++
10‘true
          TermVote'all_pulled 1
                                                 empty
          TermVote'counter 1                     empty
          TermVote'crowd 1                       empty
          TermVote'crowd_size 1
                                                 empty
          TermVote'enable_check 1
```

```
                              empty
    TermVote'enable_crowd_reset 1
                              empty
    TermVote'enable_pick 1
                              empty
    TermVote'member_pool 1
                              empty
    TermVote'picked_term 1
                              empty
    TermVote'pull_enable 1
                              empty
```

Home Properties

Home Markings None

Liveness Properties

Dead Markings 360 [12116,12115,12114,12113,12112,...]
Dead Transition Instances None
Live Transition Instances None

Fairness Properties

CloudTermExtraction′collect_terms_from_customer 1 No Fairness
CloudTermExtraction′configure_ontology_repository 1 No Fairness
CloudTermExtraction′divert 1 Fair
CloudTermExtraction′ontology_configuration_from_documents 1 No Fairness
CloudTermExtraction′reset_extraction 1 Fair
TermVote′check_outcome 1 Fair
TermVote′pick_term 1 Fair
TermVote′pull_crowd 1 No Fairness
TermVote′reset_crowd 1 Fair
TermVote′vote 1 Fair

APPENDIX C1: TERMVOTE

Statistics

```
State Space
    Nodes:   8843
    Arcs:    23332
    Secs:    38
    Status:  Full

Scc Graph
    Nodes:   4573
    Arcs:    12202
    Secs:    1
```

Boundedness Properties

Best Integer Bounds

```
                              Upper      Lower
    TermVote'agree 1           11         11
    TermVote'all_pulled 1      1          0
    TermVote'counter 1         1          1
    TermVote'crowd 1           4          0
    TermVote'crowd_size 1      1          1
    TermVote'document_terms_found 1
                               1          0
    TermVote'enable_check 1    1          0
    TermVote'enable_crowd_reset 1
```

```
                                      1              0
        TermVote'enable_pick  1   1              0
        TermVote'member_pool  1   8              4
        TermVote'picked_term  1   1              0
        TermVote'pull_enable  1   1              0
```

Best Upper Multi-set Bounds

```
     TermVote'agree  1
1`false++
10`true
     TermVote'all_pulled  1
1`1
     TermVote'counter  1
1`0++
1`1++
1`2++
1`3
     TermVote'crowd  1
1`(1,false)++
1`(1,true)++
1`(2,false)++
1`(2,true)++
1`(3,false)++
1`(3,true)++
1`(4,false)++
1`(4,true)++
1`(5,false)++
1`(5,true)++
1`(6,false)++
1`(6,true)++
1`(7,false)++
1`(7,true)++
1`(8,false)++
1`(8,true)
     TermVote'crowd_size  1
1`1++
1`2++
1`3++
1`4
     TermVote'document_terms_found  1
1`(1,1,"here",false)++
1`(1,1,"here",true)
     TermVote'enable_check  1
1`1
     TermVote'enable_crowd_reset  1
1`1
     TermVote'enable_pick  1
1`1
     TermVote'member_pool  1
1`1++
1`2++
1`3++
1`4++
1`5++
1`6++
1`7++
1`8
     TermVote'picked_term  1
1`(1,1,0,"here",false)++
1`(1,1,0,"here",true)++
1`(1,1,1,"here",false)++
1`(1,1,1,"here",true)++
```

```
1‘(1,1,2,"here",false)++
1‘(1,1,2,"here",true)++
1‘(1,1,3,"here",false)++
1‘(1,1,3,"here",true)++
1‘(1,1,4,"here",true)
    TermVote'pull_enable 1
1‘1
```

Best Lower Multi-set Bounds

```
    TermVote'agree 1
1‘false++
10‘true
  TermVote'all_pulled 1
                          empty
  TermVote'counter 1   empty
  TermVote'crowd 1     empty
  TermVote'crowd_size 1
                          empty
  TermVote'document_terms_found 1
                          empty
  TermVote'enable_check 1
                          empty
  TermVote'enable_crowd_reset 1
                          empty
  TermVote'enable_pick 1
                          empty
  TermVote'member_pool 1
                          empty
  TermVote'picked_term 1
                          empty
  TermVote'pull_enable 1
                          empty
```

Home Properties

Home Markings None

Liveness Properties

Dead Markings 70 [8843,8842,8841,8840,8839,...]
Dead Transition Instances None
Live Transition Instances None

Fairness Properties

TermVote′check_outcome 1 Impartial
TermVote′pick_term 1 Impartial
TermVote′pull_crowd 1 Fair
TermVote′reset_crowd 1 Impartial
TermVote′vote 1 Impartial

APPENDIX D1: CONCEPTDISCOVERY

Statistics

```
State Space
    Nodes:  17354
    Arcs:   130010
    Secs:   472
    Status: Full

Scc Graph
    Nodes:  17354
    Arcs:   130010
    Secs:   3
```

Boundedness Properties

Best Integer Bounds

```
                                    Upper       Lower
    ConceptDiscovery'amount_service_organization_person 1
                                    1           0
    ConceptDiscovery'amount_uncovered_user_document_terms 1
                                    1           0
    ConceptDiscovery'class_counter 1
                                    1           0
    ConceptDiscovery'concept_option 1
                                    2           2
    ConceptDiscovery'document_amount 1
                                    1           0
    ConceptDiscovery'document_processed_terms 1
                                    9           0
    ConceptDiscovery'document_service_organization_person 1
                                    2           2
    ConceptDiscovery'enable_build 1
                                    1           0
    ConceptDiscovery'enable_docu_concept 1
                                    1           0
    ConceptDiscovery'not_covered_term 1
                                    9           0
    ConceptDiscovery'online_text_counter 1
                                    1           0
    ConceptDiscovery'preliminary_concept 1
                                    21          0
    ConceptDiscovery'preliminary_concept_counter 1
                                    1           0
    ConceptDiscovery'used_documents 1
                                    12          0
```

Best Upper Multi-set Bounds

```
    ConceptDiscovery'amount_service_organization_person 1
                                1`(1,4)
    ConceptDiscovery'amount_uncovered_user_document_terms 1
                                1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)++
1`(1,5)++
1`(1,6)++
1`(1,7)++
1`(1,8)++
1`(1,9)
    ConceptDiscovery'class_counter 1
```

```
                              1`(1,0)
      ConceptDiscovery'concept_option 1
                              1`1++
1`2
      ConceptDiscovery'document_amount 1
                              1`(1,8)++
1`(1,9)++
1`(1,10)++
1`(1,11)++
1`(1,12)
      ConceptDiscovery'document_processed_terms 1
                              3`(1,"checks",false,false)++
1`(1,"grid",false,false)++
1`(1,"just",false,false)++
1`(1,"some",false,false)++
3`(1,"terms",false,false)
      ConceptDiscovery'document_service_organization_person 1
                              1`99++
1`100
      ConceptDiscovery'enable_build 1
                              1`(1,21)
      ConceptDiscovery'enable_docu_concept 1
                              1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)++
1`(1,5)++
1`(1,6)++
1`(1,7)++
1`(1,8)++
1`(1,9)++
1`(1,10)++
1`(1,11)++
1`(1,12)
      ConceptDiscovery'not_covered_term 1
                              3`(1,"checks",false)++
1`(1,"grid",false)++
1`(1,"just",false)++
1`(1,"some",false)++
3`(1,"terms",false)
      ConceptDiscovery'online_text_counter 1
                              1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)
      ConceptDiscovery'preliminary_concept 1
                              21`(1,1,false,false)++
21`(1,2,false,false)
      ConceptDiscovery'preliminary_concept_counter 1
                              1`(1,0)++
1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)++
1`(1,5)++
1`(1,6)++
1`(1,7)++
1`(1,8)++
1`(1,9)++
1`(1,10)++
1`(1,11)++
1`(1,12)++
```

```
1`(1,13)++
1`(1,14)++
1`(1,15)++
1`(1,16)++
1`(1,17)++
1`(1,18)++
1`(1,19)++
1`(1,20)
     ConceptDiscovery'used_documents 1
                         2`(1,1)++
1`(1,2)++
3`(1,4)++
2`(1,5)++
4`(1,99)++
4`(1,100)
```

Best Lower Multi-set Bounds

```
     ConceptDiscovery'amount_service_organization_person 1
                         empty
     ConceptDiscovery'amount_uncovered_user_document_terms 1
                         empty
     ConceptDiscovery'class_counter 1
                         empty
     ConceptDiscovery'concept_option 1
                         1`1++
1`2
     ConceptDiscovery'document_amount 1
                         empty
     ConceptDiscovery'document_processed_terms 1
                         empty
     ConceptDiscovery'document_service_organization_person 1
                         1`99++
1`100
     ConceptDiscovery'enable_build 1
                         empty
     ConceptDiscovery'enable_docu_concept 1
                         empty
     ConceptDiscovery'not_covered_term 1
                         empty
     ConceptDiscovery'online_text_counter 1
                         empty
     ConceptDiscovery'preliminary_concept 1
                         empty
     ConceptDiscovery'preliminary_concept_counter 1
                         empty
     ConceptDiscovery'used_documents 1
                         empty
```

Home Properties

Home Markings None

Liveness Properties

Dead Markings 22 [17354,17353,17352,17351,17350,...]

Dead Transition Instances None

Live Transition Instances None

Fairness Properties

No infinite occurrence sequences.

APPENDIX D2: BUCLRECONCEPT AND HARREVCHAS

Statistics

```
State Space
    Nodes:   25989
    Arcs:    101144
    Secs:    177
    Status:  Full

Scc Graph
    Nodes:   25989
    Arcs:    85880
    Secs:    1
```

Boundedness Properties

Best Integer Bounds

```
                                        Upper       Lower
    BuClReConcept'classification 1
                                        2           2
    BuClReConcept'classified_concepts 1
                                        3           0
    BuClReConcept'created_class 1
                                        3           0
    BuClReConcept'enable_classify 1
                                        1           0
    HarRevChAs'approve 1                1           1
    HarRevChAs'approved_concept 1
                                        3           0
    HarRevChAs'enable_assembly 1
                                        1           0
    HarRevChAs'ontology_repository_ID 1
                                        2           0
    HarRevChAs'picked_ontorepID 1
                                        1           0
    OntoRepBuild'approved_counter 1
                                        1           0
    OntoRepBuild'choice_enabled 1
                                        1           0
    OntoRepBuild'class_counter 1
                                        1           0
    OntoRepBuild'concept 1              3           0
    OntoRepBuild'concept_class_context 1
                                        3           0
    OntoRepBuild'concept_class_pattern 1
                                        3           0
    OntoRepBuild'enable_build 1
                                        1           0
    OntoRepBuild'enable_mashup 1
                                        1           0
    OntoRepBuild'harvested_context 1
                                        2           2
    OntoRepBuild'harvested_pattern 1
                                        2           2
    OntoRepBuild'new_ontology_repository 1
                                        1           0
    OntoRepBuild'preliminary_concept 1
                                        3           0
    OntoRepBuild'release_counter 1
                                        1           0
```

Best Upper Multi-set Bounds

```
      BuClReConcept'classification 1
1`1++
1`2
      BuClReConcept'classified_concepts 1
1`(1,1,1)++
1`(1,1,2)++
2`(1,2,1)++
2`(1,2,2)
      BuClReConcept'created_class 1
3`(1,1)++
3`(1,2)
      BuClReConcept'enable_classify 1
                              1`1
      HarRevChAs'approve 1
1`true
      HarRevChAs'approved_concept 1
1`(1,1,99,99)++
1`(1,1,99,100)++
1`(1,1,100,99)++
1`(1,1,100,100)++
2`(1,2,99,99)++
2`(1,2,99,100)++
2`(1,2,100,99)++
2`(1,2,100,100)
      HarRevChAs'enable_assembly 1
1`1
      HarRevChAs'ontology_repository_ID 1
1`99++
1`100
      HarRevChAs'picked_ontorepID 1
1`99++
1`100
      OntoRepBuild'approved_counter 1
1`(1,0,false)++
1`(1,0,true)++
1`(1,1,false)++
1`(1,1,true)++
1`(1,2,false)++
1`(1,2,true)++
1`(1,3,false)++
1`(1,3,true)
      OntoRepBuild'choice_enabled 1
1`(1,0)
      OntoRepBuild'class_counter 1
1`(1,0)++
1`(1,1)++
1`(1,2)++
1`(1,3)
      OntoRepBuild'concept 1
1`(1,1,false,false)++
1`(1,1,true,false)++
1`(1,1,true,true)++
2`(1,2,false,false)++
2`(1,2,true,false)++
2`(1,2,true,true)
      OntoRepBuild'concept_class_context 1
1`(1,1,99)++
1`(1,1,100)++
2`(1,2,99)++
2`(1,2,100)
      OntoRepBuild'concept_class_pattern 1
1`(1,1,99)++
```

```
1`(1,1,100)++
2`(1,2,99)++
2`(1,2,100)
     OntoRepBuild'enable_build 1
1`(1,1)++
1`(1,2)++
1`(1,3)
     OntoRepBuild'enable_mashup 1
                          1`1
     OntoRepBuild'harvested_context 1
1`99++
1`100
     OntoRepBuild'harvested_pattern 1
1`99++
1`100
     OntoRepBuild'new_ontology_repository 1
1`(1,99)++
1`(1,100)
     OntoRepBuild'preliminary_concept 1
1`(1,1,false,false)++
1`(1,1,true,false)++
1`(1,1,true,true)++
2`(1,2,false,false)++
2`(1,2,true,false)++
2`(1,2,true,true)
     OntoRepBuild'release_counter 1
1`(1,0,false)++
1`(1,0,true)++
1`(1,1,false)++
1`(1,1,true)++
1`(1,2,false)++
1`(1,2,true)++
1`(1,3,false)++
1`(1,3,true)
```

Best Lower Multi-set Bounds

```
     BuClReConcept'classification 1
1`1++
1`2
     BuClReConcept'classified_concepts 1
                          empty
     BuClReConcept'created_class 1
                          empty
     BuClReConcept'enable_classify 1
                          empty
     HarRevChAs'approve 1
                          1`true
     HarRevChAs'approved_concept 1
                          empty
     HarRevChAs'enable_assembly 1
                          empty
     HarRevChAs'ontology_repository_ID 1
                          empty
     HarRevChAs'picked_ontorepID 1
                          empty
     OntoRepBuild'approved_counter 1
                          empty
     OntoRepBuild'choice_enabled 1
                          empty
     OntoRepBuild'class_counter 1
                          empty
     OntoRepBuild'concept 1
```

```
                                  empty
        OntoRepBuild ' concept_class_context 1
                                  empty
        OntoRepBuild ' concept_class_pattern 1
                                  empty
        OntoRepBuild ' enable_build 1
                                  empty
        OntoRepBuild ' enable_mashup 1
                                  empty
        OntoRepBuild ' harvested_context 1
1‘99++
1‘100
        OntoRepBuild ' harvested_pattern 1
1‘99++
1‘100
        OntoRepBuild ' new_ontology_repository 1
                                  empty
        OntoRepBuild ' preliminary_concept 1
                                  empty
        OntoRepBuild ' release_counter 1
                                  empty
```

Home Properties

Home Markings None

Liveness Properties

Dead Markings 904 [983,982,976,975,971,...]
Dead Transition Instances None
Live Transition Instances None

Fairness Properties

Impartial Transition Instances
HarRevChAs′check_approved 1
Fair Transition Instances
BuClReConcept′build$_c$*oncept_classes*1
*BuClReConcept'classify_concepts*1
*HarRevChAs'assemble_ontology_repository*1
Just Transition Instances None
Transition Instances with No Fairness
BuClReConcept′release_concepts 1
HarRevChAs′context_harvesting 1
HarRevChAs′human_review 1
HarRevChAs′pattern_harvest 1

APPENDIX D3: MULTILANGMAN

Statistics

```
   State Space
        Nodes:  49
        Arcs:   100
        Secs:   0
        Status: Full

   Scc Graph
        Nodes:  49
        Arcs:   72
        Secs:   0
```

Boundedness Properties

Best Integer Bounds

```
                              Upper       Lower
     MultiLangMan'category 1  2           2
     MultiLangMan'category_amounts 1
                              2           2
     MultiLangMan'category_extraction 1
                              1           0
     MultiLangMan'category_hierarchy 1
                              1           0
     MultiLangMan'document_processed_terms 1
                              1           0
     MultiLangMan'done 1      1           1
     MultiLangMan'multilingual_choice 1
                              2           2
     MultiLangMan'mutlilingual_category 1
                              2           0
     MultiLangMan'ontology_configuration 1
                              2           2
     MultiLangMan'processed 1
                              1           0
```

Best Upper Multi-set Bounds

```
     MultiLangMan'category 1
1`1++
1`2
     MultiLangMan'category_amounts 1
1`1++
1`2
     MultiLangMan'category_extraction 1
1`(1,1,"are")++
1`(1,2,"are")
     MultiLangMan'category_hierarchy 1
1`(1,1,1)++
1`(1,1,2)++
1`(1,2,1)++
1`(1,2,2)
     MultiLangMan'document_processed_terms 1
1`(1,"are",false,false)++
1`(1,"are",true,false)
     MultiLangMan'done 1 1`true
     MultiLangMan'multilingual_choice 1
1`(1,"Finnish")++
1`(2,"English")
     MultiLangMan'mutlilingual_category 1
2`(1,1,1,"are",false)++
2`(1,1,1,"are",true)++
2`(1,1,2,"are",false)++
2`(1,1,2,"are",true)++
2`(1,2,1,"are",false)++
2`(1,2,1,"are",true)++
2`(1,2,2,"are",false)++
2`(1,2,2,"are",true)
     MultiLangMan'ontology_configuration 1
1`(1,1)++
1`(1,2)
     MultiLangMan'processed 1
1`(1,"are")
```

Best Lower Multi-set Bounds

```
     MultiLangMan'category 1
1`1++
```

```
1`2
      MultiLangMan'category_amounts 1
1`1++
1`2
      MultiLangMan'category_extraction 1
                          empty
      MultiLangMan'category_hierarchy 1
                          empty
      MultiLangMan'document_processed_terms 1
                          empty
      MultiLangMan'done 1 1`true
      MultiLangMan'multilingual_choice 1
1`(1,"Finnish")++
1`(2,"English")
      MultiLangMan'mutlilingual_category 1
                          empty
      MultiLangMan'ontology_configuration 1
1`(1,1)++
1`(1,2)
      MultiLangMan'processed 1
                          empty
```

Home Properties

Home Markings None

Liveness Properties

Dead Markings 18 [7,6,5,49,48,...]
Dead Transition Instances None
Live Transition Instances None

Fairness Properties

MultiLangMan′extend_category 1 Impartial
MultiLangMan′extract_category 1 Fair
MultiLangMan′pick_for_multilingual_exploration 1 Fair
MultiLangMan′specialize 1 Fair

APPENDIX E: MASHUP

Statistics

```
State Space
    Nodes:  117
    Arcs:   376
    Secs:   0
    Status: Full

Scc Graph
    Nodes:  117
    Arcs:   376
    Secs:   0
```

Boundedness Properties

Best Integer Bounds

```
                                  Upper        Lower
    Mashup'choice_enabled 1       1            0
    Mashup'choices_web_catalog 1
                                  1            0
    Mashup'chosen_done 1          1            1
    Mashup'display 1              1            0
    Mashup'document_processed_terms 1
                                  2            2
    Mashup'enable_mashup 1        1            0
    Mashup'feeds 1                1            0
    Mashup'found_web_cloud_feeds 1
                                  1            0
    Mashup'infolibrary_registry 1
                                  4            4
    Mashup'mashed_display 1       1            0
    Mashup'mashed_ducoment 1
                                  4            4
    Mashup'mashup 1               1            1
    Mashup'mashup_counter 1       1            0
    Mashup'ontology_configuration 1
                                  3            3
    Mashup'web_cloud 1            4            4
```

Best Upper Multi-set Bounds

```
    Mashup'choice_enabled 1
1`(1,0)
    Mashup'choices_web_catalog 1
1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)
    Mashup'chosen_done 1
1`true
    Mashup'display 1
1`(1,1)++
1`(1,2)++
1`(1,3)++
1`(1,4)
    Mashup'document_processed_terms 1
1`(1,"for",true,true)++
1`(1,"some",true,true)
    Mashup'enable_mashup 1
1`1
    Mashup'feeds 1
```

```
1‘(1,1)++
1‘(1,2)++
1‘(1,3)++
1‘(1,4)
      Mashup'found_web_cloud_feeds 1
1‘(1,1,1)++
1‘(1,1,2)++
1‘(1,1,3)++
1‘(1,1,4)++
1‘(1,2,1)++
1‘(1,2,2)++
1‘(1,2,3)++
1‘(1,2,4)++
1‘(1,3,1)++
1‘(1,3,2)++
1‘(1,3,3)++
1‘(1,3,4)++
1‘(1,4,1)++
1‘(1,4,2)++
1‘(1,4,3)++
1‘(1,4,4)
      Mashup'infolibrary_registry 1
1‘1++
1‘2++
1‘3++
1‘4
      Mashup'mashed_display 1
1‘(1,1)++
1‘(1,2)++
1‘(1,3)++
1‘(1,4)
      Mashup'mashed_ducoment 1
1‘1++
1‘2++
1‘3++
1‘4
      Mashup'mashup 1
1‘true
      Mashup'mashup_counter 1
1‘(1,1)
      Mashup'ontology_configuration 1
1‘(1,1)++
1‘(1,2)++
1‘(1,8)
      Mashup'web_cloud 1
1‘1++
1‘2++
1‘3++
1‘4
```

Best Lower Multi-set Bounds

```
      Mashup'choice_enabled 1
                          empty
      Mashup'choices_web_catalog 1
                          empty
      Mashup'chosen_done 1
1‘true
      Mashup'display 1    empty
      Mashup'document_processed_terms 1
1‘(1,"for",true,true)++
1‘(1,"some",true,true)
      Mashup'enable_mashup 1
```

```
                            empty
       Mashup'feeds 1       empty
       Mashup'found_web_cloud_feeds 1
                            empty
       Mashup'infolibrary_registry 1
1`1++
1`2++
1`3++
1`4
       Mashup'mashed_display 1
                            empty
       Mashup'mashed_ducoment 1
1`1++
1`2++
1`3++
1`4
       Mashup'mashup 1      1`true
       Mashup'mashup_counter 1
                            empty
       Mashup'ontology_configuration 1
1`(1,1)++
1`(1,2)++
1`(1,8)
       Mashup'web_cloud 1
1`1++
1`2++
1`3++
1`4
```

Home Properties

Home Markings None

Liveness Properties

Dead Markings 16 [117,116,115,114,113,...]
Dead Transition Instances None
Live Transition Instances None

Fairness Properties

No infinite occurrence sequences.

APPENDIX F: CARRYOUT

Statistics

```
State Space
    Nodes:  9076
    Arcs:   134830
    Secs:   19
    Status: Full

Scc Graph
    Nodes:  7126
    Arcs:   91900
    Secs:   1
```

Boundedness Properties

Best Integer Bounds

```
                                 Upper      Lower
    CarryOut'adjust_need 1      2          0
    CarryOut'approve 1          6          0
    CarryOut'choose 1           2          2
    CarryOut'chosen_service_offer 1
                                 2          0
    CarryOut'chosen_service_organization_person 1
                                 2          2
    CarryOut'completed 1        2          0
    CarryOut'continue 1         2          2
    CarryOut'display 1          2          0
    CarryOut'in_check 1         2          0
    CarryOut'matched 1          2          0
    CarryOut'matches 1          2          2
    CarryOut'ontology_configuration 1
                                 5          5
    CarryOut'own_service_request 1
                                 5          5
    CarryOut'party 1            7          0
    CarryOut'service_offer 1
                                 2          2
```

Best Upper Multi-set Bounds

```
    CarryOut'adjust_need 1
1`(1,2,1)++
1`(1,2,2)++
1`(1,4,1)++
1`(1,4,2)
    CarryOut'approve 1
6`true
    CarryOut'choose 1
1`false++
1`true
    CarryOut'chosen_service_offer 1
1`(1,2)++
1`(1,4)
    CarryOut'chosen_service_organization_person 1
1`(1,34)++
1`(1,36)
    CarryOut'completed 1
2`(1,1,1)++
2`(1,1,2)++
2`(1,1,3)++
2`(1,1,4)++
```

```
2`(1,1,5)++
2`(1,2,1)++
2`(1,2,2)++
2`(1,2,3)++
2`(1,2,4)++
2`(1,2,5)
     CarryOut'continue 1
1`false++
1`true
     CarryOut'display 1
1`(1,2)++
1`(1,4)
     CarryOut'in_check
1  1`(1,2)++
1`(1,4)
     CarryOut'matched 1
2`(1,1,1)++
2`(1,1,2)++
2`(1,1,3)++
2`(1,1,4)++
2`(1,1,5)++
2`(1,2,1)++
2`(1,2,2)++
2`(1,2,3)++
2`(1,2,4)++
2`(1,2,5)
     CarryOut'matches 1
1`false++
1`true
     CarryOut'ontology_configuration 1
1`(1,2)++
1`(1,4)++
1`(1,6)++
1`(1,9)++
1`(1,97)
     CarryOut'own_service_request 1
1`1++
1`2++
1`3++
1`4++
1`5
     CarryOut'party 1
7`(1,34)++
7`(1,36)
     CarryOut'service_offer 1
1`1++
1`2
```

Best Lower Multi-set Bounds

```
     CarryOut'adjust_need 1
                         empty
     CarryOut'approve 1  empty
     CarryOut'choose 1   1`false++
1`true
     CarryOut'chosen_service_offer 1
                         empty
     CarryOut'chosen_service_organization_person 1
                         1`(1,34)++
1`(1,36)
     CarryOut'completed 1
                         empty
     CarryOut'continue 1 1`false++
```

```
1`true
      CarryOut'display 1   empty
      CarryOut'in_check 1  empty
      CarryOut'matched 1   empty
      CarryOut'matches 1   1`false++
1`true
      CarryOut'ontology_configuration 1
                                1`(1,2)++
1`(1,4)++
1`(1,6)++
1`(1,9)++
1`(1,97)
      CarryOut'own_service_request 1
                                1`1++
1`2++
1`3++
1`4++
1`5
      CarryOut'party 1      empty
      CarryOut'service_offer 1
                                1`1++
1`2
```

Home Properties

Home Markings None

Liveness Properties

Dead Markings 1248 [9076,9075,9074,9073,9072,...]
Dead Transition Instances None
Live Transition Instances None

Fairness Properties

Impartial Transition Instances None
Fair Transition Instances None
Just Transition Instances None
Transition Instances with No Fairness
CarryOut′check_service 1
CarryOut′enact_match 1
CarryOut′match_check 1
CarryOut′modification_request 1
CarryOut′social_mine 1

REFERENCES

Aalst, W. (2002). Inheritance of Interorganizational Workflows: How to Agree to Disagree Without Loosing Control? *Information Technology and Management Journal*, 2(3).

Allen, P., Higgins, S., McRae, P., and Schlamann, H., editors (2006). *Service Orientation: Winning Strategies and Best Practices*. CAMBRIDGE UNIVERSITY PRESS.

Alonso, G., Fiedler, S., Hagen, C., Lazcano, A., H.Schuldt, and Weiler, N. (1999). WISE: business to business e-commerce. In *Proc. of the 9th International Workshop on Research Issues on Data Engineering*, pages 132–139, Sydney, Australia.

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., and Xu, M. (2007). *Web Services Agreement Specification (WS-Agreement)*. http://www.ogf.org/documents/GFD.107.pdf.

Bengtsson, P. (2002). *Architecture-Level Modifiability Analysis*. PhD thesis, Department of Software Engineering and Computer Science, Blekinge Institute of Technology, Sweden.

Bonchi, F., Brogi, A., Corfini, S., and Gadducci, F. (2007). A behavioural congruence for web services. In Arbab, F. and Sirjani, M., editors, *Proc. Int. Symposium on Fundamentals of Software Engineering (FSEN 2007)*, volume 4767 of *Lecture Notes in Computer Science*, pages 240–256. Springer.

E.A.Marks and Bell, M., editors (2006). *Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Eshuis, R. and Norta, A. (2009). Constructing process views for service outsourcing. In S.Y. Shin, S. O., editor, *Symposium Proc. of the 2009 ACM Symposium on Applied Computing (SAC)*, pages 1615–1616. ACM.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Professional Computing Series. Addison Wesley, Reading, MA, USA.

Grefen, P., Eshuis, R., Mehandjiev, N., Kouvas, G., and Weichhart, G. (2009). Internet-based support for process-oriented instant virtual enterprises. *IEEE Internet Computing*, 13(6):65–73.

Hoffner, Y., Ludwig, H., Gülcü, C., and Grefen, P. (2005). Architecture for Cross-Organizational Business Processes. In *Procs. 2^{nd} Int. Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pages 2–11, Milpitas, CA, USA.

Jensen, K. (1992). *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.* EATCS monographs on Theoretical Computer Science. Springer-Verlag, Berlin.

Jensen, K. (1996). *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.* EATCS monographs on Theoretical Computer Science. Springer-Verlag, Berlin.

Jensen, K., Michael, L., Wells, K. L., Jensen, K., and Kristensen, L. M. (2007). Coloured petri nets and cpn tools for modelling and validation of concurrent systems. In *International Journal on Software Tools for Technology Transfer*, page 2007.

Klein, M. and Kazman, R. (1999). Attribute-based architectural styles. *TECHNICAL REPORT CMU/SEI-99-TR-022 ESC-TR-99-022*.

Klien, E., Lutz, M., and Kuhn, W. Ontology-based discovery of geographic information services?an application in disaster management. *Computers, Environment and Urban Systems*, 30(1).

Kutvonen, L., Ruokolainen, T., and Metso, J. (2007). Interoperability middleware for federated business services in web-Pilarcos. *International Journal of Enterprise Information Systems, Special issue on Interoperability of Enterprise Systems and Applications*, 3(1):1–21.

Lazcano, A., Schuldt, H., Alonso, G., and Schek, H. (2001). WISE: Process Based E-Commerce. *IEEE Data Engineering Bulletin*, 24(1).

Loreto, S., Mecklin, T., Opsenica, M., and Rissanen, H. M. (2009). Service broker architecture: location business case and mashups. *Communications Magazine*, 47(4):97–103.

Martens, A. (2003a). On compatibility of web services. *Petri Net Newsletter*, 65:12–20.

Martens, A. (2003b). On usability of web services. In Calero, C., Daz, O., and Piattini, M., editors, *Proc. of 1st Web Services Quality Workshop (WQW 2003)*.

Mehandjiev, N. and Grefen, P., editors (2010). *Dynamic Business Process Formation for Instant Virtual Enterprises*. Springer.

Muñoz Frutos, H. (2009). Towards a semantic service broker for business grid. In *ESWC 2009 Heraklion: Proceedings of the 6th European Semantic Web Conference on The Semantic Web*, pages 939–943, Berlin, Heidelberg. Springer-Verlag.

Müller, C., Resinas, M., and Ruiz-Cortés, A. (2009). Explaining the non-compliance between templates and agreement offers in ws-agreement. In Baresi, L., Chi, C., and Suzuki, J., editors, *Service-Oriented Computing - ICSOC 2009, 7th International Conference*, volume 5900 of *Lecture Notes in Computer Science*, pages 237–252, Stockholm, Sweden. LNCS Springer.

N.Antonopoulos and Gillam, L., editors (2010). *Cloud Computing: Principles, Systems and Application*. Springer.

Norta, A. and Eshuis, R. (2009). Specification and Verification of Harmonized Business-Process Collaborations - Preprint Version. Accepted for publication in the journal of Information Systems Frontier, University of Helsinki, Finland.

Norta, A. and Grefen, P. (2007). Discovering Patterns for Inter-Organizational Business Collaboration. *International Journal of Cooperative Information Systems (IJCIS)*, 16:507 – 544.

Paukkeri, M., Nieminen, I., Pöllä, M., and Honkela, T. (2008). Language-independent approach to keyphrase extraction and evaluation. In Scott, D. and Uszkoreit, H., editors, *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 237–252, Manchester, UK.

Reisig, W. and Rozenberg, G., editors (1998). *Lectures on Petri nets I: Advances in Petri nets*, volume 1491 of *Lecture Notes in Computer Science*. Springer.

Reisig, W., Schmidt, K., and Stahl, C. (2005). Kommunizierende Workflow-Services modellieren und analysieren. *Informatik - Forschung und Entwicklung*, pages 90–101.

Tran, V. X., Tsuji, H., and Masuda, R. (2009). A new QoS ontology and its QoS-based ranking algorithm for Web services. *Simulation Modelling Practice and Theory*, 17(8):1378–1398.

van der Aalst, W., Lohmann, N., Massuthe, P., Stahl, C., and Wolf, K. (2008). From public views to private views - correctness-by-design for services. In Dumas, M. and Heckel, R., editors, *Proc. WS-FM 2007*, volume 4937 of *Lecture Notes in Computer Science*, pages 139–153. Springer.

Yangarber, R. and Steinberger, R. (2009). Automatic epidemiological surveillance from on-line news in medisys and puls. In *Proceedings of IMED-2009: International Meeting on Emerging Diseases and Surveillance*, Vienna, Austria.