Techniques for Process Analysis and Unification

Silvana Castano[1] and Valeria De Antonellis[2]

[1] University of Milano, ITALY
e-mail: castano@dsi.unimi.it
[2] University of Ancona and Politecnico di Milano, ITALY
e-mail: deantone@elet.polimi.it

Abstract. Information system reengineering activities are strongly related to a deep understanding of processes and related data in the organizations. This paper addresses problems related to the development of techniques and tools for process analysis and for the construction of a unified process architecture where semantic correspondences between processes are identified on the basis of their similarity. Unification of similar processes is considered a basic step for information system interoperability.

Index Terms - Process reengineering, Process analysis, Data Similarity, Process unification, Legacy systems, Interoperability.

1 Introduction

Many of the major organizations, both in the private and public environments, are characterized by the presence of heterogeneous information systems. Heterogeneity is generally related to application contexts, evolution strategies and support technologies. In fact, continuous developments and modifications of information systems to meet over the years changing requirements, and the rapid advances in information technology have lead to a proliferation of information system architectures with partially overlapping data and processes.

Actually, the availability of advanced information technology, together with the need of enhancing efficiency and quality of services promote activities for restructuring existing legacy Information Systems and, possibly, integrating information systems in different contexts. In particular, restructuring of information systems that are obsolete from the technological point of view requires methodological approaches apt to guide the transformation with the aim of preserving and reusing knowledge and existing applications [12]. Integration of autonomous, heterogeneous information systems emphasizes interoperability aspects [19,13]. In this framework, a basic starting point to obtain valuable results is represented by a deep understanding and reengineering of processes in legacy systems. Two levels of intervention can be distinguished: i) a conceptual level, where methods and tools are devoted to modeling, analysis and unification of processes and related data [2]; ii) a technological level, where methods and tools are devoted to the transformation and migration of existing software systems [10] and to the interconnection of existing databases with a federation approach [19,14].

In the paper, we focus on the conceptual level and provide techniques for process analysis in view of reengineering activities. For process analysis, we propose techniques with criteria and metrics based on similarity properties. In particular, we propose a methodological framework and accompanying tools to support the process unification activity, devoted to the definition of semantic correspondences between processes in different contexts, to make feasible Information System interoperability.
Process unification exploits the analysis techniques and the availability of a semantic dictionary where inter-process knowledge is properly organized.

The paper is organized as follows. In Section 2, characteristics of process specifications are given on the basis of our experience with the Information Systems Authority for Public Administration (AIPA in Italy), which is responsible for co-ordinating, planning and controlling the Public Administration Information Systems. In Section 3, the techniques we propose for the construction of the semantic dictionary are illustrated. In Section 4, the similarity techniques for process evaluation are discussed. In Section 5, we describe a methodological framework for process unification. In Section 6, we illustrate the tools supporting our approach. In Section 7, the comparison with related work is presented. Finally, in Section 8, concluding remarks are stated.

2 Process specifications

In the paper, we refer to information on work processes and information systems that has been collected by the Italian Information Systems Authority for Public Administration (AIPA in Italy) in the framework of a large project aiming at building an inventory of information systems, and at pursuing a deep revision and redesign of work processes of the Public Administration, to meet growing and changing needs of clients and the general public. A detailed characterization of Public Administration processes is presented in [6]. The following high-level specification is available for each process:

- The name of the process, univocally identifying the process within the Organizational Unit to which it pertains. The name is composed of an *identifier* *⟨OU_ID, P_ID⟩*, and of a *title*. In the identifier, *OU_ID* is the identifier of the Organizational Unit to which the process pertains and *P_ID* is the process identifier. The process title is a string describing the name of the process.
- An Entity-Relationship (ER) conceptual schema, describing the characteristics of the data manipulated by the process, in form of entities and relationships among entities.
- A textual information, which provides a general description of the process and its functionality. It is a natural language description of the input/output entities (among the ones specified in the associated ER schema), the starting/ending events, the type of the process (e.g., management, administration), and possible supporting application program(s).

Process specifications have been defined by AIPA, on the basis of the data collected through a set of properly defined questionnaires released to users of various Organizational Units of the Public Administration. The questionnaires were designed in order to acquire as much information as possible on process functionalities and on data circulating among them. Process specifications are currently available in electronic format for 2203 work processes selected by AIPA for analysis and reengineering interventions. These interventions aim at achieving the following main goals:

- Reconstruction of information flows between work processes of different Divisions or Ministries, based on the similarity of the data used by processes.
- Reconstruction of macroprocesses, that is, complex activities with a well defined objective achieved by means of the coordinated execution of the constituent processes. Reconstruction of macroprocesses is based on the information available on process functionality and data interactions among different processes.
- Reengineering of work processes and macroprocesses, by optimizing their execution to improve the quality of the offered services.

These activities can result very complex, due to the fact that thousands of processes must be analyzed, spread among heterogeneous information systems [2]. In addition, processes were not always documented, and the personnel with the knowledge required to understand processes and how they work is no longer available, causing data standardization and process reengineering to become crucial activities to be performed with available specifications.

In order to perform process analysis in a semi-automatic way on this large set of specifications, we provide techniques based on the information contained in the specifications, limiting as far as possible the human intervention. In particular, we provide techniques for classifying processes by means of descriptors (Section 2.1), and similarity techniques to compare different processes on the basis of their descriptors, to support the reconstruction of information flows and macroprocess (Section 4).
Examples of applying the proposed techniques on a sample of 149 process specifications related to the Labour Ministry are illustrated, which allows us to evaluate similarity and unification for processes of different Divisions and Offices of the same Ministry. The analysis is being extended on a second sample of 240 specifications related to process of different Ministries of the Public Administration, to evaluate process similarity and unification across different Ministries.

2.1 Process descriptors

In this section, we present the technique we propose for classifying processes by means of *descriptors*, defined starting from the information contained in the process specifications. A descriptor provides a high-level formal description of the characteristics of a process. A process descriptor is a 6-tuple of *features*:

$$\langle \textit{F-Area}, \textit{I-Object}, \textit{O-Object}, \textit{Operation}, \textit{Const-Object}, \textit{Circ-Object} \rangle$$

obtained by extending the technique presented in [8] for disciplined description of process functionality, with capabilities to describe process characteristics related to input/output data. In general, the number and the type of features to be included in a descriptor depends on the type of analysis to be performed on the processes. In our case, to support similarity-based analysis, we represent the following process characteristics:

1. The functional area to which the process pertains within the organization (feature *F-Area*), in order to support cross-functional process unification.
2. The data manipulated by the process, in order to evaluate data similarity between different processes. We distinguish between input entities (feature *I-Object*), representing the entities required in input by the process to start its execution, and output entities (feature *O-Object*), which are the entities produced as the result of the process execution. One or more entities of the ER schema associated with the process can be specified in these features.
3. The functionality of the process, in order to evaluate the similarity of performed operations. Process functionality is described by means of the performed operation (feature *Operation*), the mandatory constitutive entities required by the operation (feature *Cons-Object*) and the optional circumstantial entities involved in the operation (feature *Circ-Object*).

For example, let us consider the process `Labour Inspectorate Employee Retirement`, which establishes the retirable employees of the Labour Inspectorate by evaluating the contributions they paid. This process is described by means of the descriptor shown in Fig. 1.

Labour Inspectorate Employee Retirement

⟨**Functional area**⟩:	General Management of General Affairs and Personnel
⟨**I-Object**⟩:	{ Employee of Labour Inspectorate }
⟨**O-Object**⟩:	{ Employee of Labour Inspectorate, Retirable Employee of Labour Inspectorate }
⟨**Operation**⟩:	Retire
⟨**Cons-Object** ⟩:	{ Employee of Labour Inspectorate }
⟨**Circ-Object**⟩:	{ Employee dossier }

Fig. 1. An example of process descriptor ($D(P_1)$)

In our approach, process descriptors are semi-automatically constructed, by analyzing the specifications associated with processes. In particular, when possible, features are automatically filled in with information extracted from the ER schemas (e.g., entity names in features concerning objects); if necessary, features are filled in with names manually extracted from textual information in process specifications (e.g., operation names).

A feature f has a set of related names, which are the names used to classify the processes with respect to the feature. Two sets of names are defined for features, namely entity names and operation names.

3 Techniques for terminological analysis

In order to evaluate process similarity, we need to compare the names specified in the features of their descriptors and determine their degree of similarity, that we call *affinity*. To evaluate name affinity, we construct a *semantic dictionary* where entity and operation names are organized into a *concept hierarchy*, as shown in Fig. 2. Two names have affinity if they refer to the same concept or to concepts that are close in the semantic dictionary hierarchy.

The hierarchy is incrementally constructed. First, we define the bottom level concepts for groups of names corresponding to entities and operations that are similar in process descriptors. Then, we abstract higher-level concepts starting from the bottom level concepts, by means of the generalization and aggregation abstraction mechanisms. In this way, we provide the process analyst with a complete overview at different levels of abstraction of the data and functionalities of the Public Administration processes.

To identify entities and operations that are similar in process descriptors, we exploit the information contained in process specifications. In particular, for entities we analyze the ER schemas associated with processes, while, for operations, we perform a categorization based on the features describing functionality in process descriptors. To evaluate entity similarity, we construct a Thesaurus, where the semantics of entities in the ER schemas is described by means of proper terminological relationships. Issues related to Thesaurus construction are discussed in Section 3.1. In Section 3.2, we provide methodological indications for the definition of the concept hierarchy in the semantic dictionary, while in Section 3.3, we discuss name affinity evaluation using the semantic dictionary.

3.1 Thesaurus construction

A Thesaurus is a dictionary storing terms that are relevant for a given application domain, and a set of relevant terminological relationships to manage synonyms and semantically related terms, to support, for example, classification and retrieval of unstructured information, such as documents [16]. The construction of a Thesaurus is based on the analysis of a set of documents and on the identification of the relevant terms and relationships for the domain of interest. In our case, we exploit the available ER schemas to define a Thesaurus specifying the terminological relationships between entity names that are relevant for evaluating entity similarity, in view of defining the bottom level concepts in the semantic dictionary.

Two entities in different schemas are similar if they describe the same or similar real-world objects. To evaluate entity similarity, we consider both the

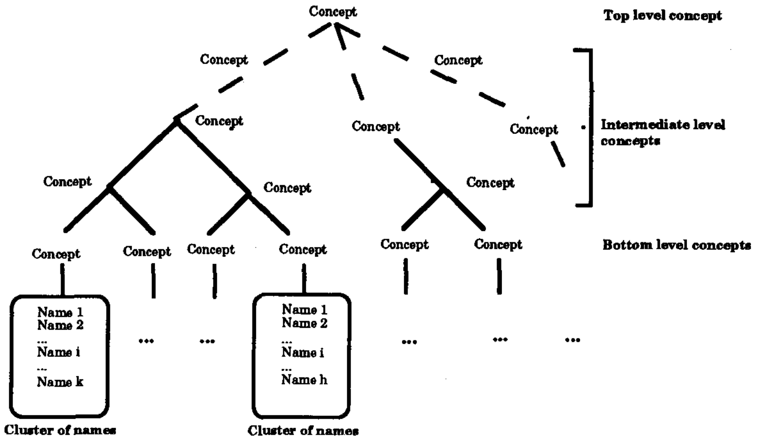

Fig. 2. Architecture of the semantic dictionary

synonymy relationship between their names (*lexical knowledge* on entity similarity) and the structural relationships in which the entities participate in the ER schemas (*schema knowledge* on entity similarity), that is, "is-a" links and relationships. In fact, two entities related by means of an "is-a" link in a generalization hierarchy have a certain degree of similarity due to the semantics associated with generalization hierarchies in conceptual schemas (i.e., inheritance mechanism, subset relationship). Two entities participating in a relationship have a degree of similarity as well, due to the semantics of the relationship in conceptual schemas (i.e., aggregation semantics). To take into account all explicit relationships and "is-a" links defined between entities in ER schemas, we construct the Thesaurus directly from available schemas.

For Thesaurus construction, difficulties are related to the fact that a high number of ER schemas is involved, and the same entity name can have been used to denote different real-world object classes (homonym problem [3]), or different names can have been used to denote the same object class (synonyms problem [3]). While synonyms do not cause problems in view of similarity evaluation, the presence of homonym can lead to the evaluation of undesirable similarities. To exclude homonym in the Thesaurus, we require normalization of the specifications; in fact, repeated entity names are proposed to the process analyst who disambiguates them, if necessary.

In the following, n_i denotes the name associated with entity e_i. Lexical and structural relationships between entities are captured in our Thesaurus by means of the following conventional binary terminological relationships between entity names:

- *USE*, defined between entity names that are considered synonyms because they represent the same class of real-world objects across schemas. We consider as synonyms: an acronym entity name and its expansion, singular entity

names and their plural, an entity name and its incorrectly spelled occurrences in different schemas (e.g., $\langle$*Emp. USE Employee*$\rangle$).
- *BT* (Broader Terms), defined between names of entities participating in generalization hierarchies in the ER process schemas. In particular, $\langle n_i BT n_j \rangle$ is defined for a pair of entities such that e_i is the generalization of e_j in a schema (e.g., $\langle$*Person BT Employee*$\rangle$).
- RT (Related Terms) defined between names of entities participating in a relationship in the ER process schemas (e.g., $\langle$*Employee RT Department*$\rangle$).

These binary relationships have an inverse relationship implied by them, which is also stored in the Thesaurus, to keep the Thesaurus consistent and complete. In particular, the inverse of BT is the NT (Narrower Terms) relationship, defined between a specialization entity and its corresponding generalization, that is, $\langle n_i BT n_j \rangle \rightarrow \langle n_j NT n_i \rangle$; the inverse of the *USE* relationship is *UF* (Used For) relationship defined between synonym names, that is, $\langle n_i USE n_j \rangle \rightarrow \langle n_j UF n_i \rangle$. The inverse of the *RT* relationship is *RT* itself, that is, *RT* is symmetric.

To evaluate entity similarity, we consider:

- *Explicit relationships*, that are defined in the Thesaurus.
- *Implicit relationships*, that can be derived from the explicit ones. They capture the paths of relationships and/or "is-a" links between entities in ER schemas involving at most three entities. In fact, we assume that some degree of affinity exists between two entities due to the fact that they participate in relationship with a third entity, or to the fact that they have a common father in a generalization hierarchy, or a mixed situation (a relationship and an "is-a" link with a common entity). Longer paths are not considered for similarity purposes, because too weak affinities can be generally inferred from them, specially when paths are composed of relationships for which no inheritance semantics holds along the path.

Since, in general, these relationships can occur more than once in the whole set of analyzed schemas, we distinguish also:

- *Homogeneous multiple relationships*, (either explicit or implicit), that is, relationships of a given type with k occurrences in the Thesaurus.
- *Heterogeneous multiple relationships*, (either explicit or implicit), that is, relationships of any type with k occurrences in the Thesaurus.

To quantify the similarity between a pair of entities, we weight terminological relationships in the Thesaurus. We assign a strength $\sigma_{\Re} \in [0,1]$ to each type $\Re$ of terminological relationship, to capture its implication for entity similarity, with $\sigma_{USE/UF} \geq \sigma_{BT/NT} \geq \sigma_{RT}$. The higher the strength, the higher the similarity implication of a given terminological relationship. During experimentation, we used $\sigma_{USE/UF} = 1$, $\sigma_{BT/NT} = 0.8$, and $\sigma_{RT} = 0.5$.

The similarity between two entities e_i and e_j belonging to different ER schemas is computed by means of a similarity coefficient, $Sim(e_i, e_j) \in [0,1]$, which takes into account the type and the strengths of the explicit and implicit relationships of e_i and e_j in the Thesaurus. The higher the number of relationships for two entities in the Thesaurus, the greater the similarity coefficient computed for them. The similarity metrics are detailed described in [7].

Definition 1 Entity similarity. Two entities e_i and e_j are similar, denoted by $e_i \sim e_j$, if their similarity coefficient $Sim(e_i, e_j)$ is greater than or equal to an imposed threshold $\alpha > 0$, that is, $e_i \sim e_j \leftrightarrow Sim(e_i, e_j) \geq \alpha$.

3.2 Semantic dictionary construction

In building the semantic dictionary, we start from the concepts at the bottom level of the hierarchy, which are defined as representative of entities and operations that are similar.

To identify groups of similar entities on the basis of the similarity coefficients computed for pairs of entities, we apply the *complete link* clustering technique [9]. This technique is generally used in information retrieval to classify documents according to similarity levels [16]. It is a hierarchical agglomerative technique which produces as the output a tree of entity clusters; each cluster has associated a similarity coefficient that holds for all possible pairs of entities belonging to the cluster.

To identify groups of similar operations, we stay on a categorization based on the triplet ⟨*Operation*, *Cons-Object*, *Circ-Object*⟩, that provides indications for classification according to [8]. In particular, the following clusters of similar operations are identified:

1. "Exchange of objects with outside" (other processes, users). Operations pertaining to this cluster deal with information, documents and/or material exchange, such as "Send retirement dossier to Administrator". Examples of operation names (i.e., verbs) occurring in process specifications are 'communicate', 'deliver', 'display', 'distribute', 'give', 'obtain', 'present', 'receive', 'show'.
2. "Creation of objects". Operations pertaining to this cluster deal with generation of objects, such as "Define a retirement dossier". Examples of verbs occurring in process specifications are 'compile', 'construct', 'define', 'make a draft of', 'make', 'prepare', 'produce', 'retire'.
3. "Transformation of objects". Operations pertaining to this cluster deal with changes to objects, varying their identity, such as "Compose retirement requests". Examples of verbs occurring in process specifications are 'compact', 'compose', 'cut', 'decompose', 'divide', 'join', 'merge', 'record', 'split'.
4. "Modification/Observation" of the status of input objects. Operations pertaining to this cluster deal with modifications of objects while preserving their type, such as "Examine requests". Examples of verbs occurring in process specifications are 'check', 'choose', 'dismiss', 'examine', 'fill in', 'list', 'protect', 'retrieve', 'select', 'sign', 'test', 'update'.
5. "Deletion or removal of objects". Operations pertaining to this cluster are operations destroying objects, such as "Annul dossier". Examples of verbs

occurring in process specifications are 'annul', 'cancel', 'delete', 'eliminate', 'erase'.

The names of entities and operations in defined clusters constitute the starting point for defining the concept hierarchy in the semantic dictionary. In particular, a bottom-level concept is defined for each cluster of names, on the basis of the characteristics and semantics of the entities/operations within the cluster. The concept defined for a cluster is the generalization of the entities/operations associated with the cluster's names. Responsible for concept definition is the process analyst, who examines the clusters and identifies a suitable concept name to represent all the cluster members. Concept hierarchies are defined separately for entities and operations. The main methodological suggestions for concept definition are the following:

- Entity name clusters: the concept name can be selected starting from the names of the entities belonging to a given cluster, or can be defined from scratch on the basis of the semantics of the entities within the cluster. The following cases can occur:
  1. Multi-term names.
     For entities with multi-term names, one (or more than one) term is (are) common to all the names belonging to the cluster, and the common term(s) denote a concept. This means that the entity names are specialization of a common generic entity. The common term(s) is (are) extracted to become the name of the concept representative of the cluster, as illustrated in the example of Fig. 3. Here the concepts **Employee of Labour Ministry**, **Employee of Labour Inspectorate**, and **Employee** are defined for each respective cluster, since they are common to all the names in each cluster. These concepts are at different levels of abstraction in the hierarchy.
  2. Single-term names.
     For single-term names, if one name denotes an entity which is a generalization of the entities represented by the other cluster's entities, the process analyst selects this name to become the concept name associated with the cluster. Otherwise, the process analyst selects a new generic name for the cluster's concept.
  3. Mixed names.
     This is a mixed situation, and the process analyst proceeds as in cases 1 and 2, by considering the common term(s) of multi-term names and the single-term names as candidates for the definition of the concept name.
- Operation name clusters: the concept defined for a cluster corresponds to the type of the operation associated with the cluster. As a consequence, the concepts **Exchange**, **Creation**, **Transformation**, **Modification /Observation**, and **Deletion** are defined, as illustrated in Fig. 4.

Higher-level concepts in the hierarchy are incrementally abstracted, if necessary, starting from the bottom level concepts, by means of the generalization and

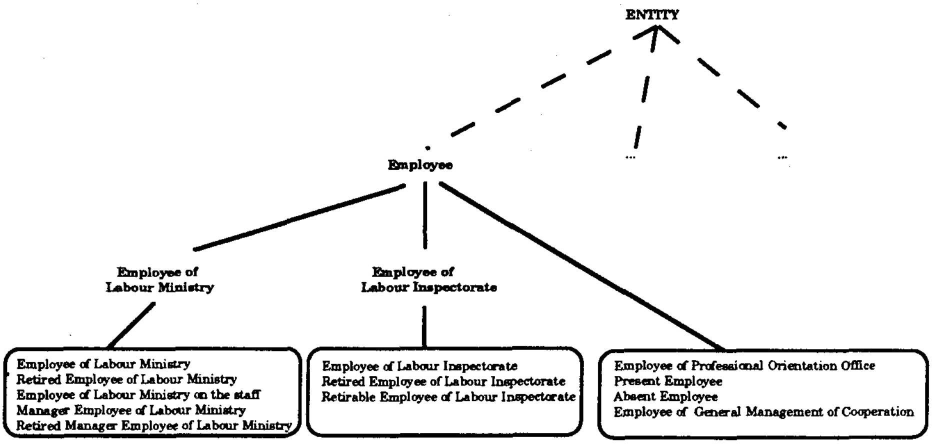
ENTITY
...
...
Employee
Employee of
Labour Ministry
Employee of
Labour Inspectorate
Employee of Labour Ministry
Retired Employee of Labour Ministry
Employee of Labour Ministry on the staff
Manager Employee of Labour Ministry
Retired Manager Employee of Labour Ministry
Employee of Labour Inspectorate
Retired Employee of Labour Inspectorate
Retirable Employee of Labour Inspectorate
Employee of Professional Orientation Office
Present Employee
Absent Employee
Employee of General Management of Cooperation

Fig. 3. An example of concept hierarchy for entities

the aggregation abstraction mechanisms, taking into account also the knowledge about the standards, rules and conventions adopted in the Public Administration.

3.3 Accessing the semantic dictionary

Concepts in the semantic dictionary are connected by means of links with other concepts in the hierarchy. Moreover, bottom level concepts are connected by means of links with corresponding clusters of names. To operationally evaluate name affinity, we assign a strength σ to links in the semantic dictionary (e.g., in our experimentation, we used $\sigma = 0.8$).

The affinity of two names depends on the length of the path between them in the semantic dictionary. The higher the number of links in the path between two names, the lower the affinity of the involved names. A path between two names in the semantic dictionary is denoted by the symbol "$\Rightarrow^l$", where $l > 0$ is the length of the path. The strength of a path "$\Rightarrow^l$" is computed as the combination of the strength of the involved links, using the monotonic function $\tau^N : N \times [0,1] \rightarrow [0,1]$, with $\tau^N(l,\sigma) = (\sigma)^l$.

Definition 2 Name Affinity Coefficient. The Affinity Coefficient $A(n_i, n_j)$ between two names n_i and n_j is the measure of their affinity in the semantic dictionary computed as follows:

$$A(n_i, n_j) = \begin{cases} 1 & \text{if } (n_i = n_j) \\ \tau^N(l,\sigma) & \text{if } n_i \Rightarrow^l n_j \\ 0 & \text{otherwise} \end{cases}$$

According to this definition, $A(n_i, n_j)$ is a numerical value in the range $[0,1]$; it is 1 if the names are identical, is the τ^N strength of the path between them in

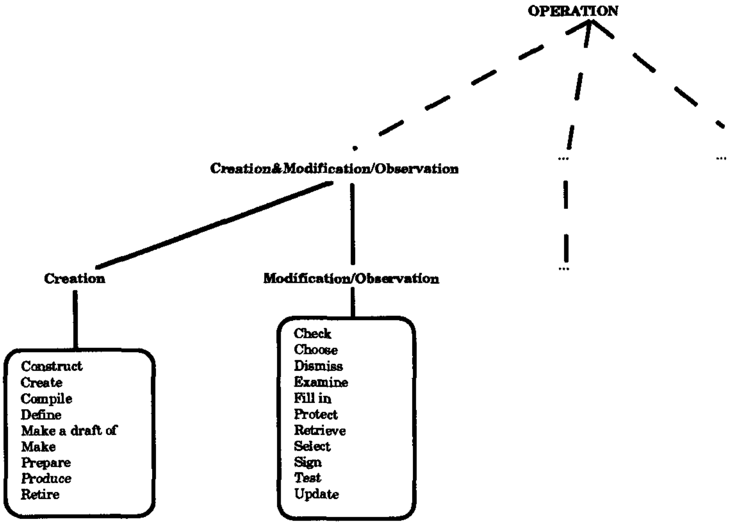

Fig. 4. An example of concept hierarchy for operations

the semantic dictionary, and is 0 otherwise. An affinity threshold can be imposed to filter out the names with affinity.

Definition 3 Name Affinity. Two names n_i and n_j have affinity, denoted by $n_i \sim n_j$, if their affinity coefficient is greater than or equal to an imposed threshold $\beta > 0$, that is, $n_i \sim n_j \leftrightarrow A(n_i, n_j) \geq \beta$.

4 Techniques for process similarity evaluation

Process similarity is evaluated on the basis of the descriptors associated with them, by comparing the involved features. Before illustrating how to measure process similarity, let us describe how feature similarity is determined for process descriptors.

Two features f and f' are *comparable* if they refer to the same set of names (i.e., entity names, operation names). The similarity between a pair of comparable features in different process descriptors is a function computed on the names they contain. Let P_i be a process specification, and $D(P_i)$ its corresponding descriptor. The notation $D(P_i).f$ indicates the name n or the set of names $n_1, \ldots, n_q$ specified in the feature f of descriptor $D(P_i)$.

Definition 4 Feature similarity. The *feature similarity* of a pair of comparable features f and f' of process descriptors $D(P_i)$ and $D(P_j)$, denoted by $Sim(D(P_i).f, D(P_j).f')$, is the measure of the affinity between their names, computed as follows.

$$Sim(D(P_i).f, D(P_j).f') = \begin{cases} A(n,n') & \text{if } f, f' \text{ are single-name} \\ & \text{and } n \sim n' \\ \frac{2 \cdot |F|}{|D(P_i).f| + |D(P_j).f'|} & \text{if } f, f' \text{ are multiple-name} \\ & \text{and } F \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where $F = \{\langle n,n' \rangle \mid n \in D(P_i).f, n' \in D(P_j).f', n \sim n'\}$ is a set composed of the pairs of names of $D(P_i).f$ and $D(P_j).f'$ that have affinity, and notation $\mid D(P_k).f \mid$ indicates the number of names contained in feature f of the descriptor $D(P_k)$.

Definition 4 states that the similarity of two comparable features is the affinity coefficient of their names in the semantic dictionary, if both features contain only one name, is the number of their names that have affinity multiplied by two and divided by the total number of their names (Dice's metric [16]), if features contain more than one name, and is 0 otherwise. According to the Dice's metric, an affinity mapping Φ is defined between the names of two compared features $D(P_i).f$ and $D(P_j).f'$. We require a $1-1$ mapping, that is, each $n \in D(P_i).f$ is mapped into at most one name $n' \in D(P_j).f'$ and vice versa. The set F is composed of the pair of names $\langle n, n' \rangle$ participating in the affinity mapping Φ. The mapping Φ can be total or partial. Φ is total if for each name $n \in D(P_i).f$ there exists a corresponding name $n' \in D(P_j).f'$ with affinity, and vice versa. Φ is partial if some name in $D(P_i).f$ or in $D(P_i).f'$ is left unmapped. The greater the number of pairs that participate in Φ, the higher the feature similarity.

The similarity between two process specifications is computed with respect to the involved entities (entity-based similarity) and to the performed functionality (functionality-based similarity), by means of proper *similarity coefficients*.

4.1 Entity-based similarity

Definition 5 Entity-based similarity coefficient. The *Entity-based similarity coefficient* of two process specifications P_i and P_j having descriptors $D(P_i)$ and $D(P_j)$, denoted by $ESim(P_i, P_j)$, is the measure of the similarity of their comparable and equal features related to input and output entities:

$$ESim(P_i, P_j) = \sum_{f \in \Im} Sim(D(P_i).f, D(P_j).f)$$

where $\Im$={I-Object,O-Object}.

Definition 5 states that the entity-based similarity of two process specifications is the sum of the similarity of their comparable and equal features related to the input and output entities. According to what stated for name affinity coefficients, $ESim(P_i, P_j)$ can assume values in the range $[0, 2]$. It has value 0 when all features $f \in \Im$ are not similar; it has value 2 when both features *I-Object* and *O-Object* have similarity equal to 1.

4.2 Functionality-based similarity

Definition 6 Functionality-based similarity coefficient. The *Functionality -based similarity coefficient* of two process specifications P_i and P_j having descriptors $D(P_i)$ and $D(P_j)$, denoted by $FSim(P_i, P_j)$, is the measure of the similarity of their comparable and equal features related to functionality:

$$FSim(P_i, P_j) = \sum_{f \in \Im} Sim(D(P_i).f, D(P_j).f)$$

where $\Im = \{$Operation, Cons-Object, Circ-Object$\}$.

Definition 6 states that the functionality similarity of two process specifications is the sum of the similarity coefficients of their comparable and equal features used to describe process functionality. This coefficient returns a similarity value in the range range $[0, 3]$, depending on the number of features that have similarity, and on the feature similarity value Sim.

5 A methodological framework for process unification

The term "unification" is used to denote the combination of process specifications by identification of semantic correspondences between them, as inspired by literature on semantic heterogeneity [14].

The goal of our work is to facilitate Information Systems interconnection and interoperability by constructing a unified process architecture. In the proposed approach, we distinguish two steps. First, naming and structural conflicts between process specifications must be detected and resolved with the help of the semantic dictionary. Second, semantic correspondences between process specifications must be identified for their combination.

5.1 Conflict resolution

Conflicts have been widely studied in schema integration literature [3]. They are generally classified as follows:

- *Name conflicts.* These conflicts arise in presence of homonym and/or synonyms.
- *Structural conflicts.* They are distinguished in *type conflicts* and *link conflicts.* The former arise when the same concept has been modeled using different constructs. The latter arise when the same entities are related through different types of links, or through links having different integrity constraints (e.g., relationships with different cardinalities).

In our approach, the availability of the semantic dictionary allows the schema analyst to properly handle the problems to gain good resolution. In particular, the semantic dictionary allows the normalization of entity and operation names used for semantically similar processes, according to the concept hierarchy therein contained. In addition, for structural conflicts, the process analyst examines the ER schemas associated with the entity names contained in the same cluster in the semantic dictionary, and decides how to resolve them on the basis also of his own experience and knowledge on the Public Administration rules. In general, structural conflicts are resolved by selecting one representation as the reference representation for the involved similar process specifications, as described in [5].

5.2 Semantic correspondences

Semantic correspondences are established between processes on the basis of their level of similarity. Given two process specifications P_i and P_j with associated descriptors $D(P_i)$ and $D(P_j)$, three significant cases can be identified:

1. *Semantic equivalence.* This is the strongest measure of similarity between two processes, and indicates that P_i and P_j represent the same real-world activity. With respect to our similarity coefficients, two processes are semantically equivalent if:
   (a) $ESim(P_i, P_j) = 2$ and $FSim(P_i, P_j) = 3$ and
   (b) a total $1-1$ mapping Φ is defined for names of each comparable feature of $D(P_i)$ and $D(P_j)$ and all names in the affinity pairs are equal.

   As a consequence, two semantic equivalent processes present the same involved objects and the same functionality.
2. *Semantic relationship.* This is a weaker measure of similarity between two process specifications than semantic equivalence, and indicates that P_i and P_j represent partially overlapping real-world activities, that is, activities performing similar functionalities on similar data. With respect to our similarity coefficients, two processes have a semantic relationship if:
   (a) $1 \leq ESim(P_i, P_j) \leq 2$ and $2 \leq FSim(P_i, P_j) \leq 3$ and
   (b) a $1-1$ mapping Φ, either partial or total, is defined for names of each comparable feature of $D(P_i)$ and $D(P_j)$.

   As an example of processes with semantic relationship, let us consider the processes P_2 and P_3, recording the absences of employees, whose descriptors $D(P_2)$ and $D(P_3)$ are shown in Fig. 5 and in Fig. 6, respectively. We have that $ESim(P_2, P_3) = 2$ because Φ is total for both features *I-Object* of P_2 and P_3 and *O-Object* of P_2 and P_3. In fact, **Employee of General Management of Cooperation** of P_2 and **Employee of Professional Orientation Office** of P_3 have affinity, because they refer to the same concept **Employee** in the semantic dictionary, while remaining entity names in the considered features are equal in both descriptors $D(P_2)$ and $D(P_3)$. $FSim(P_2, P_3) = 3$ because both process operations refer to the **Transformation** concept in the semantic dictionary, the circumstantial objects have affinity in the semantic dictionary, and the constitutive objects are equal in both descriptors.
3. *Commonality.* This is the weakest measure of similarity between two process specifications, and indicates that P_i and P_j represent overlapping real-world

Personnel Absence Recording

⟨Functional area⟩:	General Management of Cooperation
⟨I-Object⟩:	{ Employee of General Management of Cooperation, Present Employee }
⟨O-Object⟩:	{Employee of General Management of Cooperation, Absent Employee }
⟨Operation⟩:	Record
⟨Cons-Object ⟩:	{Employee of General Management of Cooperation }
⟨Circ-Object⟩:	{ Presence Card }

Fig. 5. An example of process descriptor $(D(P_2))$

Absence Recording

⟨Functional area⟩:	Central Office for Professional Education and Orientation
⟨I-Object⟩:	{Employee of Professional Orientation Office, Present Employee }
⟨O-Object⟩:	{Employee of Professional Orientation Office, Absent Employee}
⟨Operation⟩:	Record
⟨Cons-Object ⟩:	{ Employee of Professional Orientation Office }
⟨Circ-Object⟩:	{ Presence Card }

Fig. 6. An example of process descriptor $(D(P_3))$

activities, with respect to their manipulated data and/or functionality. With respect to our similarity coefficients, two processes have commonality if they do not fall in the categories 1. and 2. and if $ESim(P_i, P_j) \geq \alpha$ and/or $FSim(P_i, P_j) \geq \beta$, where α and β are similarity thresholds defined by the process analyst to filter out similar processes. This means that a partial $1-1$ mapping Φ is defined for names of a subset comparable features of $D(P_i)$ and $D(P_j)$. As an example of processes with resemblance, let us consider the processes P_1 and P_2, whose descriptors $D(P_1)$ and $D(P_2)$ are shown in Fig. 1 and in Fig. 5, respectively. For them we have that $ESim(P_1, P_2) =$ 1.66, since similarity of features $D(P_1)$.*I-Object* and $D(P_2)$.*I-Object* is 0.66 because the entity **Employee of Labour Inspectorate** of P_1 has affinity with **Employee of General Management of Cooperation**, while similarity of features $D(P_1)$.*O-Object* and $D(P_2)$.*O-Object* is 1 because the two entities specified in $D(P_1)$.*O-Object* have affinity with the two entities specified in $D(P_2)$.*O-Object*. Moreover, $FSim(P_1, P_2) = 0.512$. In fact, operations of P_1 and P_2 do not have affinity since they refer to different operation concepts in the semantic dictionary (i.e., **Creation** for P_1 and **Transformation** for P_2, supposing an affinity threshold $\beta = 0.5$ for operations). The similarity between features $D(P_1)$.*Cons-Object* and $D(P_2)$.*Cons-Object* is 0.512, which is the affinity value of **Employee of Labour Inspectorate** and **Employee**

of General Management of Cooperation in the semantic dictionary. Finally, the similarity of features $D(P_1).\mathit{Circ\text{-}Object}$ and $D(P_2).\mathit{Circ\text{-}Object}$ is 0 because **Employee dossier** and **Presence card** do not have affinity in the dictionary.

5.3 Process unification

Once naming and structural conflicts have been resolved, it is possible to proceed to the construction of a unified architecture of semantically related processes. Depending on the type of semantic correspondence, the following cases are identified:

1. *Process paradigm.*
For semantic equivalent processes, we say that a "process paradigm", i.e., a standardized definition of the process can be defined. Process paradigms are reference specifications that can be reused in developing new applications similar to existing applications, and can guide maintenance and evolution strategies.
2. *Process hierarchy.*
For semantically related processes (e.g., P_2 and P_3), we say that a "process hierarchy", i.e., an "is-a" hierarchy, between them can be defined, over the involved functionalities and, possibly, the objects. According to "is-a" hierarchies, features of supertype and subtype are properly distributed. Use of process hierarchies is the basis for accommodating multiple user perspectives on comparable, semantically related processes.
3. *Process cluster.*
For processes with commonality (e.g., P_1 and P_2), we say that a "process cluster", i.e., a group of processes related by a similarity value, can be defined. Clusters are useful to facilitate browsing of domain knowledge by classification for significant features. For example, rather high entity similarity and low functionality similarity coefficients for processes in a cluster can indicate the existence of information flows between the processes in the cluster, which can be examined to possibly reconstruct macroprocesses. Identification of information flows between processes and reconstruction of macroprocess is discussed in [6], where suitable *Closeness* coefficients are described for this purpose.

6 Supporting tools

An environment to support the process analysis techniques previously illustrated has been developed, based on a repository developed by the AIPA for storage of the ER schemas and the information in process specifications. The implementation environment is PC-based, using Access 2.0. The AIPA repository provides functionalities for visualization of schemas and associated information, both with textual information format and with a simple graphical editor for ER schemas.

Concerning process analysis support, the following functionalities have been implemented:

- Definition of process descriptors; the tool extracts features related to objects (i.e., *I-Object, O-Object, Cons-Object* and *Circ-Object*) directly from the schemas stored in the repository; the *Operation* feature is interactively specified by the process analyst on the basis of the process specifications retrieved from the repository.
- Construction and consultation of the Thesaurus. The Thesaurus is automatically constructed in form of Access tables, by analyzing the ER schemas stored in the repository. A separate table is defined for the *USE* relationship for optimization purposes; spelling checker tools are employed to identify equivalent entity names. The tables for *BT/NT* and *RT* relationships are constructed by inserting a row for each pair of entities connected by means of an "is-a" link or participating in a relationship, and by inserting the corresponding σ value for the involved relationship. Functionalities for the consultation of the Thesaurus have been implemented to support:
  1. the selection of a pair of entities and the computation of their similarity coefficient;
  2. the selection of an entity e_i and the retrieval of the entities that are similar to e_i, with a similarity coefficient *Sim* greater than or equal to a (interactively set) threshold;
  3. the section of an entity and the retrieval of the entities of the Thesaurus that have a relationship with e_i in the Thesaurus.
- Grouping of similar entities into clusters, using the complete link clustering technique. Entities are submitted to pairwise similarity comparisons, and the similarity matrix is generated for the complete link clustering algorithm. The entity clusters obtained with the technique are presented to the process analyst as a tree of entities, similarly to the interface presented in [17]. Starting from the defined clusters, it is possible to interactively extract the corresponding concepts, on the basis also of the structure of names included in the cluster. Defined concepts are automatically associated with the cluster from which they are extracted. Clusters for operation names are interactively defined for each category of operations, and the type of operation becomes the concept associated with the corresponding cluster. A graphical interface is under development, to show the clusters of similar names and their corresponding abstract concepts.
- Process similarity evaluation, based on the entity similarity and the functionality similarity coefficients. The process analyst can select from a menu the coefficient(s) to be used for similarity computation, and a reference process for similarity evaluation. The user of the process analyzer can also choose to produce reports retrieving all processes related to the reference process in decreasing order of similarity, which are the basis for the unification strategies.

Retrieval and graphical presentation functionalities have been realized using the development environment provided by Access.

7 Related work

The problems discussed in the paper are related to researches concerning semantic heterogeneity in multidatabases and process reengineering.
Semantic heterogeneity.
Techniques to identify semantic similarity between data are essential in federated database systems, in view of schema integration and query processing activities [19]. Recent approaches to deal with semantic heterogeneity suggest the use of advanced data dictionaries, where taxonomies for the data in the federation are defined to provide flexible query processing facilities. In [4], a hierarchical data structure called "Summary Schema Model" is proposed, where concepts represented in the federation schemas are mapped to a pre-defined taxonomy, organized at different levels of abstraction according to linguistic relationships. Here, a manual activity is required to map concepts of local databases into a corresponding taxonomy term. In [15], fuzzy techniques are discussed to evaluate the semantic similarity between classes in different conceptual schemas. The proposed techniques exploit both an associative network of terms and the knowledge about the semantic relevance of attributes to classes, which are properly combined by means of similarity functions. In [14], a semantic dictionary architecture is proposed for multidatabase systems, where federated generic knowledge is organized according to pre-defined semantic relationships, which provide support for retrieval of information similar or related to the information managed in a local database. Also in this case, a manual activity is required to specify the relationships for the objects of different local schemas in the dictionary. Our approach tries to limit the manual activity required to build the semantic dictionary for the ER schemas, by identifying concept hierarchies from the Thesaurus, instead of referring to pre-existing taxonomies. Thesaurus relationships are automatically extracted from the ER schemas available in the repository. The concept hierarchy obtained in this way is more specific than a pre-defined taxonomy, but has the advantage of capturing similarities between entities on the basis of their actual use. However, a manual support is required to the process analyst to abstract higher level concepts in the hierarchy. The development of a semi-automatic technique for the semantic dictionary construction has been an essential requirement of our work, due to the large number of conceptual schemas and process descriptors to be analyzed.
Process reengineering.
In the area of process reengineering, workflow specifications are studied to provide formal descriptions of processes of an organization and workflow management systems for their implementation are available to facilitate process analysis and reengineering [11]. A conceptual description of tasks, executing entities, and task coordination constraints for business processes of an organization facilitates process understanding, evaluation and redesign to optimize existing processes

and adapt them to changing requirements. In reengineering processes at the technological level, new software lifecycles are required, to adequately represent reengineering processes in the lifecycle and support software development based on existing legacy software, to acquire and (re)use the domain knowledge therein contained [1]. In [2], data reverse engineering techniques are discussed, to support reengineering and cross-functional integration of information systems in the Public Sector, focusing on techniques to address data disintegration problems in large organizations. In this framework, our similarity techniques constitute a support to perform process analysis in view of reengineering. Process descriptors provide a high-level description of processes, pointing out the interaction aspects between processes in form of data exchanges. Similarity techniques help the process analyst in identifying groups of processes manipulating the same or similar data, and groups of processes that are candidate to constitute a macroprocess, which are a good basis to evaluate data distribution strategies and process optimization interventions during reengineering.

8 Concluding remarks

In the paper, we have presented an approach to support information system process reengineering at a conceptual level, by addressing issues related to process specification comparison and unification. In particular, we have proposed: i) techniques for analyzing a large set of process specifications, based on the construction of formal descriptors, providing a high-level characterization of the processes in terms of manipulated data and performed functionality in a given functional area; ii) techniques for evaluating data and functionality similarity between different processes, based on the construction of a Thesaurus, capturing the semantics of the ER schemas associated with processes, and of a semantic dictionary, where a concept hierarchy is introduced to provide the process analyst with aggregated and generalized knowledge for entities and operations; iii) a methodological framework for process unification, to identify groups of semantically related processes that can be considered together in reengineering interventions, because they perform similar operations and/or involve similar data.

Actually, entity attributes are not considered for entity affinity evaluation, because they are specified only for a restricted number of ER schemas in process specifications. However, the approach can be easily extended to include attributes. One way to extend the approach is to define a terminological relationship in the Thesaurus representing the attributes of an entity (e.g., CT, Component-Terms), which is used to determine the affinity coefficients together with existing terminological relationships. A second way is to apply the present approach to attribute names separately, that is, a Thesaurus of attribute names is constructed and clusters of similar attributes are derived on the basis of the affinity coefficients computed on the Thesaurus. An attribute hierarchy can be defined for attribute clusters similarly to the approach for schema integration presented in [18]. Entity affinity is then evaluated on the basis also of the attribute hierarchy. The presented tools are being experimented and their extension is under investigation.

Acknowledgements
Part of this work has been supported by the Information System Authority for Public Administration (AIPA). The authors thank Dr. P. Giani and Dr. S. Konstadopulu who developed the process analysis support tools, and the AIPA for providing the process specifications.

References

1. J.D. Ahrens, N.S. Prywes, "Transition to a Legacy- and Reuse- Based Software Life Cycle", *IEEE Computer,* October 1995.
2. P. Aiken, A. Muntz, R. Richards, "DoD Legacy Systems - Reverse Engineering Data Requirements", *Communications of the ACM,* vol.37, no.5, May 1994.
3. C.Batini, M. Lenzerini, S. Navathe, "A Comprehensive Analysis of Methodologies for Database Schema Integration", *ACM Computing Surveys,* September 1986.
4. M.W. Bright, A.R. Hurson, S. Pakzad, "Automated Resolution of Semantic Heterogeneity in Multidatabases", *ACM Transactions On Database Systems,* vol.19, no.2, June 1994.
5. S. Castano, V. De Antonellis, "Reference Conceptual Architectures for Reengineering Information Systems", *International Journal of Cooperative Information Systems,* vol.4, nos.2 & 3, 1995.
6. S. Castano, V. De Antonellis, "Reengineering Processes in Public Administrations", in *Proc. of OO-ER'95, Int. Conf. on the Object-Oriented and Entity-Relationship Modeling,* Gold Coast, Australia, December 1995, Springer Verlag.
7. S. Castano, V. De Antonellis, "A Systematic Approach to Process Analysis for Reengineering", Technical Report, Politecnico di Milano, 1996 (in preparation).
8. V. De Antonellis, B. Zonta, "A disciplined Approach to Office Analysis", *IEEE Transactions on Software Engineering,* vol.16, no.8, August 1990.
9. B. Everitt, *Cluster Analysis,* Heinemann Educational Books Ltd, Social Science Research Council, 1974.
10. D. H.C. Gall, R.R. Klosch, R.T. Mittermeir, "Object-Oriented Re-Architecturing: An Approach to Long-Term Information Systems Evolution", in *Proc. of the ICSE'95, Int. Conference on Software Engineering Conference,* Seattle, USA, April 1995.
11. D. Georgakopoulos, M. Hornik, A. Sheth, "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure", *Distributed and Parallel Databases,,* Kluwer Academic Publishers, Vol.3, 1995.
12. I. Jacobson, *Object-Oriented Software Engineering - A Use Case Driven Approach,* ACM Press, Addison-Wesley, 1992.
13. W. Kim, I. Choi, S. Gala, M. Scheevel, "On Resolving Schematic Heterogeneity in Multidatabase Systems", *Distributed and Parallel Databases,* vol.1, no.3, 1993, and in *Modern Database Systems-The Object Model, Interoperability and Beyond,* W. Kim (Editor), ACM Press, 1995.
14. J. Hammer, D. McLeod, "An Approach to Resolving Semantic Heterogeneity in a Federation of Autonomous Heterogeneous Database Systems", *International Journal of Intelligent and Cooperative Information Systems,* vol.2, no.1, June 1993.

15. P. Fankhauser, M. Kracker, E.J. Neuhold, "Semantic vs. Structural Resemblance of Classes", *SIGMOD RECORD*, vol.20, no.4, December 1991.
16. G. Salton, *Automatic Text Processing - The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, 1989.
17. G. Salton, J. Allan, C. Buckley, "Automatic Structuring and Retrieval of Large Text Files", *Communications of the ACM*, vol.37, no.2, February 1994.
18. A.P. Sheth, S.K. Gala, S.B. Navathe, "On Automatic Reasoning For Schema Integration", *International Journal of Intelligent and Cooperative Information Systems*, vol.2, no.1, June 1993.
19. A.P. Sheth and J.P. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, vol. 22, no. 3, September 1990.